Module 4

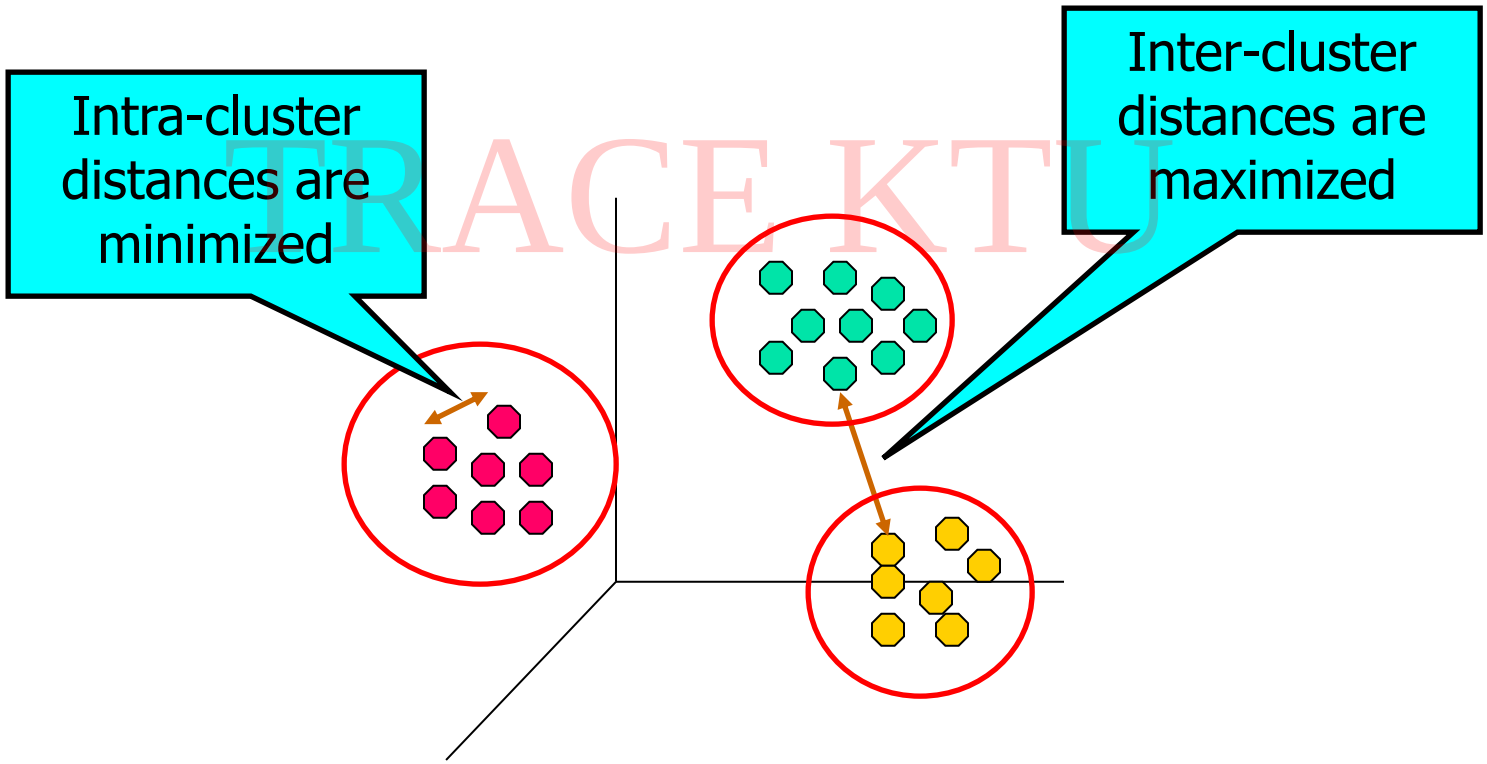
Module-4 (Unsupervised Learning)

Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction – Principal Component Analysis.

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters.
  - Clustering or cluster analysis is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms.

What is Cluster Analysis?



Differentiate between inter cluster distance and intra cluster distance

- Intra-cluster distance is the distance between a data item and the cluster centroid within a cluster. Inter-cluster distance is the distance between the data items in distinct clusters.

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Climate: understanding earth climate, find patterns of atmospheric and ocean

Similarity Measures
Distance between data points

Name	Formula
Euclidean distance	$\|\vec{x}-\vec{y}\|_2 = \sqrt{(x_1-y_1)^2+\cdots+(x_n-y_n)^2}$
Squared Euclidean distance	$\|\vec{x}-\vec{y}\|_2^2 = (x_1-y_1)^2+\cdots+(x_n-y_n)^2$
Manhattan distance	$\|\vec{x}-\vec{y}\|_1 = \|x_1-y_1\|+\cdots+\|x_n-y_n\|$
Maximum distance	$\|\vec{x}-\vec{y}\|_\infty = \max\{\|x_1-y_1\|,\ldots,\|x_n-y_n\|\}$

City block distance /Manhanttan
Chess board distance/Max distance

- Minkowski Distance
  - Minkowski Distance is the generalized form of Euclidean and Manhattan Distance

The formula for Minkowski Distance is given as:

$$D = \left(\sum_{i=1}^{n} |p_i - q_i|^p \right)^{1/p}$$

- Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

- (i) Compute the Euclidean distance between the two objects.
- (ii) Compute the Manhattan distance between the two objects
- . (iii) Compute the Minkowski distance between the two objects, using p = 3

- Consider two data points in two dimensional A(15,18) and B(18,29).Calculate
- Manhattan distance and Chessboard distance between A & B

Partitioning Algorithms

- Construct a partition of a database *D* of *m* objects into a set of *k* clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Heuristic method: *k-means* (MacQueen, 1967)

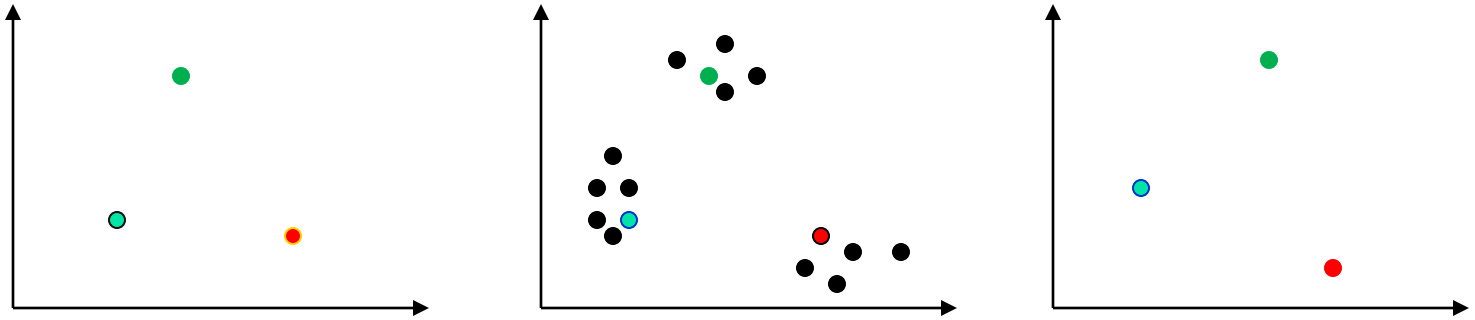
K-means

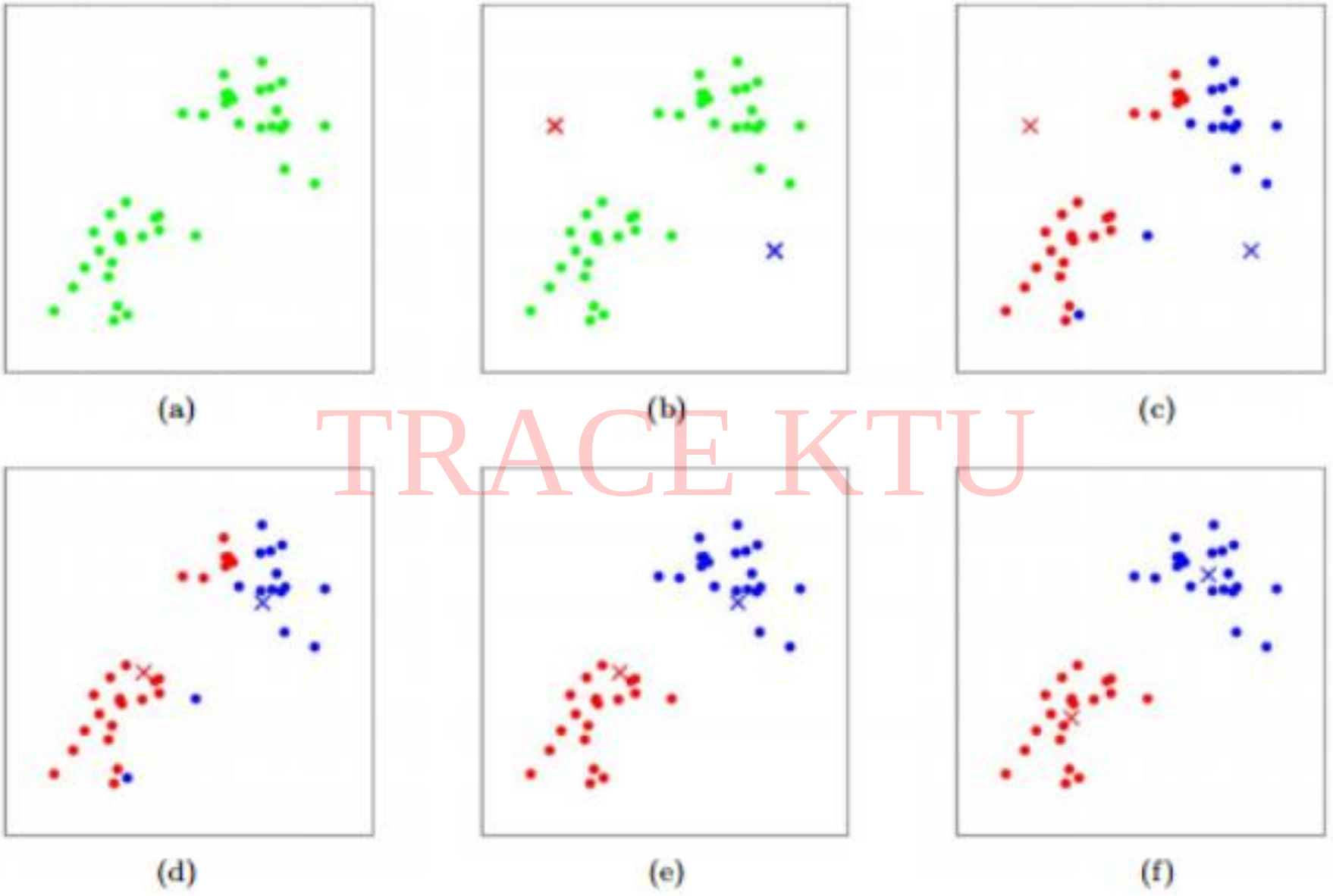
- The k-means clustering algorithm is one of the simplest unsupervised learning algorithms for solving the clustering problem.
- Let it be required to classify a given data set into a certain number of clusters, say, k clusters.

K-means Clustering algorithm

Given *k*

1. Randomly choose *k* data points (seeds) to be the initial cluster centres
2. Assign each data point to the closest cluster centre
3. Re-compute the cluster centres using the current cluster memberships.
   - ie: take the averages of the data points associated with a centre and replace the centre with the average, and this is done for each of the centres.
4. If a convergence criterion is not met, go to 2.





Stopping/convergence criterion

OR

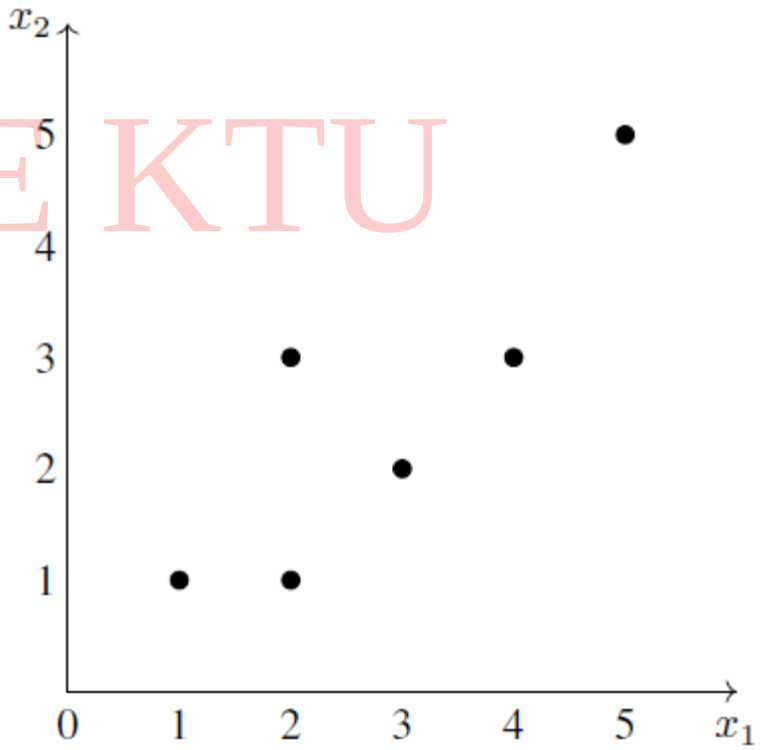
1. no re-assignments of data points to different clusters
2. no (or minimum) change of centroids
3. minimum decrease in the *sum of squared error*

$$SSE = \sum_{i=1}^{k} \sum_{x \in S_i} \|x_i - \mu_i\|^2$$

Example-K-Means [1/8]

Use k-means clustering algorithm to divide the following data into two clusters and also compute the representative data points for the clusters.

x_1	1	2	2	3	4	5
x_2	1	1	3	2	3	5



Example-Kmeans [2/8]

- In the problem, the required number of clusters is 2 and we take k = 2.
- We choose two points arbitrarily as the initial cluster centres.

x_1	1	2	2	3	4	5
x_2	1	1	3	2	3	5

$\vec{v}_1 = (2,1), \quad \vec{v}_2 = (2,3)$

- We compute the distances of the given data points from the cluster centers.

