UNIT–V:

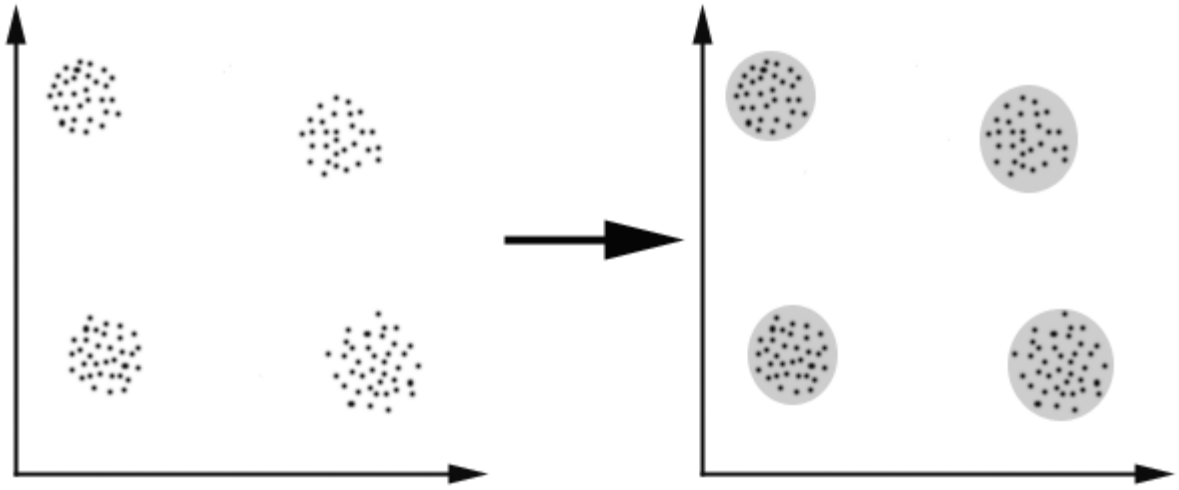
Unsupervised Learning: Introduction to clustering, Hierarchical: AGNES, DIANA, **Partitional:** K-means clustering, K-Mode Clustering, Self-Organizing Map, Expectation Maximization, Gaussian Mixture Models, Principal components analysis (PCA).

Unsupervised Learning

In some pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called *clustering*, or to determine how the data is distributed in the space, known as *density estimation*. To put forward in simpler terms, for a n-sampled space x1 to xn, true class labels are not provided for each sample, hence known as *learning without teacher*.

- Annotating large datasets is very costly and hence we can label only a few examples manually. Example: Speech Recognition
- There may be cases where we don't know how many/what classes is the data divided into. Example: Data Mining
- We may want to use clustering to gain some insight into the structure of the data before designing a classifier.

Clustering can be considered the most important *unsupervised learning* problem; so, as every other problem of this kind, it deals with finding a *structure* in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A *cluster* is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.



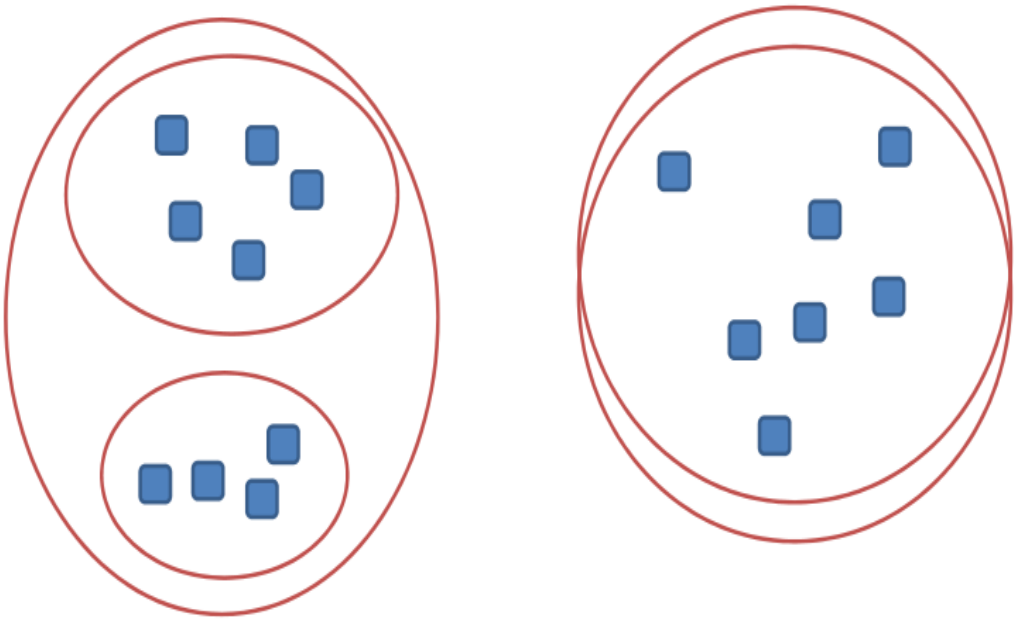
***Distance-based clustering*.**

Given a set of points, with a notion of distance between points, grouping the points into some number of *clusters*, such that

- internal (within the cluster) distances should be small i.e members of clusters are close/similar to each other.
- external (intra-cluster) distances should be large i.e. members of different clusters are dissimilar.