Example-Kmeans [3/8]

- We compute the distances of the given data points from the cluster centers

$\vec{x}_i$	Data point	Distance from $\vec{v}_1 = (2,1)$	Distance from $\vec{v}_2 = (2,3)$	Minimum distance	Assigned center
$\vec{x}_1$	$(1,1)$	1	2.24	1	$\vec{v}_1$
$\vec{x}_2$	$(2,1)$	0	2	0	$\vec{v}_1$
$\vec{x}_3$	$(2,3)$	2	0	0	$\vec{v}_2$
$\vec{x}_4$	$(3,2)$	1.41	1.41	**1.41**	$\vec{v}_1$
$\vec{x}_5$	$(4,3)$	2.82	2	2	$\vec{v}_2$
$\vec{x}_6$	$(5,5)$	5	3.61	3.61	$\vec{v}_2$

Distance:

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

Cluster 1: $\{\vec{x}_1, \vec{x}_2, \vec{x}_4\}$ represented by $\vec{v}_1$

Number of data points in Cluster 1: $c_1 = 3$.

Cluster 2 : $\{\vec{x}_3, \vec{x}_5, \vec{x}_6\}$ represented by $\vec{v}_2$

Number of data points in Cluster 2: $c_2 = 3$.

Example-Kmeans [4/8]

- Cluster centres are recalculated

$$\vec{v}_1 = \frac{1}{c_1}(\vec{x}_1 + \vec{x}_2 + \vec{x}_4)$$
$$= \frac{1}{3}(\vec{x}_1 + \vec{x}_2 + \vec{x}_4)$$
$$= (2.00, 1.33)$$
$$\vec{v}_2 = \frac{1}{c_2}(\vec{x}_3 + \vec{x}_5 + \vec{x}_6)$$
$$= \frac{1}{3}(\vec{x}_3 + \vec{x}_5 + \vec{x}_6)$$
$$= (3.67, 3.67)$$

Example-Kmeans [5/8]

- Compute distance of the given data points from new cluster centres [(2, 1.33) & (3.67, 3.67)]

$\vec{x}_i$	Data point	Distance from $\vec{v}_1$ = **(2,1.33)**	Distance from $\vec{v}_2$ = **(3.67,3.67)**	Minimum distance	Assigned center
$\vec{x}_1$	$(1,1)$	1.05	3.77	1.05	$\vec{v}_1$
$\vec{x}_2$	$(2,1)$	0.33	3.14	0.33	$\vec{v}_1$
$\vec{x}_3$	$(2,3)$	1.67	1.80	1.67	$\vec{v}_1$
$\vec{x}_4$	$(3,2)$	1.20	1.80	1.20	$\vec{v}_1$
$\vec{x}_5$	$(4,3)$	2.60	0.75	0.75	$\vec{v}_2$
$\vec{x}_6$	$(5,5)$	4.74	1.89	1.89	$\vec{v}_2$

Cluster 1 : $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4\}$ represented by $\vec{v}_1$

Number of data points in Cluster 1: $c_1 = 4$.

Cluster 2 : $\{\vec{x}_5, \vec{x}_6\}$ represented by $\vec{v}_2$

Number of data points in Cluster 1: $c_2 = 2$.

Example-Kmeans [6/8]

- Cluster centres are recalculated

$$\vec{v}_1 = \frac{1}{c_1}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$

$$= \frac{1}{4}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$

$$= (2.00, 1.75)$$

$$\vec{v}_2 = \frac{1}{c_2}(\vec{x}_5 + \vec{x}_6)$$

$$= \frac{1}{2}(\vec{x}_5 + \vec{x}_6)$$

$$= ($$ 4.5,4

Example-Kmeans [7/8]

- Compute distance of the given data points from new cluster centres [(2, 1.75) & (4.5, 4.0)]

$\vec{x}_i$	Data point	Distance from $\vec{v}_1$ =(2,1.75)	Distance from $\vec{v}_2$ = (4.5,4	Minimum distance	Assigned center
$\vec{x}_1$	$(1,1)$	1.25	4.61	1.25	$\vec{v}_1$
$\vec{x}_2$	$(2,1)$	0.75	3.91	0.75	$\vec{v}_1$
$\vec{x}_3$	$(2,3)$	1.25	2.69	1.25	$\vec{v}_1$
$\vec{x}_4$	$(3,2)$	1.03	2.50	1.03	$\vec{v}_1$
$\vec{x}_5$	$(4,3)$	2.36	1.12	1.12	$\vec{v}_2$
$\vec{x}_6$	$(5,5)$	4.42	1.12	1.12	$\vec{v}_2$

Cluster 1 : $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4\}$ represented by $\vec{v}_1$

Cluster 2 : $\{\vec{x}_5, \vec{x}_6\}$ represented by $\vec{v}_2$

Example-Kmeans [8/8]

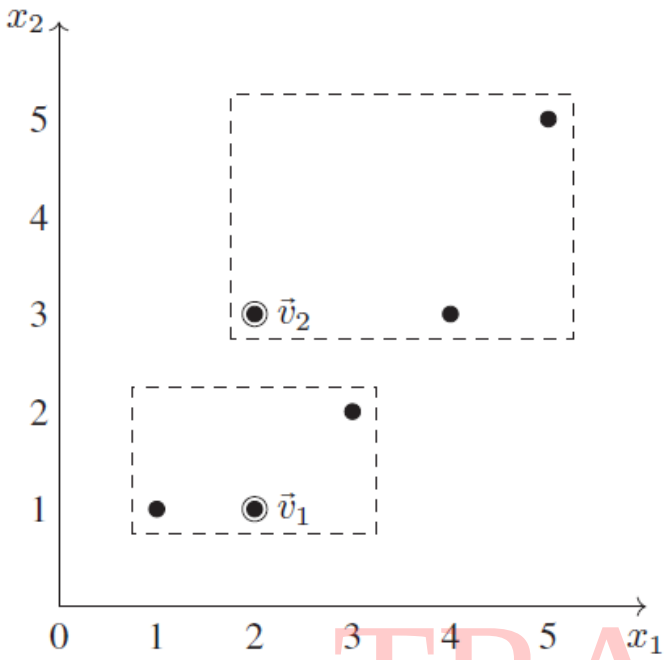
- Cluster centres are recalculated

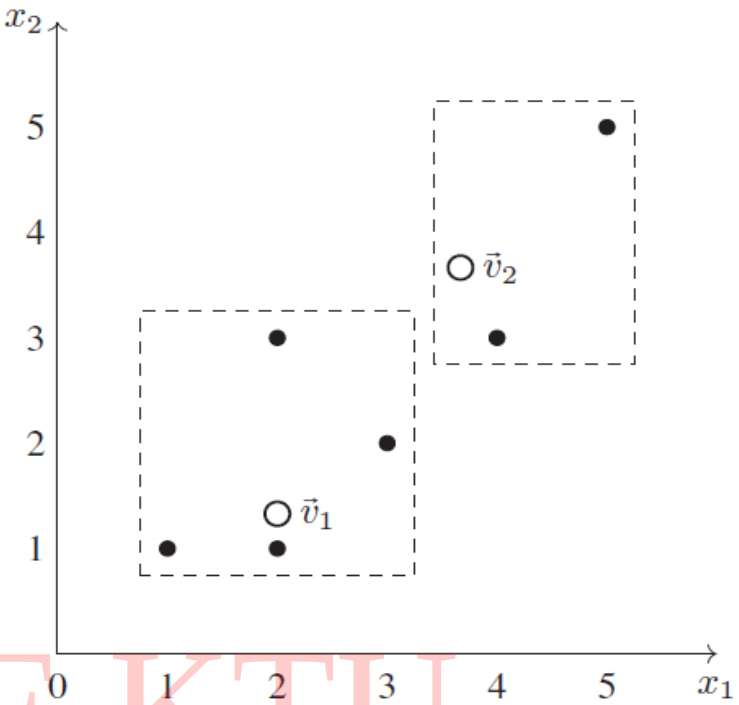
$$\vec{v}_1 = \frac{1}{c_1}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$
$$= \frac{1}{4}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \vec{x}_4)$$
$$= (2.00, 1.75)$$
$$\vec{v}_2 = \frac{1}{c_2}(\vec{x}_5 + \vec{x}_6)$$
$$= \frac{1}{2}(\vec{x}_5 + \vec{x}_6)$$
$$= \boxed{4.5,4}$$

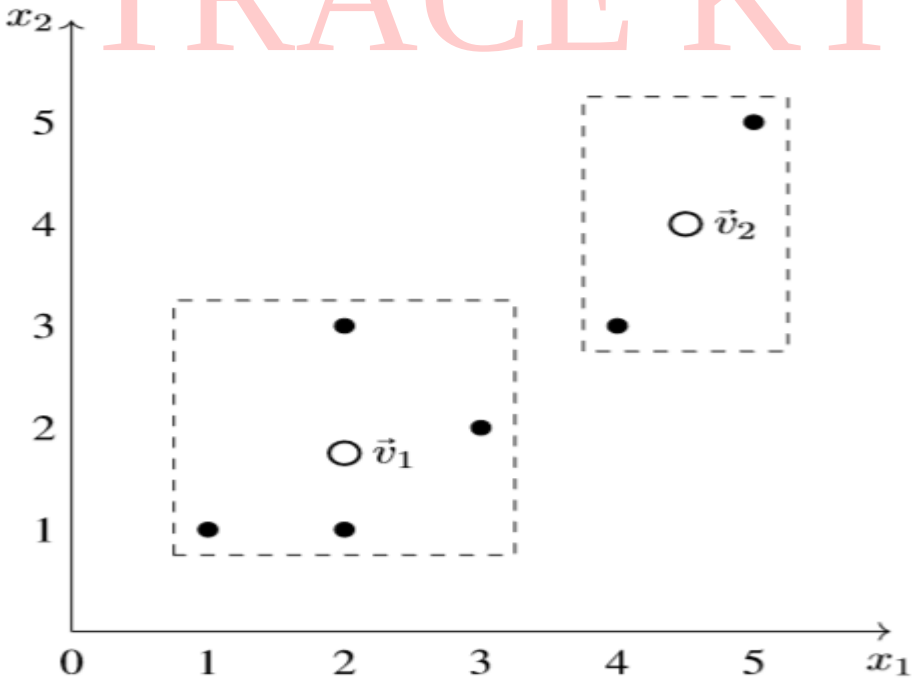
- Finally Clusters are:

Cluster 1 : $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4\}$ represented by $\vec{v}_1 = (2.00, 1.75)$

Cluster 2 : $\{\vec{x}_5, \vec{x}_6\}$ represented by $\vec{v}_2 =$ (4.5,4)


x_2
5
4
3
2
1
$\vec{v}_2$
$\vec{v}_1$
0
1
2
3
4
5
x_1


x_2
5
4
3
2
1
$\vec{v}_2$
$\vec{v}_1$
0
1
2
3
4
5
x_1


x_2
5
4
3
2
1
$\vec{v}_2$
$\vec{v}_1$
0
1
2
3
4
5
x_1

Algorithm-Kmeans

- Basic Idea:

The algorithm aims to find a partition the set X into k mutually disjoint subsets $S = \{S_1, S_2, ..., S_k\}$ and a set of data points V which minimizes the following within-cluster sum of errors:

$$\sum_{i=1}^{k} \sum_{\vec{x} \in S_i} \|\vec{x} - \vec{v}_i\|^2$$

Algorithm-Kmeans

Step 1. Randomly select k cluster centers $\vec{v}_1, \ldots, \vec{v}_k$.

Step 2. Calculate the distance between each data point $\vec{x}_i$ and each cluster center $\vec{v}_j$.

Step 3. For each $j = 1, 2, \ldots, N$, assign the data point $\vec{x}_j$ to the cluster center $\vec{v}_i$ for which the distance $\|\vec{x}_j - \vec{v}_i\|$ is minimum. Let $\vec{x}_{i1}, \vec{x}_{i2}, \ldots, \vec{x}_{ic_i}$ be the data points assigned to $\vec{v}_i$.

Step 4. Recalculate the cluster centres using

$$\vec{v}_i = \frac{1}{c_i}(\vec{x}_{i1} + \cdots + \vec{x}_{ic_i}), \quad i = 1, 2, \ldots, k.$$

Step 5. Recalculate the distance between each data point and newly obtained cluster centers.

Step 6. If no data point was reassigned then stop. Otherwise repeat from Step 3.

- The following are some of the methods for choosing the initial vi's.
  - Randomly take some k data points as the initial vi's.
  - Calculate the mean of all data and add small random vectors to the mean to get the k initial vi's.
  - Calculate the principal component, divide its range into k equal intervals,and then take the means of these groups as the initial centres.

Disadvantages

- Requires apriori specification of the number of cluster centers.
- The final custer centres depend on the initial cluster centre chosen.
- Hard assignment of data points to clusters-there are situations when same data point may belong to two/more clusters.
- Applicable only when mean is defined i.e. fails for categorical data.
- Application:
  - Image Segmentation
  - Data Compression

- Use K Means clustering to cluster the following data into two groups. Assume cluster centroid are m1=2 and m2=4. The distance function used is Euclidean distance. { 2, 4, 10, 12, 3, 20, 30, 11, 25 }.

3. Suppose you want to cluster the eight points shown below using **k**-means

	A_1	A_2
x_1	2	10
x_2	2	5
x_3	8	4
x_4	5	8
x_5	7	5
x_6	6	4
x_7	1	2
x_8	4	9

Assume that **k = 3** and that initially the points are assigned to clusters as follows:

$C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_4, x_5, x_6\}$, $C_3 = \{x_7, x_8\}$. Apply the **k**-means algorithm until convergence, using the Manhattan distance.

C1=(2+2+8)/3,(10+5+4)/3
(4,6.33)
C2=
C3=

Another Question:
Suppose:Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2

Given Points	Distance from center (2, 10) of Cluster-01	Distance from center (5, 8) of Cluster-02	Distance from center (1, 2) of Cluster-03	Point belongs to Cluster
A1(2, 10)	0	5	9	C1
A2(2, 5)	5	6	4	C3
A3(8, 4)	12	7	9	C2
A4(5, 8)	5	0	10	C2
A5(7, 5)	10	5	9	C2
A6(6, 4)	10	5	7	C2
A7(1, 2)	9	10	0	C3
A8(4, 9)	3	2	10	C2

Hierarchical clustering

- is a method of cluster analysis which seeks to build a hierarchy of clusters (or groups) in a given dataset.
- At the lowest level, each cluster contains a single observation. At the highest level there is only one cluster containing all of the data.
- The decision regarding whether two clusters are to be merged or not is taken based on the measure of dissimilarity between the clusters.

Hierarchical Clustering

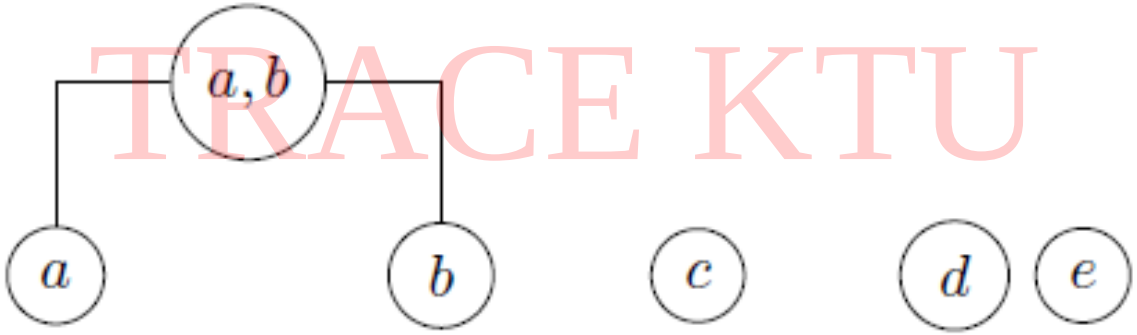
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits.
  - The dendrogram may be drawn with the root node at the top and the branches growing vertically downwards.
  - It may also be drawn with the root node at the left and the branches growing horizontally rightwards

Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative: (bottom up)
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left.
    - N observations-N-1 hierarchy
  - Divisive: (top down method) *DIANA* (*DIvisive ANAlysis*)
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains an individual point (or there are k clusters)

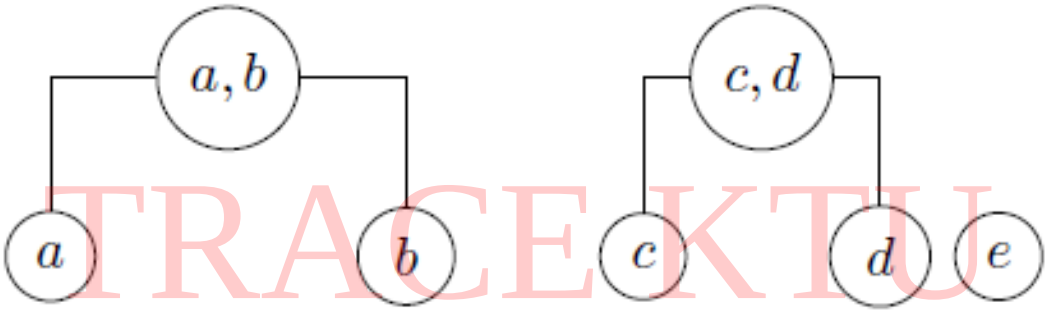
Agglomerative Clustering [1/5]

Step 1



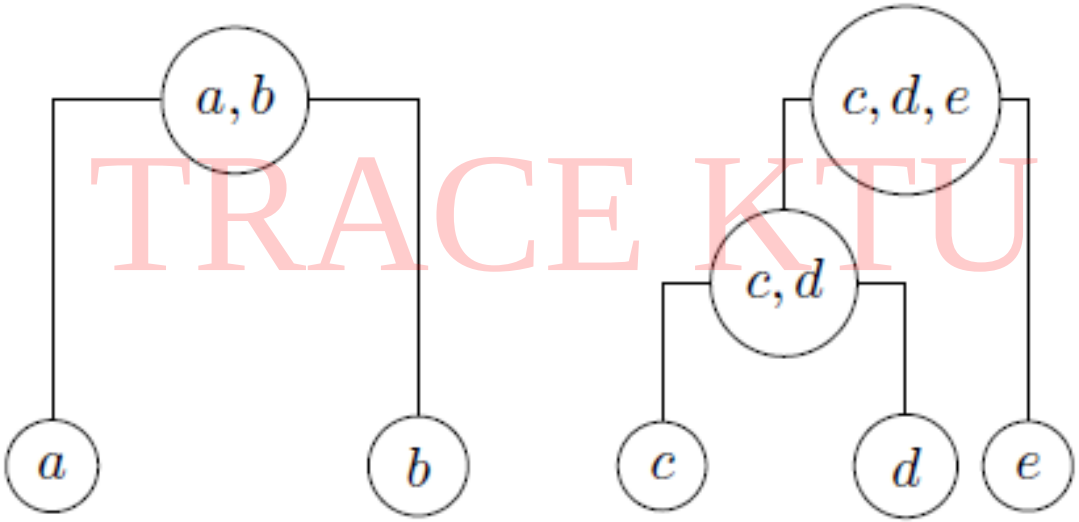
Step 2

Agglomerative Clustering [2/5]



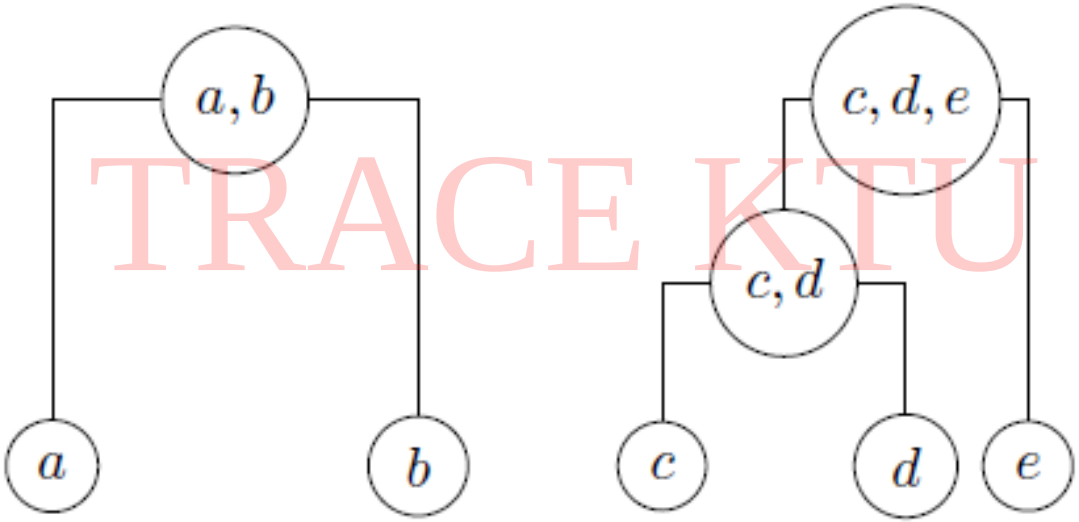
Step 3

Agglomerative Clustering [3/5]



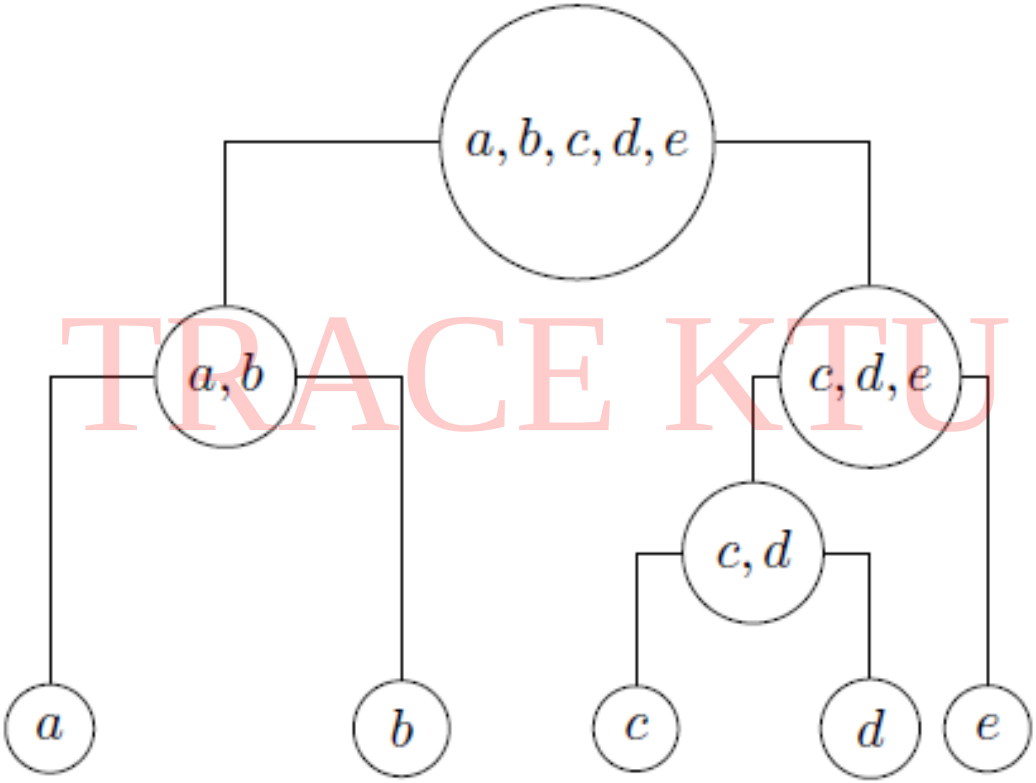
Step 4

Agglomerative Clustering [4/5]



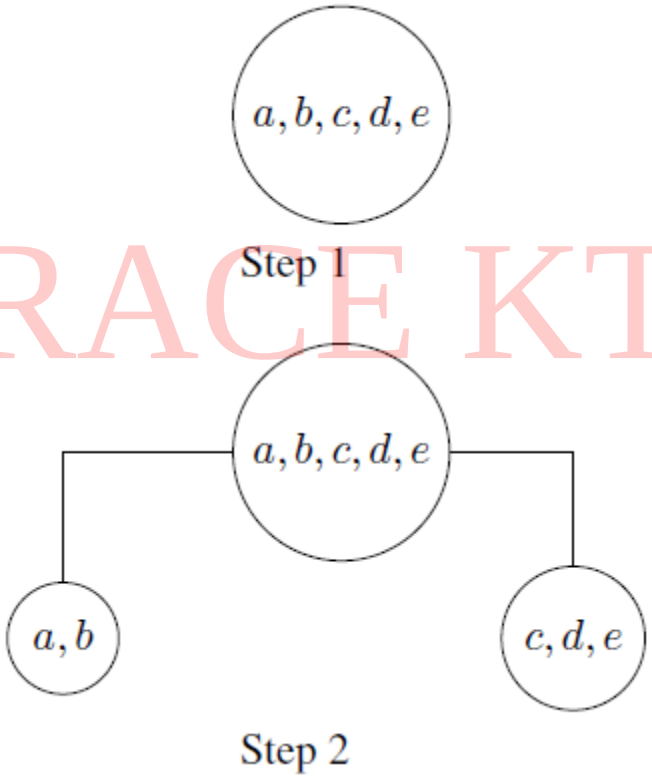
Step 4

Agglomerative Clustering [5/5]