The Goals of Clustering

The goal of clustering is to determine the internal grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user who should supply this criterion, in such a way that the result of the clustering will suit their needs.



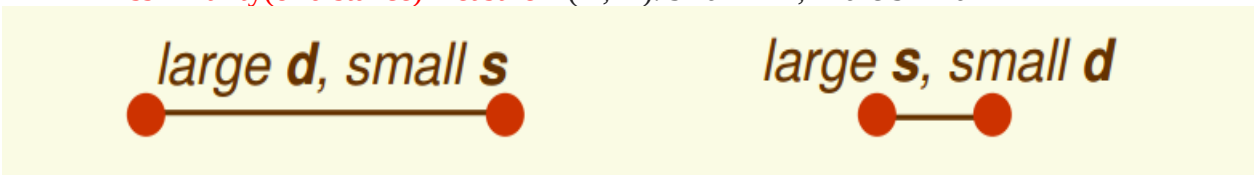
In the above image, how do we know what is the best clustering solution?

To find a particular clustering solution , we need to define the similarity measures for the clusters.

Proximity Measures

For clustering, we need to define a proximity measure for two data points. Proximity here means how similar/dissimilar the samples are with respect to each other.

- Similarity measure S(xi,xk): large if xi,xk are similar
- Dissimilarity(or distance) measure D(xi,xk): small if xi,xk are similar



There are various similarity measures which can be used.

- Vectors: Cosine Distance

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{\|\mathbf{x}\| \ \|\mathbf{x}'\|}$$

- Sets: Jaccard Distance

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

(If A and B are both empty, we define $J(A,B) = 1$.)

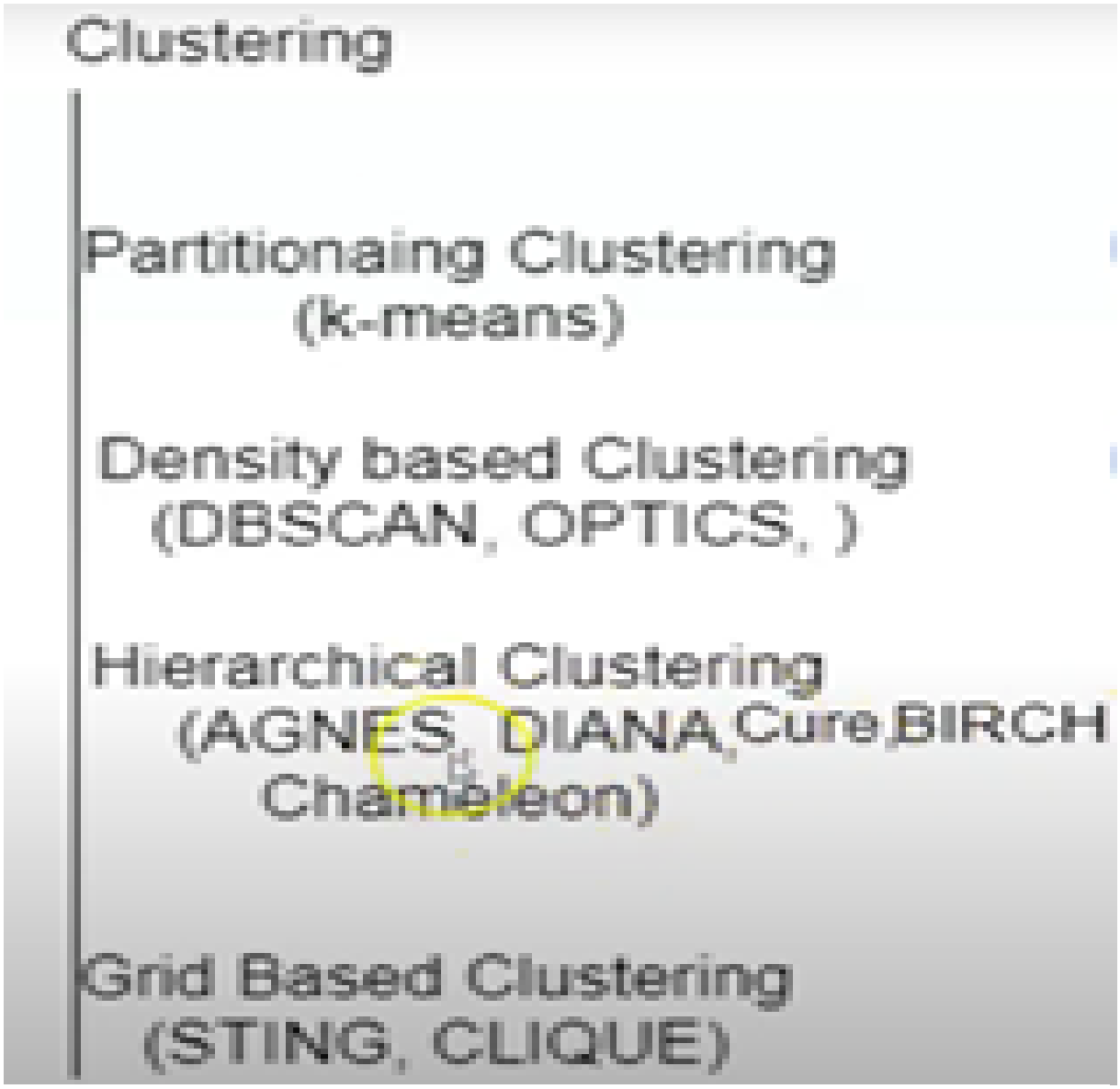
$$0 \leq J(A,B) \leq 1.$$

- Points: Euclidean Distance

  q=2

$$d(\mathbf{x}, \mathbf{x}') = \left(\sum_{k=1}^{d} |x_k - x'_k|^q \right)^{1/q},$$