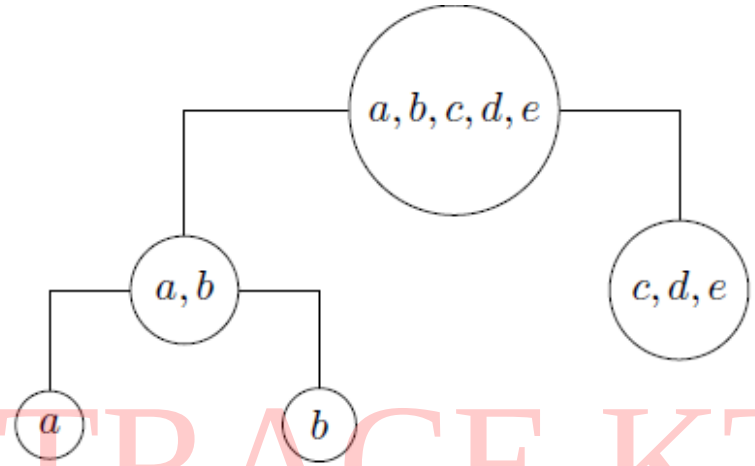


Step 5

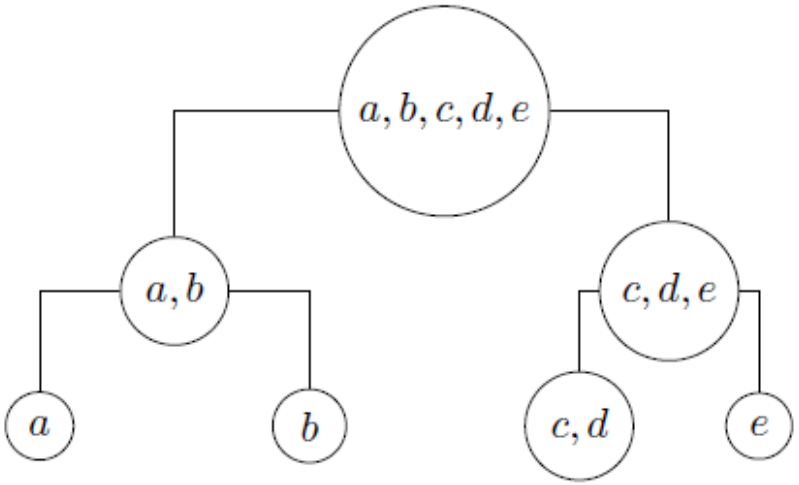
Divisive Method



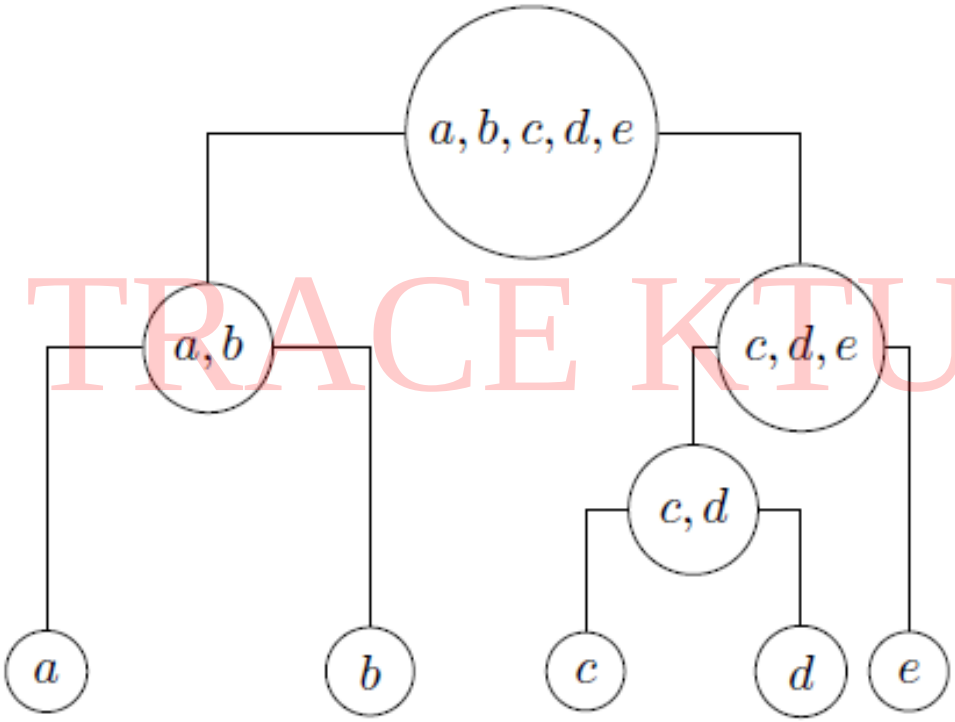
Divisive Method



Step 3



Divisive Method



Step 5

Measures of distance between groups of datapoints

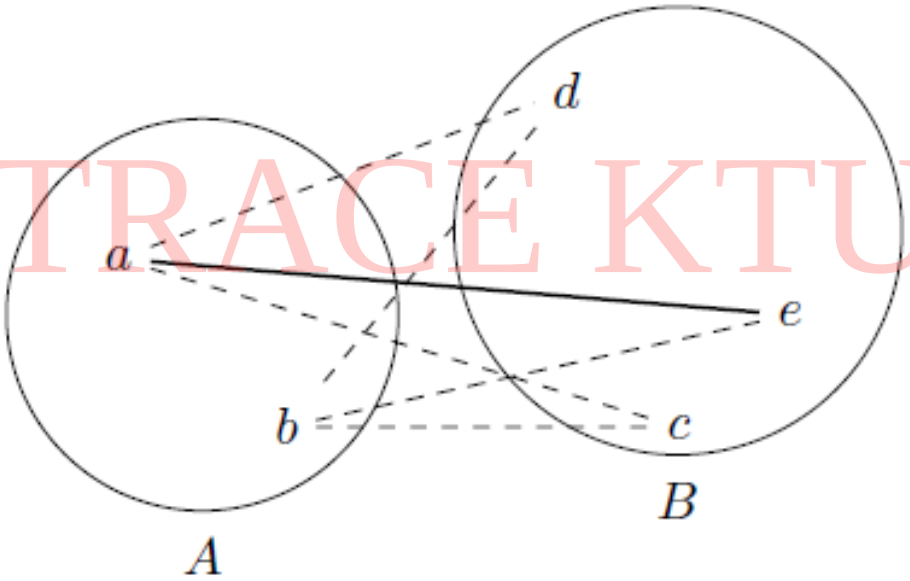
- Let A and B be two groups of observations .
- Let x and y be arbitrary data points in A and B respectively.
- Suppose we have chosen some formula, say Euclidean distance formula, to measure the distance between data points.
- Let d(x,y)denote the distance between x and y.
- d(A,B) the distance between the groups A and B.
  - The following are some of the different methods in which d(A,B)is defined.

Measures of dissimilarity

- In order to decide which clusters should be combined(for agglomerative),or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required.
- In most methods of hierarchical clustering,
  - measure of distance between data points.
  - The distance between two groups of observations.

Measure of distance between groups of data points

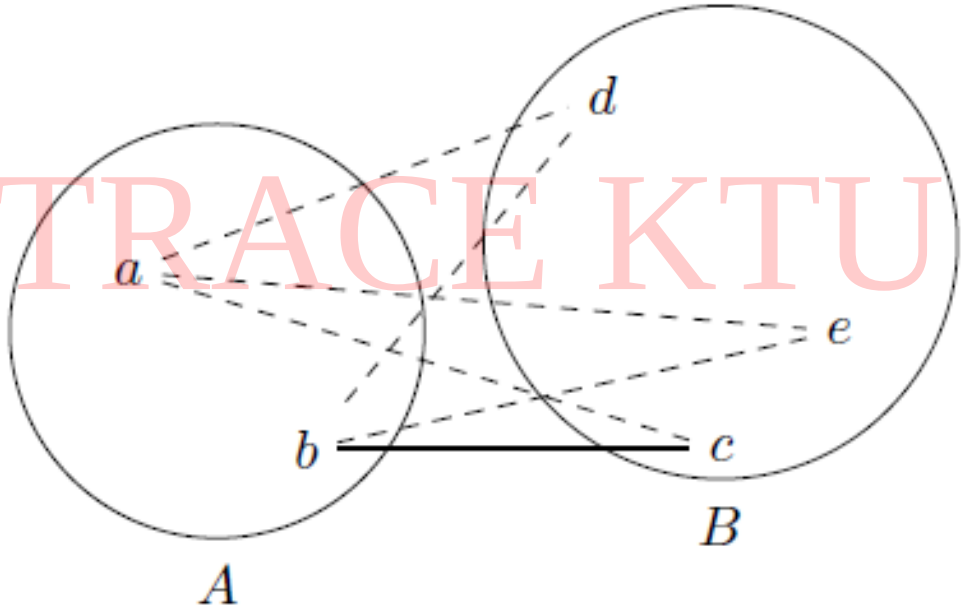
- Complete Clustering Linkage (*farthest neighbour clustering*)



$$d(A, B) = \max\{d(x, y) : x \in A, y \in B\}$$

Measure of distance between groups of data points

- Single Linkage Clustering (*Nearest neighbour clustering*)



$$d(A, B) = \min\{d(x, y) : x \in A, y \in B\}$$

Measure of distance between groups of data points

- Average Linkage Clustering (*Mean/Average clustering*)

$$d(A,B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x,y)$$ where $|A|$, $|B|$ are respectively the number of elements in A and B.

Algorithm-Agglomerative Clustering

Step 1. Start by assigning each item to its own cluster, so that we have *N* clusters.

Step 2. Find the closest pair of clusters and merge them into a single cluster, so that now we have one less cluster.

Step 3. Compute distances between the new cluster and each of the old clusters.

Step 4. Repeat Steps 2 and 3 until all items are clustered into a single cluster of size *N*.

Example

Given the dataset $\{a, b, c, d, e\}$ and the following distance matrix, construct a dendrogram by complete-linkage hierarchical clustering using the agglomerative method.

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

Example

Initial clustering (singleton sets) C_1: $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}$.

The following table gives the distances between the various clusters in C_1

	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{e\}$
$\{a\}$	0	9	3	6	11
$\{b\}$	9	0	7	5	10
$\{c\}$	3	7	0	9	**2**
$\{d\}$	6	5	9	0	8
$\{e\}$	11	10	**2**	8	0

- Minimum distance is the distance the clusters {c},{e}

Example

The new set of clusters C_2: $\{a\}, \{b\}, \{d\}, \{c, e\}$.

	$\{a\}$	$\{b\}$	$\{d\}$	$\{c, e\}$
$\{a\}$	0	9	6	11
$\{b\}$	9	0	**5**	10
$\{d\}$	6	**5**	0	9
$\{c, e\}$	11	10	9	0

Let us compute the distance of $\{c, e\}$ from other clusters.

$$d(\{c, e\}, \{a\}) = \max\{d(c, a), d(e, a)\} = \max\{3, 11\} = 11.$$

$$d(\{c, e\}, \{b\}) = \max\{d(c, b), d(e, b)\} = \max\{7, 10\} = 10.$$

$$d(\{c, e\}, \{d\}) = \max\{d(c, d), d(e, d)\} = \max\{9, 8\} = 9.$$

Example

The new set of clusters C_3: $\{a\}, \{b,d\}, \{c,e\}$

The following table gives the distances between the various clusters in C_3

	$\{a\}$	$\{b,d\}$	$\{c,e\}$
$\{a\}$	0	**9**	11
$\{b,d\}$	**9**	0	10
$\{c,e\}$	11	10	0

Let us compute the distance of $\{b,d\}$ from other clusters.

$$d(\{b,d\},\{a\}) = \max\{d(b,a), d(d,a)\} = \max\{9,6\} = 9.$$

$$d(\{b,d\},\{c,e\}) = \max\{d(b,c), d(b,e), d(d,c), d(d,e)\} = \max\{7,10,9,8\} = 10.$$

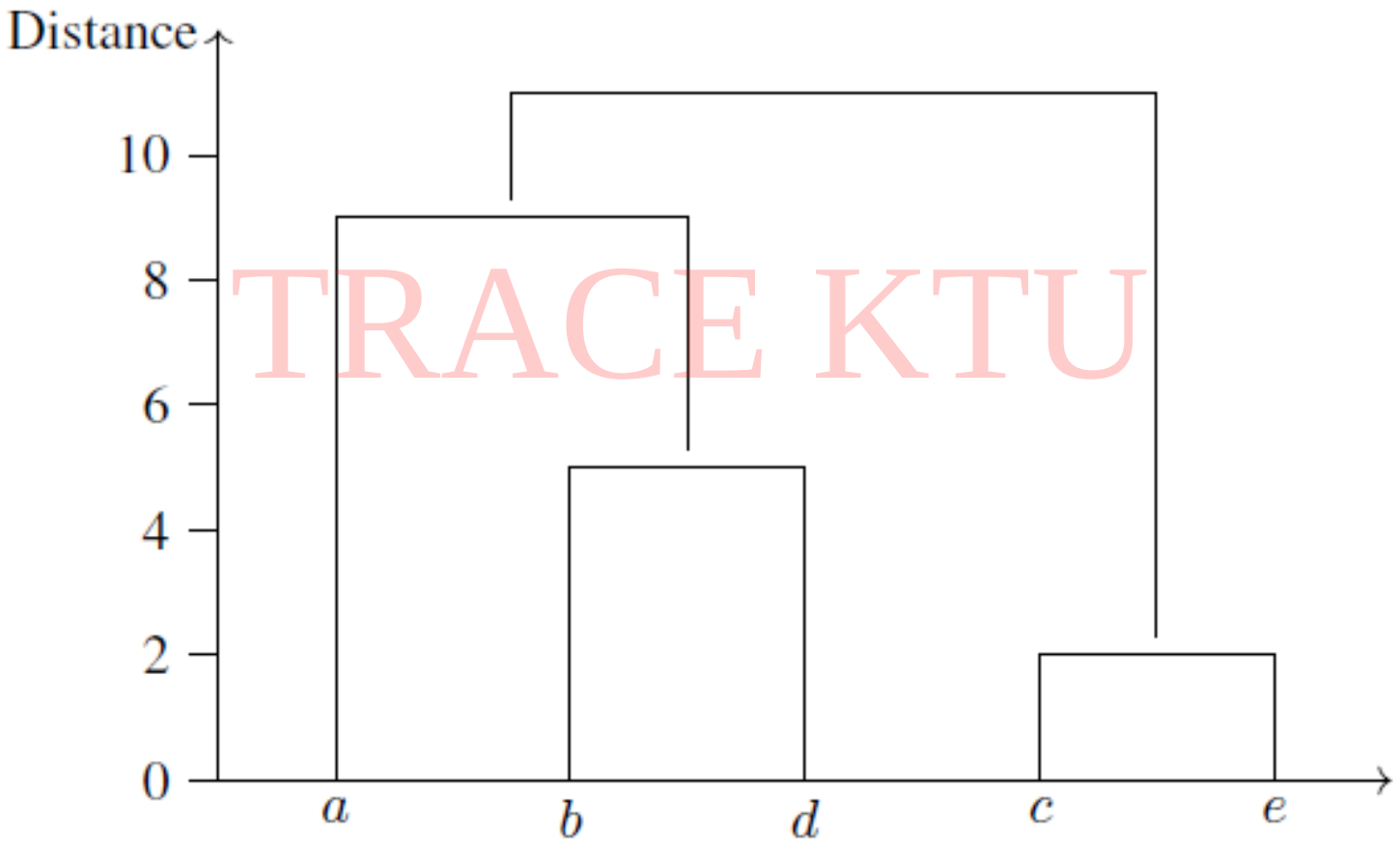
Example

The new set of clusters C_4: $\{a, b, d\}$, $\{c, e\}$

$$d(\{a,b,d\},\{c,e\}) = \max\{d(a,c), d(a,e), d(b,c), d(b,e), d(d,c), d(d,e)\}$$
$$= \max\{3, 11, 7, 10, 9, 8\}$$
$$= 11$$

Final cluster={a,b,d,c,e}

Example-Dendrogram



Examples:

Using Single Linkage

b) For the given data points, construct the dendrogram using C method.