Clustering Algorithms

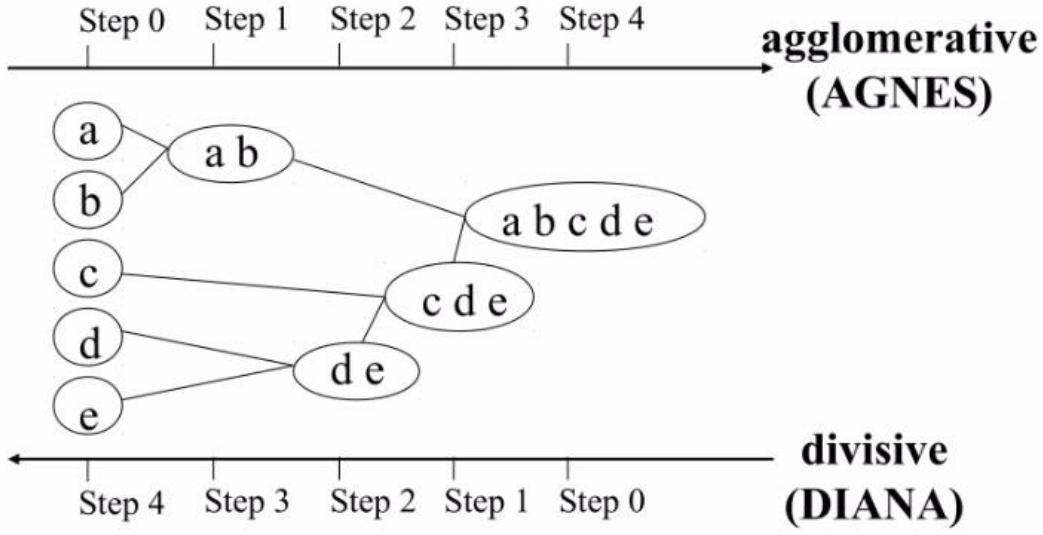


Difficulties faced in Hierarchical Clustering

- Selection of merge/split points
- Cannot revert operation
- Scalability

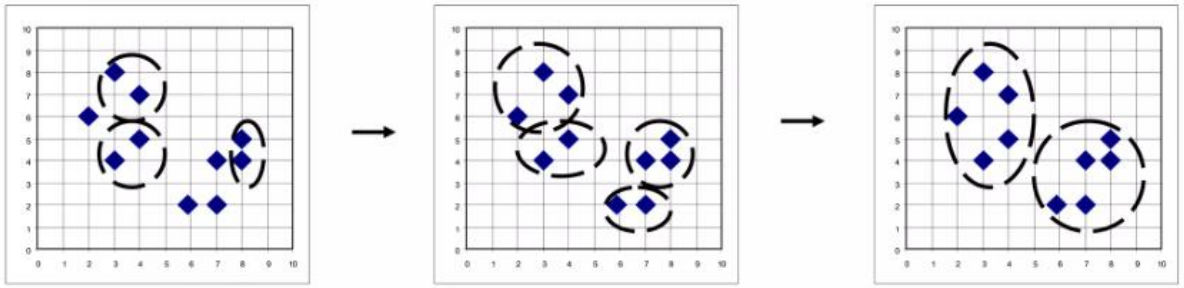
Hierarchical Clustering

- Use distance matrix as clustering criteria.
- This method does not require the number of clusters ***k*** as an input, but needs a termination condition



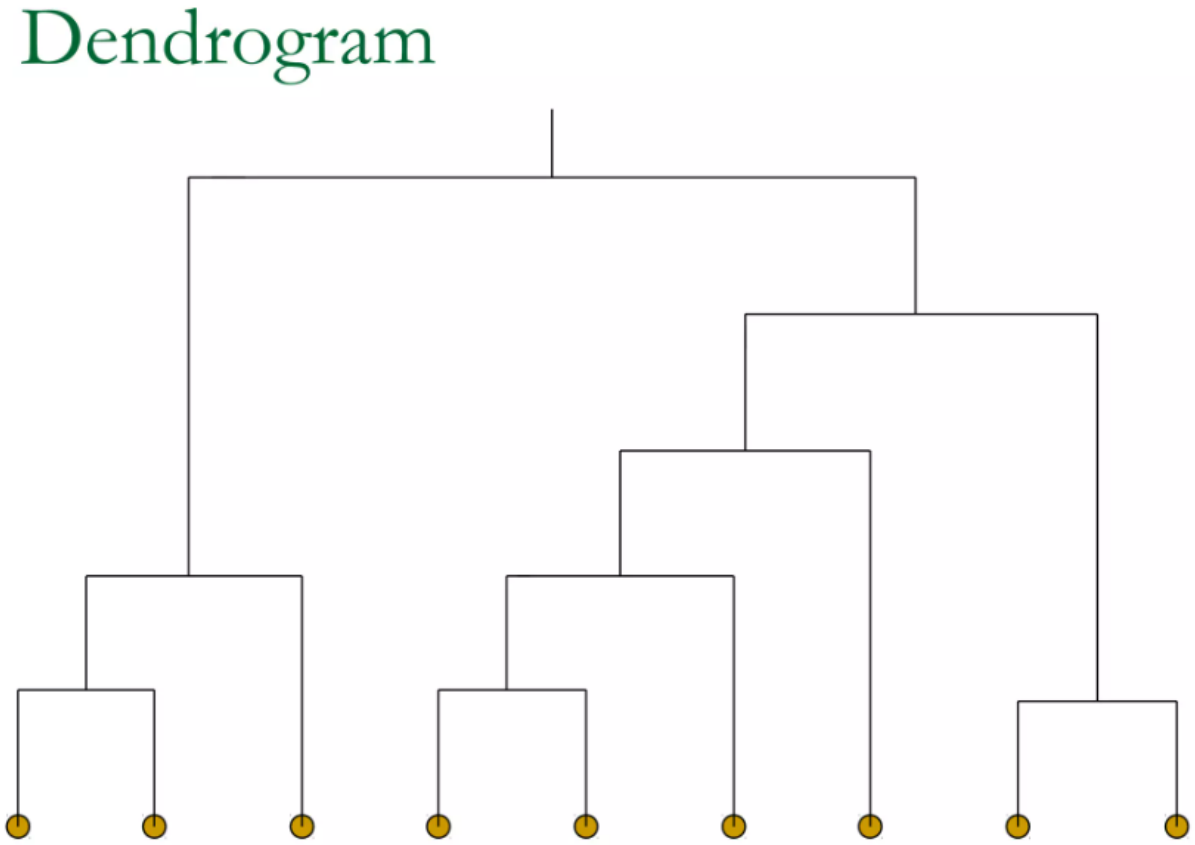
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



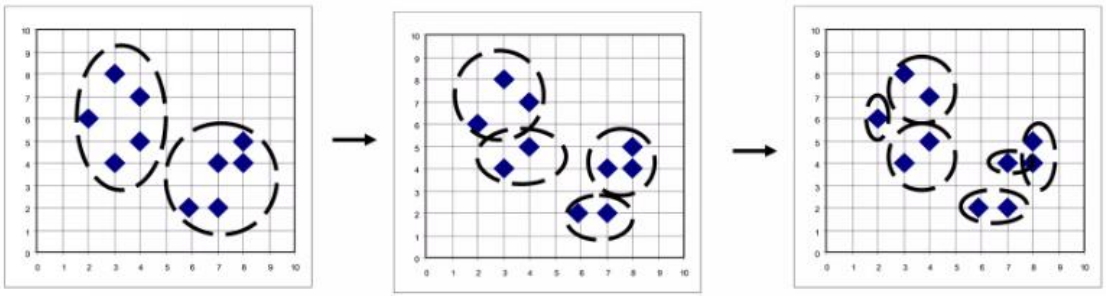
Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is this:

- Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Steps:-

Step-1: The DIANA clustering is followed by Agglomerative Hierarchical Clustering up to the cluster contains all the objects. Then the Divisive Analysis Clustering (DIANA) follows the top-down approach assuming it single cluster having level L (0) = n and sequence number m = 0.

- Step-2: The most dissimilar pair of clusters in the current cluster is found out; that is (r), (s) in which d [(r), (s)] = min d [(i), (j)], where min is the complete pairs of cluster in the current cluster.

- Step-3: The sequence number is incremented in the manner $m = m + 1$. The cluster is broken into clusters (r) and (s) to form next cluster to make the level of clustering: L (m1) = d [(r)] and L (m2) = d [(s)].

- Step-4: The distance matrix (D) is updated by adding the rows and columns corresponding to clusters (r) and (s). The similarity between the new cluster, denoted by (r, s) and old cluster (k) is defined in this way:
D[(k),(r,s)]=mind[(k),(r)],d[(k),(s)]

If all objects are distinct clusters, then stop; otherwise proceed to step-2.

K Means

Back at K-Means Clustering

length	width	height
169	65.7	49.6
166.3	64.4	53
173.2	66.3	50.2
183.5	67.7	52
159.3	64.2	55.6
178.2	67.9	52
170.7	67.9	49.7
175.6	66.5	54.9
186.7	68.4	56.7
165.3	63.8	54.5

- ✓ Pick cluster centroids from the available data
- ✓ calculate distance of centroids with each row

Euclidean distance between the first and second row

$$= \sqrt{(\text{Length}_1 - \text{Length}_2)^2 + (\text{Width}_1 - \text{width}_2)^2 + (\text{Height}_1 - \text{Height}_2)^2}$$

- ✓ update centroids every time a row gets added to the cluster

K-Means Clustering – Solved Example

- Suppose that the data mining task is to cluster points into three clusters,
- where the points are
- *A*1(2, 10), *A*2(2, 5), *A*3(8, 4), *B*1(5, 8), *B*2(7, 5), *B*3(6, 4), *C*1(1, 2), *C*2(4, 9).
- The distance function is Euclidean distance.
- Suppose initially we assign *A*1, *B*1, and *C*1 as the center of each cluster, respectively.