	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	0.30

U can write all the values

ANSWER OF A:FIRST FIND THE DISTANCE TO EACH POINT:

- Calculate Euclidean distance, create the distance matrix.

$$\text{Distance } [(x,y), (a,b)] = \sqrt{(x-a)^2+(x-b)^2}$$

Distance (P1,P2) $= \sqrt{(0.40-0.22)^2+(0.53-0.38)^2}$

(0.40,0.53), (0.22,0.38) $= \sqrt{(0.18)^2+(0.15)^2}$

$= \sqrt{0.0324+0.0225}$

$= \sqrt{0.0549}$

$= 0.23$

- The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

- The updated distance matrix for cluster P3, P6

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

- The distance matrix is

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

- The updated distance matrix for cluster P2,P5

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

he distance matrix is

Select any of .15 value

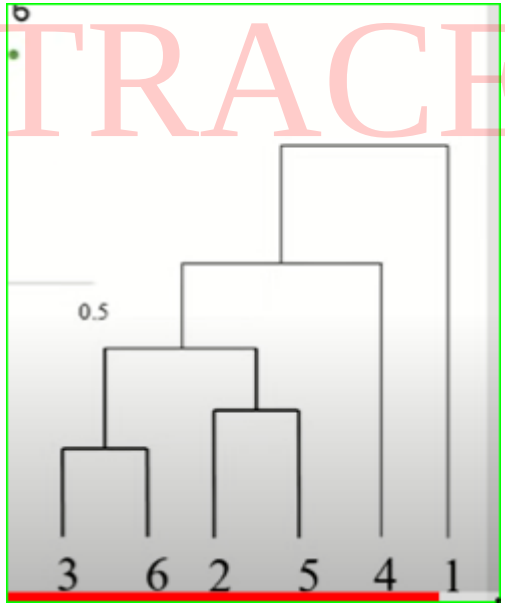
	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

- The updated distance matrix for cluster P2,P5,P3,P6

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

- The updated distance matrix for cluster P2,P5,P3,P6,P4

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.22	0



Using Average linkage

- The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

½*1

- The distance matrix is, AVG[dist(P3,P6),P1]

dist((P3,P6),P1) = ½ (dist(P3,P1)+dist(P6,P1)

= ½ (0.22+0.23)

= ½ (0.45)

- The distance matrix is, AVG[(dist(P3,P6),P2]

dist((P3,P6),P2) = ½ (dist(P3,P2)+dist(P6,P2)

= ½ (0.15+0.25)

= ½ (0.4)

= 0.2

- The distance matrix is, AVG[dist(P3,P6),P4]

dist((P3,P6),P4) = ½ (dist(P3,P4)+dist(P6,P4)

= ½ (0.15+0.22)

= ½ (0.37)

= 0.19

- The distance matrix is, AVG[dist(P3,P6),P5]

dist((P3,P6),P5) = ½ (dist(P3,P5)+dist(P6,P5)

= ½ (0.28+0.39)

= ½ (0.67)

= 0.34

The updated distance matrix for cluster (P3,P6)

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.23	0.2	0		
P4	0.37	0.20	0.19	0	
P5	0.34	0.14	0.34	0.29	0

- The distance matrix is,

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.23	0.2	0		
P4	0.37	0.20	0.19	0	
P5	0.34	0.14	0.34	0.29	0

To update the distance matrix , AVG(dist(P2,P5),P1

dist((P2,P5),P1) = ½ (dist(P2,P1)+dist(P5,P1)

= ½ (0.23+0.34)

= ½ (0.57)

= 0.29

- To update the distance matrix , AVG(dist(P2,P5),(P3,P6))

dist((P2,P5),(P3,P6)) = ½ (dist(P2,(P3,P6))+dist(P5,(P3,P6))

= ½ (0.2+0.34)

= ½ (0.54)

= 0.27

Can do like this also
(1/4(.15+.25+.28+.39)

To update the distance matrix , AVG(dist(P2,P5),P4)

dist((P2,P5),(P4)) = ½ (dist(P2,P4)+dist(P5,P4)

= ½ (0.20+0.29)

= ½ (0.49)

= 0.25

- The updated distance matrix for cluster (P2,P5)

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.29	0		
P3,P6	0.23	0.27	0	
P4	0.37	0.25	0.19	0

The distance matrix is, ½ dist(P2,P5)

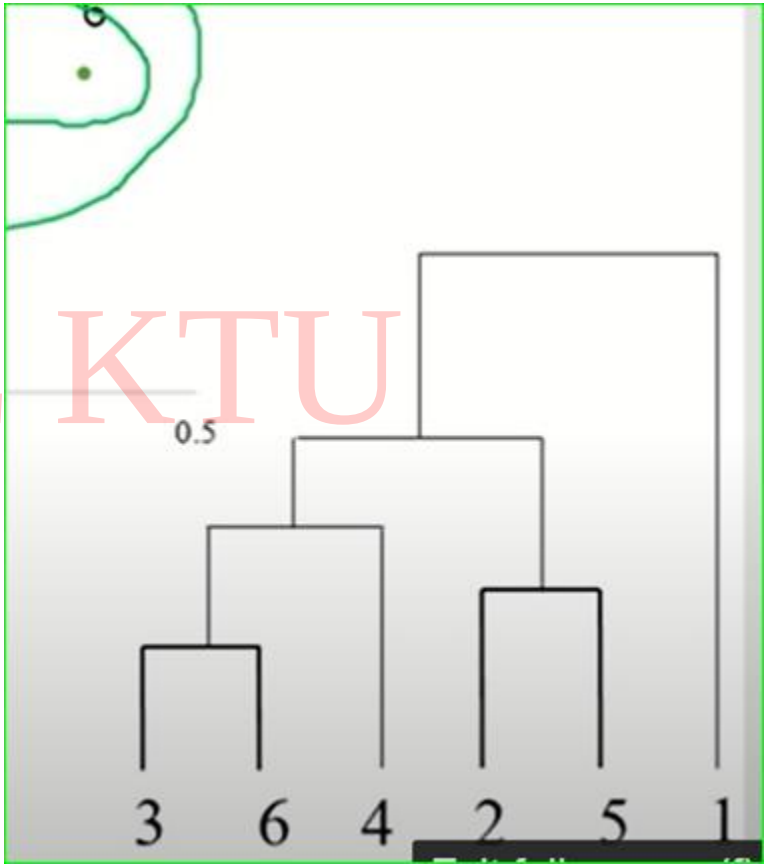
	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.29	0		
P3,P6	0.23	0.27	0	
P4	0.37	0.25	0.19	0

The distance matrix is,

	P1	P2,P5	P3,P6,P4
P1	0		
P2,P5	0.29	0	
P3,P6,P4	0.3	0.26	0

The distance matrix is,

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.3	0



(a) Suppose that we have the following data (one variable). Use single linkage Agglomerative clustering to identify the clusters.

Data: (2, 5, 9, 15, 16, 18, 25, 33, 33, 45).

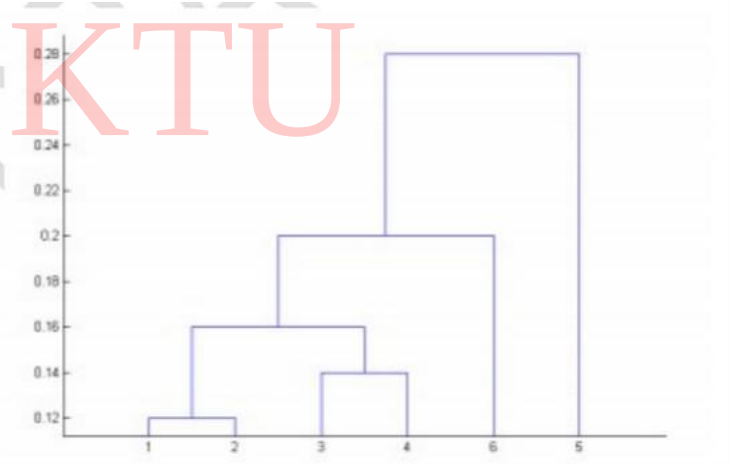
Given the following distance matrix, construct the dendogram using single linkage, complete linkage and average linkage clustering algorithm.

Item	A	B	C	D	E
A	0	2	3	3	4
B	2	0	3	5	4
C	3	3	0	2	6
D	3	5	2	0	4
E	4	4	6	4	0

****

Show the final result of hierarchical clustering with single link by drawing a dendrogram.

	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0



Divisive Algorithm-*DIANA* (*DIvisive ANAlysis*)

Step 1. Suppose that cluster C_l is going to be split into clusters C_i and C_j.

Step 2. Let $C_i = C_l$ and $C_j = \emptyset$.

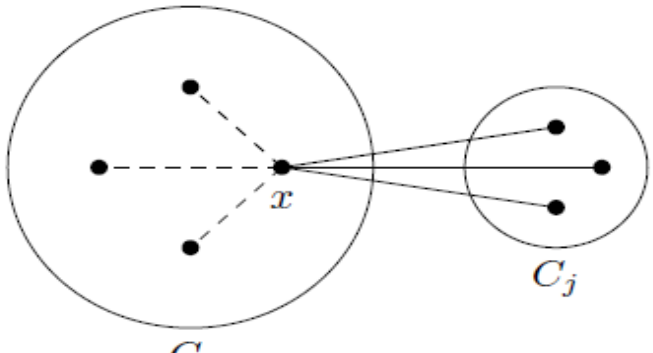
Step 3. For each object $x \in C_i$:

(a) For the first iteration, compute the average distance of x to all other objects.

Step 4. (a) For the first iteration, move the object with the maximum average distance to C_j.

(b) For the remaining iterations, find an object x in C_i for which D_x is the largest. If $D_x > 0$ then move x to C_j.

D_x = average $\{d(x, y) : y \in C_i\}$ – average$\{d(x, y) : y \in C_j\}$.



Divisive Algorithm

- Step 5. Repeat until all differences D_x are negative
- Step 6. Select the cluster with largest diameter. Again divide the cluster following Steps1-5.

- Step 7: Repeat step 6 until all clusters contain only a single object.

Divisive Algorithm-An Example

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

Average dissimilarity of a

$$= \frac{1}{4}(d(a,b) + d(a,c) + d(a,e)) = \frac{1}{4}(9+3+6+11) = 7.25$$

+d(a,d)

Similarly we have :

Average dissimilarity of $b = 7.75$

Average dissimilarity of $c = 5.25$

Average dissimilarity of $d = 7.00$

Average dissimilarity of $e = 7.75$

Divisive Algorithm-An Example

The highest average distance is 7.75 and there are two corresponding objects. We choose one of them, b, arbitrarily. We move b to C_j.

We now have

$$C_i = \{a, c, d, e\}, \quad C_j = \varnothing \cup \{b\} = \{b\}.$$

2-nd iteration.

$$D_a = \frac{1}{3}(d(a,c) + d(a,d) + d(a,e)) - \frac{1}{1}(d(a,b)) = \frac{20}{3} - 9 = -2.33$$

$$D_c = \frac{1}{3}(d(c,a) + d(c,d) + d(c,e)) - \frac{1}{1}(d(c,b)) = \frac{14}{3} - 7 = -2.33$$

$$D_d = \frac{1}{3}(d(d,a) + d(d,c) + d(d,e)) - \frac{1}{1}(d(c,b)) = \frac{23}{3} - 7 = 0.67$$

2.67

$$D_e = \frac{1}{3}(d(e,a) + d(e,c) + d(e,d)) - \frac{1}{1}(d(e,b)) = \frac{21}{3} - 7 = 0$$

-3

D_d is the largest and $D_d > 0$. So we move, d to C_j.

We now have

$$C_i = \{a, c, e\}, \quad C_j = \{b\} \cup \{d\} = \{b, d\}.$$

Divisive Algorithm-An Example

3-rd iteration

$$D_a = \frac{1}{2}(d(a,c) + d(a,e)) - \frac{1}{2}(d(a,b) + d(a,d)) = \frac{14}{2} - \frac{15}{2} = -0.5$$

$$D_c = \frac{1}{2}(d(c,a) + d(c,e)) - \frac{1}{2}(d(c,b) + d(c,d)) = \frac{5}{2} - \frac{16}{2} = -$$

-5.5

$$D_e = \frac{1}{2}(d(e,a) + d(e,c)) - \frac{1}{2}(d(e,b) + d(e,d)) = \frac{13}{2} - \frac{18}{2} = -2.5$$

All are negative. So we stop and form the clusters C_i and C_j.

C_i and C_j, we compute their diameters.

$$\begin{aligned} \text{diameter}(C_i) &= \max\{d(a,c), d(a,e), d(c,e)\} \\ &= \max\{3, 11, 2\} \\ &= 11 \\ \text{diameter}(C_j) &= \max\{d(b,d)\} \\ &= 5 \end{aligned}$$

Divisive Algorithm-An Example

The cluster with the largest diameter is C_i. So we now split C_i.

We repeat the process by taking $C_l = \{a, c, e\}$.

Expectation-maximisation algorithm

- Types of Clustering
- clustering can be divided into two subgroups :
  - **Hard Clustering:** In hard clustering, each data point either belongs to a cluster completely or not.
  - **Soft Clustering**: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.
  - Mixture Models
    - Probabilistic way of doing Soft Clustering.
    - Each cluster-Generative model(Gaussian /Multinomial)
    - EM Algorithm automatically discover all parameters for k Clusters.

Expectation-maximisation algorithm

- The expectation-maximisation algorithm (sometimes abbreviated as the EM algorithm)
  - is used to find maximum likelihood estimates of the parameters of a statistical model in cases where the equations cannot be solved directly.

- The EM Algorithm is not really an algorithm. Rather it is a general procedure to create algorithms for specific MLE problems.