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

Data Points			Distance to						Cluster	New Cluster
			2	10	5	8	1	2		
A1	2	10								
A2	2	5								
A3	8	4								
B1	5	8								
B2	7	5								
B3	6	4								
C1	1	2								
C2	4	9								

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

New Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			2	10	5	8	1	2		
A1	2	10	0.00		3.61		8.06		1	
A2	2	5	5.00		4.24		3.16		3	
A3	8	4	8.49		5.00		7.28		2	
B1	5	8	3.61		0.00		7.21		2	
B2	7	5	7.07		3.61		6.71		2	
B3	6	4	7.21		4.12		5.39		2	
C1	1	2	8.06		7.21		0.00		3	
C2	4	9	2.24		1.41		7.62		2	

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Current Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

New Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			2	10	6	6	1.5	1.5		
A1	2	10	0.00		5.66		6.52		1	1
A2	2	5	5.00		4.12		1.58		3	3
A3	8	4	8.49		2.83		6.52		2	2
B1	5	8	3.61		2.24		5.70		2	2
B2	7	5	7.07		1.41		5.70		2	2
B3	6	4	7.21		2.00		4.53		2	2
C1	1	2	8.06		6.40		1.58		3	3
C2	4	9	2.24		3.61		6.04		2	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Current Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

New Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			3	9.5	6.5	5.25	1.5	3.5		
A1	2	10	1.12		6.54		6.52		1	1
A2	2	5	4.61		4.51		1.58		3	3
A3	8	4	7.43		1.95		6.52		2	2
B1	5	8	2.50		3.13		5.70		2	1
B2	7	5	6.02		0.56		5.70		2	2
B3	6	4	6.26		1.35		4.53		2	2
C1	1	2	7.76		6.39		1.58		3	3
C2	4	9	1.12		4.51		6.04		1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Current Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			3.67	9	7	4.33	1.5	3.5		
A1	2	10	1.94		7.56		6.52		1	1
A2	2	5	4.33		5.04		1.58		3	3
A3	8	4	6.62		1.05		6.52		2	2
B1	5	8	1.67		4.18		5.70		1	1
B2	7	5	5.21		0.67		5.70		2	2
B3	6	4	5.52		1.05		4.53		2	2
C1	1	2	7.49		6.44		1.58		3	3
C2	4	9	0.33		5.55		6.04		1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

As new cluster values are simillar to the old cluster values , wewill stop the process and finalize the clusters as

Cluster-1 A1,B1,C2

Cluster-2 A3,B2,B3

Cluster-3 A2, C1

K Modes Clustering

Step by step clustering a sample binary data

- ✓ K Modes - Preferred method for clustering binary data
- ✓ Calculate Distance Measure
- Simple Matching - Dissimilarity Measure $\frac{b + c}{a + b + c + d}$
- Jaccard
- Dice

K-Modes Clustering

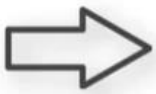
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	
Row 1	0	0	0	1	0	0	0	1	0	1	cluster centroid 1
Row 2	0	0	1	1	0	1	0	0	0	0	
Row 3	0	0	0	0	0	1	0	0	0	0	
Row 4	0	0	0	1	0	0	1	0	1	1	
Row 5	0	0	0	0	1	1	0	0	0	0	cluster centroid 2
Row 6	1	1	0	0	1	1	0	0	0	1	
Row 7	1	1	1	0	1	1	0	1	0	1	
Row 8	0	1	0	0	0	0	1	1	1	0	
Row 9	1	1	1	1	0	1	0	0	1	1	
Row 10	0	0	1	0	0	0	1	1	0	1	cluster centroid 3

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	1	0	0	0	1	0	1

cluster centroid 1

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Row 2	0	0	1	1	0	1	0	0	0	0

Distance Matrix



	1	0
1	**a = 1**	**b = 2**
0	**c = 2**	**d = 6**

Simple Matching - Dissimilarity Measure

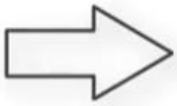
$$\frac{b + c}{a + b + c + d} = \frac{2 + 2}{1 + 1 + 2 + 6} = \frac{4}{10} = 0.4$$

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	0	1	1	0	0	0	0

cluster centroid 2

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Row 2	0	0	1	1	0	1	0	0	0	0

Distance Matrix



	1	0
1	**a = 1**	**b = 1**
0	**c = 2**	**d = 6**

Simple Matching - Dissimilarity Measure

$$\frac{b + c}{a + b + c + d} = \frac{1 + 2}{1 + 2 + 2 + 5} = \frac{3}{10} = 0.3$$