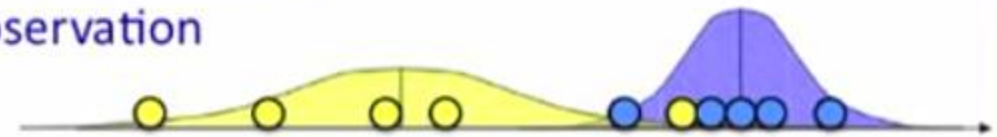
- Gaussian Mixture model is an kind of statistical model which involves latent variables and hence can’t solve using MLE.
- In machine Learning ,Clustering is an example of missing data problem..
  - ie, here the missing data are cluster labels.
  - Gaussian Mixture models can be used to cluster unlabeled data points.
  - That is, not knowing what samples came from which class,our goal is to use
    - Gaussian Mixture model to assign data points to appropriate cluster.
  - Since Gaussian Mixture model contains latent variables ,we apply EM algorithm to solve the problem

Observations $x_1 \ldots x_n$

- K=2 Gaussians with unknown μ, σ^2
- estimation trivial if we know the source of each observation

$$\mu_b = \frac{x_1 + x_2 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_1)^2 + \ldots + (x_n - \mu_n)^2}{n_b}$$



EM Algorithm

- Initialize the parameters $\theta_1, \theta_2, \ldots, \theta_k$ randomly
- Let each parameter corresponds to a cluster center (mean)
- Iterate between two steps
  - **E**xpectation step: (probabilistically) assign points to clusters
  - **M**aximation step: estimate model parameters that maximize the likelihood for the given assignment of points

- each cluster: a generative model (Gaussian or multinomial)
- parameters (e.g. mean/covariance are unknown)

- Step 1 :Initialise the parameter θ to be estimated
- Step 2:Expectation Step(E-Step):
  - Using observed available data of the dataset guess or estimate the value of missing data.
- Step 3:Maximization Step(M-Step):
  - Complete data generated after E step is used to update the parameter θ that maximizes the likelihood function
  - Step4:Repeat step 2,3 until converge

The EM algorithm for Gaussian mixtures

Problem

Suppose we are given a set of N observations

$$\{x_1, x_2, \ldots, x_N\}$$

of a numeric variable X. Let X be a mix of k normal distributions and let the probability density function of X be

$$f(x) = \pi_1 f_1(x) + \cdots + \pi_k f_k(x)$$

where

$$\pi_i \geq 0, \quad i = 1, 2, \ldots, k$$

$$\pi_i + \cdots + \pi_k = 1$$

$$f_i(x) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}, \quad i = 1, 2, \ldots, k.$$

Estimate the parameters $\mu_1, \ldots, \mu_k$, $\sigma_1, \ldots, \sigma_k$ and $\pi_1 \ldots, \pi_k$.

Log-likelihood function

Let θ denote the set of parameters $\mu_i, \sigma_i, \pi_i (i = 1, \ldots, k)$. The log-likelihood function for the abov
problem is given below:

$$\log L(\theta) = \log f(x_1) + \cdots + \log f(x_N)$$

$$= \sum_{i=1}^{N} \log \left(\frac{\pi_1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(x_i - \mu_1)^2}{2\sigma_1^2}} + \cdots + \frac{\pi_k}{\sigma_k \sqrt{2\pi}} e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \right) \qquad (13.11$$

The algorithm

The algorithm

Step 1. Initialise the means μ_i's, the variances σ_i^2's and the mixing coefficients π_i's.

Step 2. Calculate the following for $n = 1, \ldots, N$ and $i = 1, \ldots, k$:

$$\gamma_{in} = \frac{\pi_i f_i(x_n)}{\pi_1 f_1(x_n) + \cdots + \pi_k f_k(x_n)}$$

$$N_i = \gamma_{i1} + \cdots + \gamma_{iN}$$

Step 3. Recalculate the parameters using the following:

$$\mu_i^{(\text{new})} = \frac{1}{N_i}\left(\gamma_{i1}x_1 + \cdots \gamma_{iN}x_N\right)$$

$$\sigma_i^{2(\text{new})} = \frac{1}{N_i}\left(\gamma_{i1}(x_1 - \mu_i^{(\text{new})})^2 + \cdots + \gamma_{iN}(x_1 - \mu_i^{(\text{new})})^2\right)$$

$$\pi_i^{(\text{new})} = \frac{N_i}{N}$$

tep 4. Evaluate the log-likelihood function given in Eq.(13.11) and check for convergence of either the parameters or the log-likelihood function. If the convergence criterion is not satisfied, return to Step 2.

What is dimensionality reduction?

Is the process of reducing the number of variables/attributes to obtain a set of smaller number of attributes also known as *principal attributes*.

- Implementation
  - Feature Selection: Determine *k* out of *N* attributes that gives information about the data. Also, known as *subset selection*.
  - Feature Extraction: Find new set of *k* features that are the combination of *N* attributes.(2D-1)

Dimensionality Reduction

- **FEATURE SELECTION**
  - In feature selection, we are interested in finding k of the total of n features that give us the most information and we discard the other (n−k) dimensions.
  - Example: Subset selection Method
- **FEATURE EXTRACTION**
  - interested in finding a new set of k features that are the combination of the original n features.(4-features---(4 D-
  - Example: Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA)

Measures of error

- In regression problems, we may use the Mean Squared Error (MSE) or the Root Mean Squared Error (RMSE) as the measure of error.
- MSE is the sum, over all the data points, of the square of the difference between the predicted and actual target variables, divided by the number of data points. If y1,...,yn are the observed values and ˆ yi,..., ˆ yn are the predicted values, then

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- In classification problems, we may use the misclassification rate as a measure of the error. This is defined as follows:

$$\text{misclassification rate} = \frac{\text{no. of misclassified examples}}{\text{total no. of examples}}$$

Why dimensionality reduction is useful?

- for reduced memory and computation
- When an input is decided to be unnecessary, we save the cost of extracting it.(time)
- Simpler models are more robust on small datasets.
- When data can be explained with fewer features, we get a better idea about the process that underlies the data, which allows knowledge extraction.
- When data can be represented in a few dimensions without loss of information, it can be plotted and analyzed visually for structure and outliers.

Subset selection

- is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.
- feature selection technique is used when the data contains many features that are either **redundant or irrelevant**, and can thus be removed without incurring much loss of information.
- Feature selection techniques are used for four reasons:
  - It reduces the complexity of a model and makes it easier to interpret.
  - It enables the machine learning algorithm to train faster.
  - It improves the accuracy of a model if the right subset is chosen.
  - It reduces overfitting.

- Feature selection
  - Subset selection
    - forward selection and backward selection methods are two feature selection approaches

Forward selection

- In forward selection, we start with no variables and add them one by one, at each step adding the one that decreases the error the most, until any further addition does not decrease the error.

Forward selection

Procedure

We use the following notations:

n	:	number of input variables
$x_1, \ldots, x_n$	:	input variables
F_i	:	a subset of the set of input variables
$E(F_i)$	:	error incurred on the validation sample when only the inputs in F_i are used

1. Set $F_0 = \varnothing$ and $E(F_0) = \infty$.
2. For $i = 0, 1, \ldots$, repeat the following until $E(F_{i+1}) \geq E(F_i)$:
   (a) For all possible input variables x_j, train the model with the input variables $F_i \cup \{x_j\}$ and calculate $E(F_i \cup \{x_j\})$ on the validation set.
   (b) Choose that input variable x_m that causes the least error $E(F_i \cup \{x_j\})$:

$$m = \arg\min_j E(F_i \cup \{x_j\})$$

   (c) Set $F_{i+1} = F_i \cup \{x_m\}$.
3. The set F_i is outputted as the best subset.

Backward selection

- In sequential backward selection, we start with the set containing all features and at each step remove the one feature that causes the least error.(till the error become min)

Backward selection

Procedure

We use the following notations:

n	:	number of input variables
$x_1, \ldots, x_n$	:	input variables
F_i	:	a subset of the set of input variables
$E(F_i)$	:	error incurred on the validation sample when only the inputs in F_i are used

1. Set $F_0 = \{x_1, \ldots, x_n\}$ and $E(F_0) = \infty$.
2. For $i = 0, 1, \ldots$, repeat the following until $E(F_{i+1}) \geq E(F_i)$:
   (a) For all possible input variables x_j, train the model with the input variables $F_i - \{x_j\}$) and calculate $E(F_i - \{x_j\})$ on the validation set.
   (b) Choose that input variable x_m that causes the least error $E(F_i - \{x_j\})$:

$$m = \arg\min_j E(F_i - \{x_j\})$$

   (c) Set $F_{i+1} = F_i - \{x_m\}$.
3. The set F_i is outputted as the best subset.

What is dimensionality reduction?

Is the process of reducing the number of variables/attributes to obtain a set of smaller number of attributes also known as *principal attributes*.

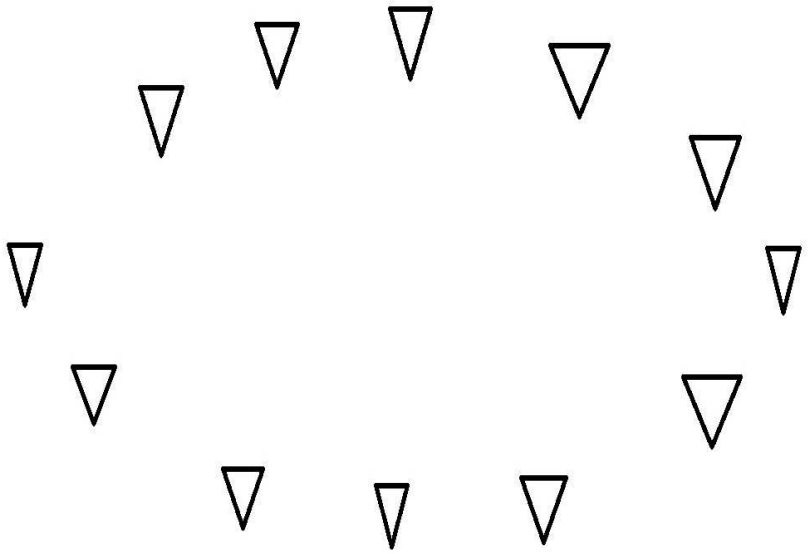
- Implementation
  - Feature Selection: Determine *k* out of *N* attributes that gives information about the data. Also, known as *subset selection*.
  - Feature Extraction: Find new set of *k* features that are the combination of *N* attributes.(2D-1)

Feature Extraction

PCA

- **Principal Components Analysis (PCA)** is a technique that can be used to simplify a dataset .
- It is a linear transformation that chooses a new coordinate system for the data set such that
  - greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component),
  - the second greatest variance on the second axis, and so on.
- PCA can be used for reducing dimensionality by eliminating the later principal components.

What is Principal Component Analysis?



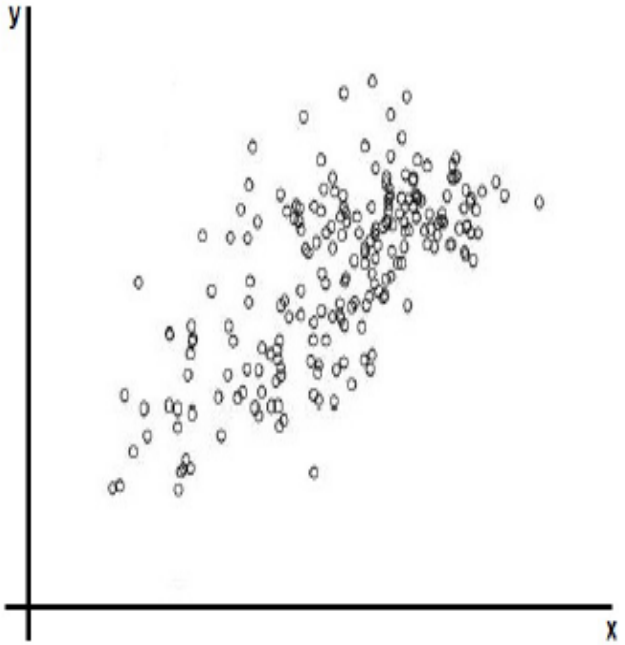
- They are the directions where there is the most variance, the directions where the data is most spread out.

Principal Component Analysis- PCA

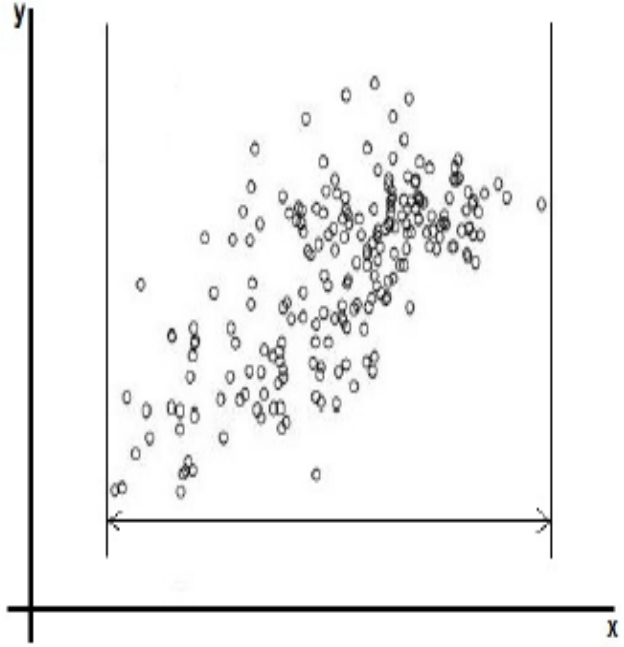
- Is a statistical procedure
- Uses orthogonal transformation to convert set of observations with correlated variables into a set of values that are linearly uncorrelated variables, also known as *principal components*.
- Number of principal components are less than or equal to the variables/observations.
- PCA Computes a new set of variables[Principal components] & express the data in terms of these new variables.
- New variables represent the same amount of information as the original variables in the sense that we can restore the original data from the transformed one..

- The first principal component has largest variance (accounts for maximum variability in the data).
- Succeeding component has highest variance with the constraint that it is orthogonal to the preceding components.