K-Modes Clustering

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	0	1	1	0	0	0	0

cluster centroid 2

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Row 2	0	0	1	1	0	1	0	0	0	0

Simple Matching - Dissimilarity Measure

$$\frac{b+c}{a+b+c+d} = \frac{1+2}{1+2+2+5} = \frac{3}{10} = 0.3$$

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Row 3	0	0	0	0	0	1	0	0	0	0

cluster centroid 1

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	1	0	0	0	1	0	1

Distance Matrix

	1	0
1	a = 0	b = 3
0	c = 1	d = 6

Simple Matching - Dissimilarity

$$\frac{b+c}{a+b+c+d} = \frac{3+1}{0+3+1+6} = \frac{4}{10} = 0.4$$

cluster centroid 2

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	0	1	1	0	0	0	0

	1	0
1	a = 1	b = 1
0	c = 0	d = 8

$$\frac{1+0}{1+1+0+8} = \frac{1}{10} = 0.1$$

cluster centroid 3

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	1	0	0	0	1	1	0	1

	1	0
1	a = 0	b = 4
0	c = 1	d = 5

$$\frac{4+1}{0+4+1+5} = \frac{5}{10} = 0.5$$

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Row 6	1	1	0	0	1	1	0	0	0	1

cluster centroid 1

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	1	0	0	0	1	0	1

Distance Matrix

	1	0
1	a = 1	b = 2
0	c = 4	d = 3

Simple Matching - Dissimilarity

$$\frac{b+c}{a+b+c+d} = \frac{2+4}{1+2+4+3} = \frac{6}{10} = 0.6$$

cluster centroid 2

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	0	1	1	0	0	0	0

	1	0
1	a = 2	b = 0
0	c = 3	d = 5

$$\frac{0+3}{2+0+3+5} = \frac{3}{10} = 0.3$$

cluster centroid 3

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	1	0	0	0	1	1	0	1

	1	0
1	a = 1	b = 3
0	c = 4	d = 2

$$\frac{3+4}{1+3+4+2} = \frac{7}{10} = 0.7$$

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Row 7	1	1	1	0	1	1	0	1	0	1

cluster centroid 1

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	1	0	0	0	1	0	1

Distance Matrix

	1	0
1	a = 2	b = 1
0	c = 5	d = 2

Simple Matching - Dissimilarity

$$\frac{b+c}{a+b+c+d} = \frac{1+5}{2+1+5+2} = \frac{6}{10} = 0.6$$

cluster centroid 2

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	0	1	1	0	0	0	0

	1	0
1	a = 2	b = 0
0	c = 5	d = 3

$$\frac{0+5}{2+0+5+3} = \frac{5}{10} = 0.5$$

cluster centroid 3

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	1	0	0	0	1	1	0	1

	1	0
1	a = 3	b = 1
0	c = 4	d = 2

$$\frac{1+4}{3+1+4+2} = \frac{5}{10} = 0.5$$

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Row 8	0	1	0	0	0	0	1	1	1	0

cluster centroid 1

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	1	0	0	0	1	0	1

Distance Matrix

	1	0
1	a = 1	b = 2
0	c = 3	d = 4

Simple Matching - Dissimilarity

$$\frac{b+c}{a+b+c+d} = \frac{2+3}{1+2+3+4} = \frac{5}{10} = 0.5$$

cluster centroid 2

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	0	1	1	0	0	0	0

	1	0
1	a = 0	b = 2
0	c = 4	d = 4

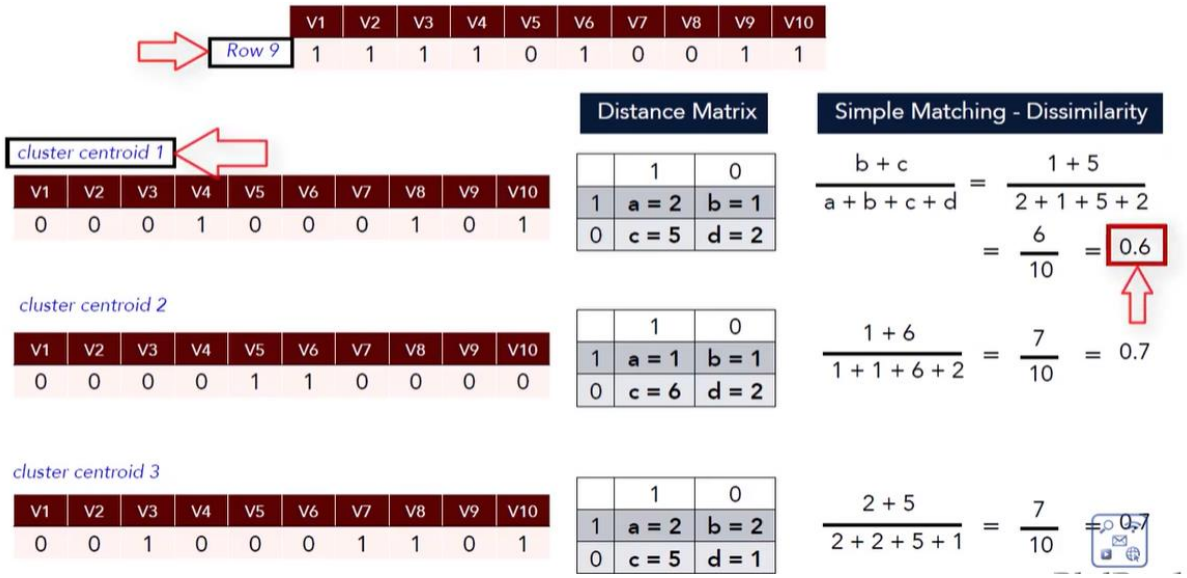
$$\frac{2+4}{0+2+4+4} = \frac{6}{10} = 0.6$$

cluster centroid 3

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	1	0	0	0	1	1	0	1

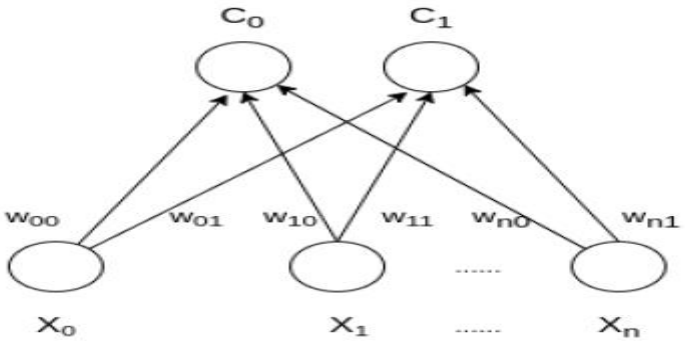
	1	0
1	a = 2	b = 2
0	c = 2	d = 4

$$\frac{2+2}{2+2+2+4} = \frac{4}{10} = 0.4$$



Self Organizing Maps – Kohonen Maps

Self Organizing Map (or Kohonen Map or SOM) is a type of Artificial Neural Network which is also inspired by biological models of neural systems from the 1970s. It follows an unsupervised learning approach and trained its network through a competitive learning algorithm. SOM is used for clustering and mapping (or dimensionality reduction) techniques to map multidimensional data onto lower-dimensional which allows people to reduce complex problems for easy interpretation. SOM has two layers, one is the Input layer and the other one is the Output layer.



Algorithm

Training:

Step 1: Initialize the weights wij random value may be assumed. Initialize the learning rate α.

Step 2: Calculate squared Euclidean distance.

$$D(j) = \Sigma (wij - xi)^2 \quad \text{where } i=1 \text{ to } n \text{ and } j=1 \text{ to } m$$

Step 3: Find index J, when D(j) is minimum that will be considered as winning index.

Step 4: For each j within a specific neighborhood of j and for all i, calculate the new weight.

$$wij(new)=wij(old) + \alpha[xi - wij(old)]$$

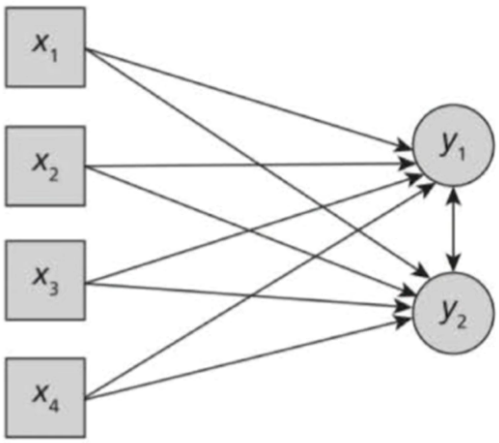
Step 5: Update the learning rule by using :

$$\alpha(t+1) = 0.5 * t$$

Step 6: Test the Stopping Condition.

Example:-

- Consider the network shown in Figure which considers four training samples each vector of length 4 and two output units.
- Train the SOFM network by determining the class memberships of the input data
- Training Samples:
- X1: (1, 0, 1, 0) X2: (1, 0, 0, 0)
- X3: (1, 1, 1, 1) X4: (0, 1, 1, 0)



- Output Units: Unit 1, Unit 2
- Learning rate η(t) = 0.6
- Initial Weight matrix
- $\begin{bmatrix} Unit\ 1 \\ Unit\ 2 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.5 & 0.7 & 0.2 \\ 0.6 & 0.5 & 0.4 & 0.2 \end{bmatrix}$

Iteration 1:

Training Sample x_1: (1, 0, 1, 0)

Weight matrix:

$$\begin{bmatrix} \text{Unit 1} \\ \text{Unit 2} \end{bmatrix} : \begin{bmatrix} 0.3 & 0.5 & 0.7 & 0.2 \\ 0.6 & 0.7 & 0.4 & 0.3 \end{bmatrix}$$

Compute Euclidean distance between x_1: (1, 0, 1, 0) and Unit 1 weights.

$$d^2 = (0.3 - 1)^2 + (0.5 - 0)^2 + (0.7 - 1)^2 + (0.2 - 0)^2 = 0.87$$

Compute Euclidean distance between x_1: (1, 0, 1, 0) and Unit 2 weights.

$$d^2 = (0.6 - 1)^2 + (0.7 - 0)^2 + (0.4 - 1)^2 + (0.3 - 0)^2 = 1.1$$

Unit 1 wins

New weight = old weight + learning rate*(Input - old weight)

$$w_j(t+1) = w_j(t) + \eta(t)(x_s - w_j(t))$$

Update the weights of the winning unit.