Graphical Illustaration

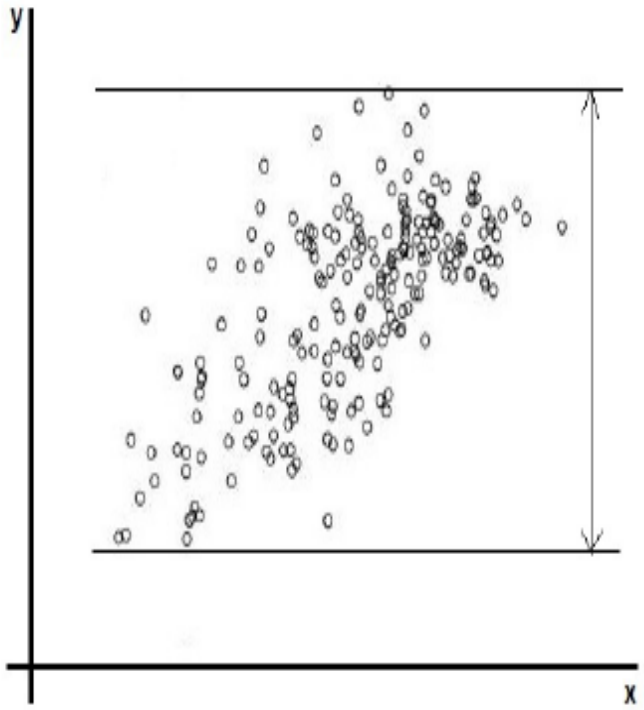


(a) Scatter diagram

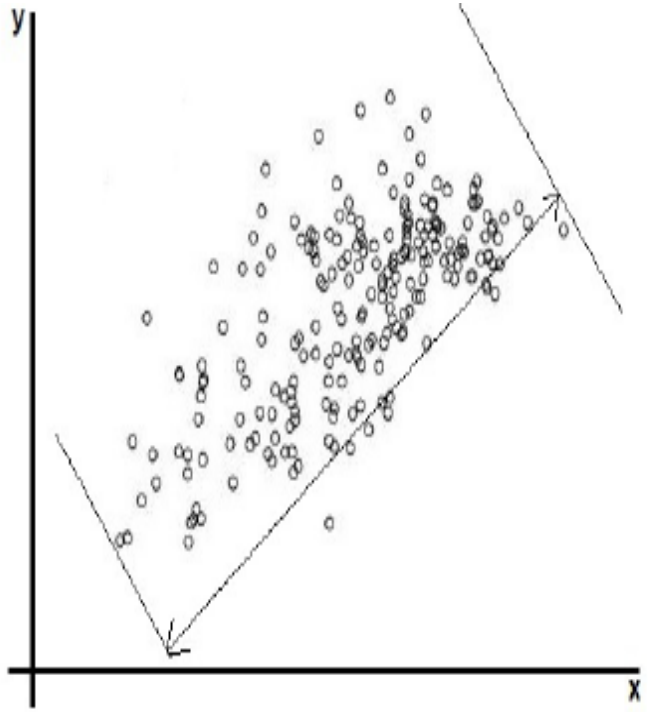


(b) Spread along x-direction

Graphical Illustaration

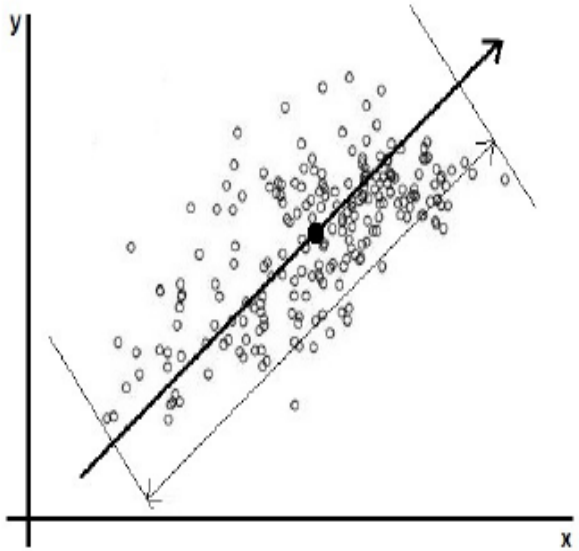


(c) Spread along y-direction

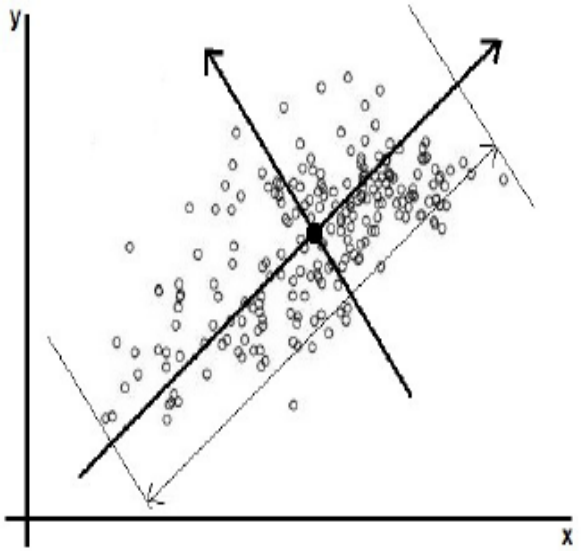


(d) Largest spread

Graphical Illustaration

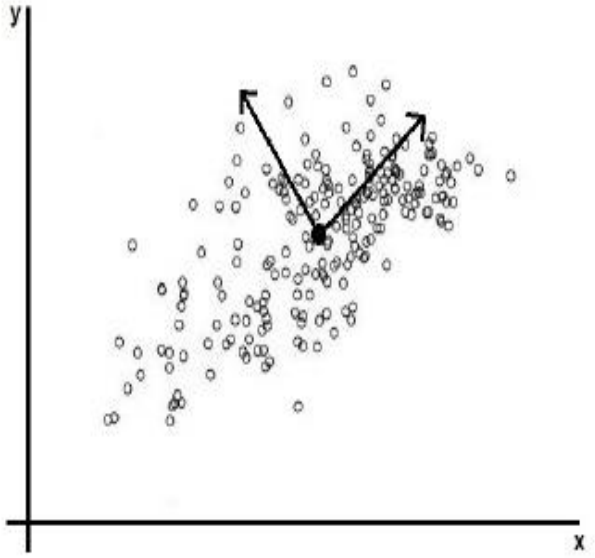


(e) Direction of largest spread : Direction of the first principal component (solid dot is the point whose coordinates are the means of x and y)



(f) Directions of principal components

Graphical Illustaration



(g) Principal component vectors (unit vectors in the directions of principal components)

How does PCA work?

- Compute the means of the variables
- the covariance matrix *X* of data points.
- Calculate *eigen vectors* and corresponding *eigen values*.
- Sort the eigen vectors according to their eigen values in decreasing order.
- Choose first *k* eigen vectors and that will be the new *k* dimensions.[direction of first PCA]
- Transform the original *n* dimensional data points into *k* dimensions.

- By finding the eigenvalues and eigenvectors of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset.
- This is the principal component.

Principal Component Vectors

Step 1: Consider the dataset having *n* features denoted by X_1, X_2, ..,X_m. Let there be *N* examples. Let the values of *i*-the feature X_i be X_{i1},X_{i2},..,X_{im}.

Example:

Features	Example 1	Example 2	⋯	Example N
X_1	X_{11}	X_{12}	⋯	X_{1N}
X_2	X_{21}	X_{22}	⋯	X_{2N}
⋮				
X_i	X_{i1}	X_{i2}	⋯	X_{iN}
⋮				
X_n	X_{n1}	X_{n2}	⋯	X_{nN}

Feature	Example 1	Example 2	Example 3	Example 4
X_1	4	8	13	7
X_2	11	4	5	14

Principal Component Vectors

Step 2: Compute the means of variables

Mean of variable X_i i.e. $\bar{X}_i$

$$\bar{X}_i = \frac{1}{N}(X_{i1} + X_{i2} + \cdots + X_{iN}).$$

Example:

$$\bar{X}_1 = \tfrac{1}{4}(4 + 8 + 13 + 7) = 8,$$
$$\bar{X}_2 = \tfrac{1}{4}(11 + 4 + 5 + 14) = 8.5.$$

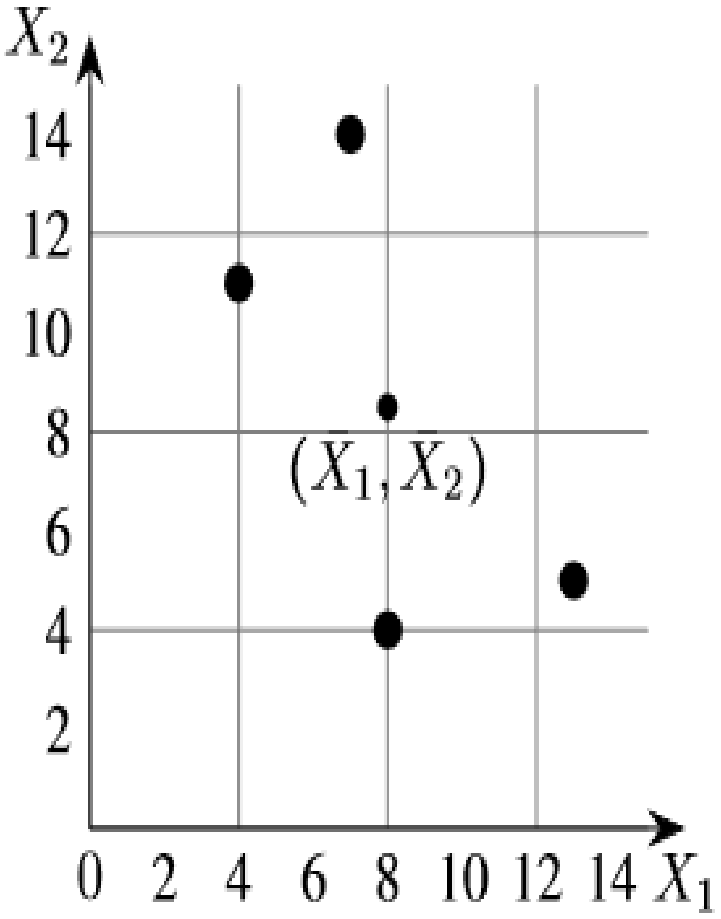
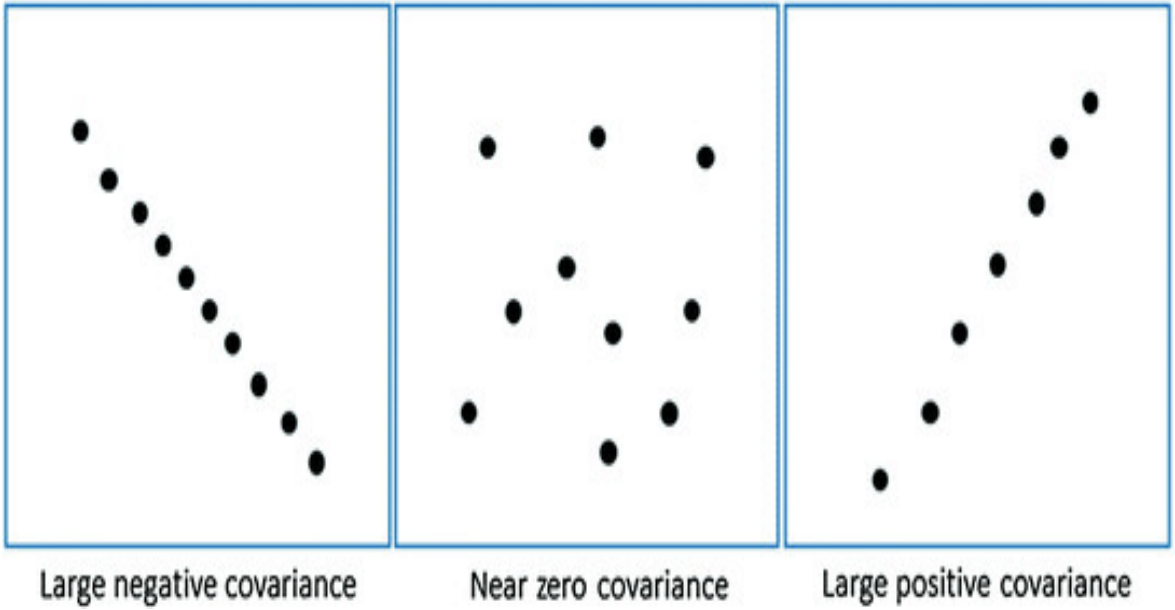


Figure 4.2: Scatter plot of data in Table 4.2

Covariance

It is a measure of the extent to which corresponding elements from two sets of ordered data move in the same direction.



Principal Component Vectors

Step 3: Calculate the covariance matrix

The covariance of the ordered pair (X_i, X_j) is defined as:

$$\operatorname{Cov}(X_i, X_j) = \frac{1}{N-1} \sum_{k=1}^{N} (X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j).$$

$$S = \begin{bmatrix} \operatorname{Cov}(X_1, X_1) & \operatorname{Cov}(X_1, X_2) & \cdots & \operatorname{Cov}(X_1, X_n) \\ \operatorname{Cov}(X_2, X_1) & \operatorname{Cov}(X_2, X_2) & \cdots & \operatorname{Cov}(X_2, X_n) \\ \vdots & & & \\ \operatorname{Cov}(X_n, X_1) & \operatorname{Cov}(X_n, X_2) & \cdots & \operatorname{Cov}(X_n, X_n) \end{bmatrix}$$

Principal Component Vectors

Step 3: Calculate the covariance matrix

Feature	Example 1	Example 2	Example 3	Example 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\mathrm{Cov}(X_i, X_j) = \frac{1}{N-1}\sum_{k=1}^{N}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j).$$

$$\mathrm{Cov}(X_1,X_1) = \tfrac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right)$$
$$= 14$$

$\mathrm{Cov}(X_1,X_2) = -11$

$\mathrm{Cov}(X_2,X_1) = -11$

$\mathrm{Cov}(X_2,X_2) = 23$

The covariance matrix is

$$S = \begin{bmatrix} \mathrm{Cov}(X_1,X_1) & \mathrm{Cov}(X_1,X_2) \\ \mathrm{Cov}(X_2,X_1) & \mathrm{Cov}(X_2,X_2) \end{bmatrix}$$
$$= \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Principal Component Vectors

Step 4: Calculate the eigen values and eigenvectors of the covariance matrix.

a) Set up equation of the form

$$\det(S - \lambda I) = 0$$

Example:

$$= \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix}$$

$$= (14-\lambda)(23-\lambda) - (-11) \times (-11)$$

$$= \lambda^2 - 37\lambda + 201$$

$\lambda_1 = 30.3849$ and $\lambda_2 = 6.6151$(desecending){eigen values}

Principal Component Vectors

Step 4: Calculate the eigen values and eigen vectors of the covariance matrix.

b) If $\lambda = \lambda'$ is an eigenvalue, the corresponding eigen vector is

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$ N*1

$$e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}.$$

such that

$$(S - \lambda' I)U = 0.$$

$= \lambda_1, \lambda_2$ (say)

4. Computation of the eigenvectors

To find the first principal components, we need only compute the eigenvector corresponding to the largest eigenvalue. In the present example, the largest eigenvalue is λ_1 and so we compute the eigenvector corresponding to λ_1.

The eigenvector corresponding to $\lambda = \lambda_1$ is a vector $U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ satisfying the following equation:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = (S - \lambda_1 I) \boxed{\text{U1}}$$

$$= \begin{bmatrix} 14 - \lambda_1 & -11 \\ -11 & 23 - \lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$= \begin{bmatrix} (14 - \lambda_1)u_1 - 11u_2 \\ -11u_1 + (23 - \lambda_1)u_2 \end{bmatrix}$$

This is equivalent to the following two equations:

$$(14 - \lambda_1)u_1 - 11u_2 = 0$$
$$-11u_1 + (23 - \lambda_1)u_2 = 0$$

Using the theory of systems of linear equations, we note that these equations are not independent and solutions are given by

$$\frac{u_1}{11} = \frac{u_2}{14 - \lambda_1} = t,$$

that is

$$u_1 = 11t, \quad u_2 = (14 - \lambda_1)t,$$

where t is any real number. Taking $t = 1$, we get an eigenvector corresponding to λ_1 as

iii) We now normalise the eigenvectors. Given any vector X we normalise it by dividing X by its length. The length (or, the norm) of the vector

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

is defined as

$$\|X\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

Given any eigenvector U, the corresponding normalised eigenvector is computed as

$$\frac{1}{\|U\|} U.$$

We compute the n normalised eigenvectors $e_1, e_2, \ldots, e_n$ by

$$e_i = \frac{1}{\|U_i\|} U_i, \quad i = 1, 2, \ldots, n.$$

where t is any real number. Taking $t = 1$, we get an eigenvector corresponding to λ_1 as

$$U_1 = \begin{bmatrix} 11 \\ 14 - \lambda_1 \end{bmatrix}.$$

To find a unit eigenvector, we compute the length of X_1 which is given by

$$\begin{aligned} \|U_1\| &= \sqrt{11^2 + (14 - \lambda_1)^2} \\ &= \sqrt{11^2 + (14 - 30.3849)^2} \\ &= 19.7348 \end{aligned}$$

Therefore, a unit eigenvector corresponding to $lambda_1$ is

$$\begin{aligned} e_1 &= \begin{bmatrix} 11/\|U_1\| \\ (14 - \lambda_1)/\|U_1\| \end{bmatrix} \\ &= \begin{bmatrix} 11/19.7348 \\ (14 - 30.3849)/19.7348 \end{bmatrix} \\ &= \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix} \end{aligned}$$

By carrying out similar computations, the unit eigenvector e_2 corresponding to the eigenvalue $\lambda = \lambda_2$ can be shown to be

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}.$$

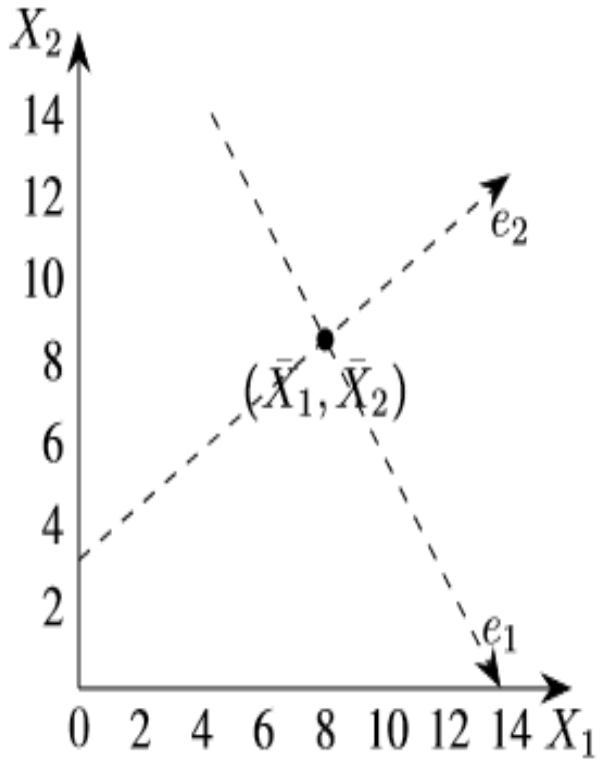


Figure 4.3: Coordinate system for principal components

Principal Component Vectors

Step 5: Determine new data set

1) Arrange the eigenvalues in the descending order i.e., $\lambda_1 > \lambda_2 > \lambda_3 > \ldots > \lambda_n$ and the corresponding eigenvectors be $e_1, e_2, e_3, \ldots, e_n$

2)Choose a positive integer *p* such that $1 \leq p \leq n$

3)Choose the eigenvectors corresponding to the eigen values $\lambda_1, \lambda_2 \ldots \lambda_p$. We write the eigenvector as row vector. Here, *T* denotes the transpose

$$F = \begin{bmatrix} e_1^T \\ e_2^T \\ \vdots \\ e_p^T \end{bmatrix},$$

Principal Component Vectors

Step 5: Determine *n* x *N* matrix.

$$X = \begin{bmatrix} X_{11} - \bar{X}_1 & X_{12} - \bar{X}_1 & \cdots & X_{1N} - \bar{X}_1 \\ X_{21} - \bar{X}_2 & X_{22} - \bar{X}_2 & \cdots & X_{2N} - \bar{X}_2 \\ \vdots & & & \\ X_{n1} - \bar{X}_n & X_{n2} - \bar{X}_n & \cdots & X_{nN} - \bar{X}_n \end{bmatrix}$$

Step 6: Next Compute the matrix

$$X_{\text{new}} = F X.$$

Transpose matrix of eigen ve

5. Computation of first principal components

Let $\begin{bmatrix} X_{1k} \\ X_{2k} \end{bmatrix}$ be the k-th sample in Table 4.2. The first principal component of this example is given by (here "T" denotes the transpose of the matrix)

$$e_1^T \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix}$$
$$= 0.5574(X_{1k} - \bar{X}_1) - 0.8303(X_{2k} - \bar{X}_2).$$

For example, the first principal component corresponding to the first example $\begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix} = \begin{bmatrix} 4 \\ 11 \end{bmatrix}$ is calculated as follows:

$$\begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} X_{11} - \bar{X}_1 \\ X_{21} - \bar{X}_2 \end{bmatrix} = 0.5574(X_{11} - \bar{X}_1) - 0.8303(X_{21} - \bar{X}_2)$$
$$= 0.5574(4 - 8) - 0.8303(11 - 8,5)$$
$$= -4.30535$$

The results of calculations are summarised in Table 4.3.

X_1	4	8	13	7
X_2	11	4	5	14
First principal components	-4.3052	3.7361	5.6928	-5.1238

2D→1D

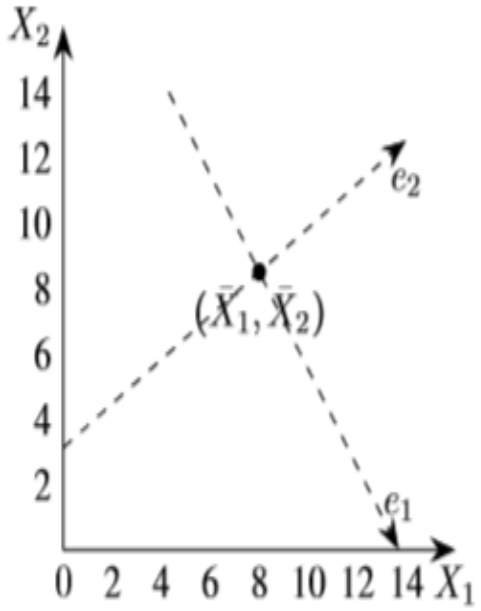
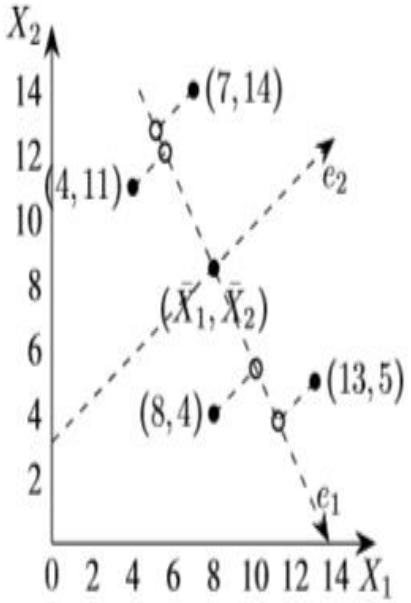


Figure 4.3: Coordinate system for principal components



jections of data points on the axis of the first principal component

ponents	-4.305187	3.736129	5.692828	-5.123769

Table 4.4: One-dimensional approximation to the data in Table 4.2

approximations can be unambiguously specified by a single number, namely, the e_1-coordinate of approximation. Thus the two-dimensional data set given in Table 4.2 can be represented approximately by the following one-dimensional data set (see Figure 4.5):

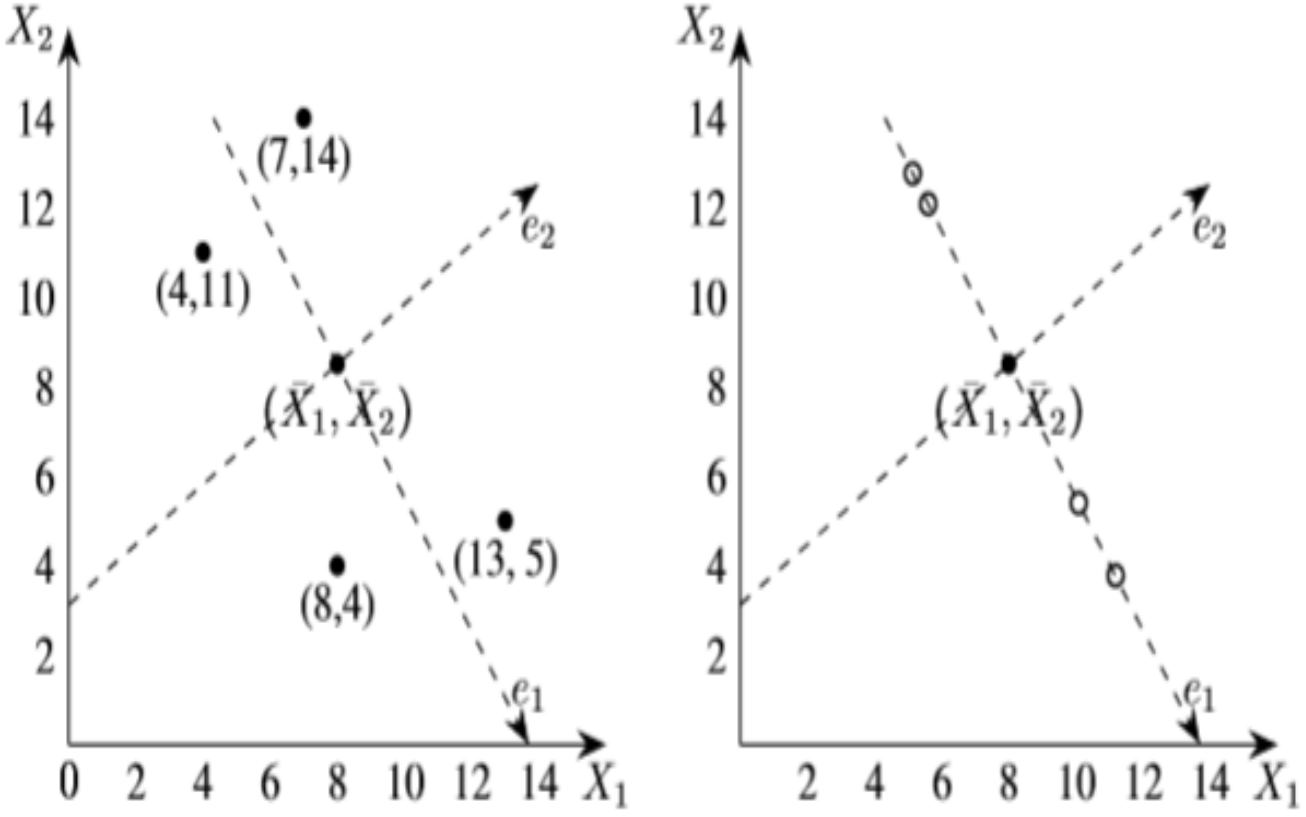


Figure 4.5: Geometrical representation of one-dimensional approximation to the data in Table 4.2

Example Applications

- Face Recognition
- Image Compression
- Pattern finding
- Gene Expression Analysis
- Data Reduction
- Data Classification
- Trend Analysis
- Factor Analysis
- Noise Reduction

Thank You