New Unit 1 weights = [0.3 0.5 0.7 0.2] + 0.6 ([1 0 1 0] – [0.3 0.5 0.7 0.2])

= [0.3 0.5 0.7 0.2] + 0.6 [0.7 –0.5 0.3 –0.2]

= [0.3 0.5 0.7 0.2] + [0.42 –0.30 0.18 –0.12]

= [0.72 0.2 0.88 0.08]

$$\begin{bmatrix}\text{Unit 1}\\ \text{Unit 2}\end{bmatrix} : \begin{bmatrix}0.72 & 0.2 & 0.88 & 0.08\\ 0.6 & 0.7 & 0.4 & 0.3\end{bmatrix}$$

Iteration 2:

Training Sample x_2: (1, 0, 0, 0)

Weight matrix:

$$\begin{bmatrix}\text{Unit 1}\\ \text{Unit 2}\end{bmatrix} : \begin{bmatrix}0.72 & 0.2 & 0.88 & 0.08\\ 0.6 & 0.7 & 0.4 & 0.3\end{bmatrix}$$

Compute Euclidean distance between x_2: (1, 0, 0, 0) and Unit 1 weights.

$$d^2 = (0.72 - 1)^2 + (0.2 - 0)^2 + (0.88 - 0)^2 + (0.08 - 0)^2 = 0.74$$

Compute Euclidean distance between x_2: (1, 0, 0, 0) and Unit 2 weights.

$$d^2 = (0.6 - 1)^2 + (0.7 - 0)^2 + (0.4 - 0)^2 + (0.3 - 0)^2 = 0.9$$

Unit 1 wins

$$w_j(t+1) = w_j(t) + \eta(t)(x_s - w_j(t))$$

Update the weights of the winning unit:

New Unit 1 weights = [0.72 0.2 0.88 0.08] + 0.6 ([1 0 0 0] – [0.72 0.2 0.88 0.08])

= [0.72 0.2 0.88 0.08] + 0.6 [0.28 –0.2 –0.88 –0.08]

= [0.72 0.2 0.88 0.08] + [0.17 –0.12 –0.53 –0.05]

= [0.89 0.08 0.35 0.03]

$$\begin{bmatrix}\text{Unit 1}\\ \text{Unit 2}\end{bmatrix} : \begin{bmatrix}0.89 & 0.08 & 0.35 & 0.03\\ 0.6 & 0.7 & 0.4 & 0.3\end{bmatrix}$$

Iteration 3:

Training Sample x_3: (1, 1, 1, 1)

Weight matrix:

$$\begin{bmatrix}\text{Unit 1}\\ \text{Unit 2}\end{bmatrix} : \begin{bmatrix}0.89 & 0.08 & 0.35 & 0.03\\ 0.6 & 0.7 & 0.4 & 0.3\end{bmatrix}$$

Compute Euclidean distance between x_3: (1, 1, 1, 1) and Unit 1 weights.

$d^2 = (0.89 - 1)^2 + (0.08 - 1)^2 + (0.35 - 1)^2 + (0.03 - 1)^2$

$= 2.2$

Compute Euclidean distance between x_3: (1, 1, 1, 1) and Unit 2 weights.

$d^2 = (0.6 - 1)^2 + (0.7 - 1)^2 + (0.4 - 1)^2 + (0.3 - 1)^2$

$= 1.1$

Unit 2 wins

$$w_j(t+1) = w_j(t) + \eta(t)(x_s - w_j(t))$$

Update the weights of the winning unit:

New Unit 2 weights = [0.6 0.7 0.4 0.3] + 0.6 ([1 1 1 1] – [0.6 0.7 0.4 0.3])

= [0.6 0.7 0.4 0.3] + 0.6 [0.4 0.3 0.6 0.7]

= [0.6 0.7 0.4 0.3] + [0.24 0.18 0.36 0.42] = [0.84 0.88 0.76 0.72]

$$\begin{bmatrix}\text{Unit 1}\\ \text{Unit 2}\end{bmatrix} : \begin{bmatrix}0.89 & 0.08 & 0.35 & 0.03\\ 0.84 & 0.88 & 0.76 & 0.72\end{bmatrix}$$

Iteration 4:

Training Sample x_4: (0, 1, 1, 0)

Weight matrix:

$$\begin{bmatrix}\text{Unit 1}\\ \text{Unit 2}\end{bmatrix}:\begin{bmatrix}0.89 & 0.08 & 0.35 & 0.03\\ 0.84 & 0.88 & 0.76 & 0.72\end{bmatrix}$$

Compute Euclidean distance between x_4: (0, 1, 1, 0) and Unit 1 weights.

$d^2 = (0.89 - 0)^2 + (0.08 - 1)^2 + (0.35 - 1)^2 + (0.03 - 0)^2$

$= 2.06$

Compute Euclidean distance between x_1: (0, 1, 1, 0) and Unit 2 weights.

$d^2 = (0.84 - 0)^2 + (0.88 - 1)^2 + (0.76 - 1)^2 + (0.72 - 0)^2$

$= 1.3$

$$w_j(t+1) = w_j(t) + \eta(t)(x_s - w_j(t))$$

Unit 2 wins

Update the weights of the winning unit:

New Unit 2 weights $= [0.84\ \ 0.88\ \ 0.76\ \ 0.72] + 0.6\,([0\ \ 1\ \ 1\ \ 0] - [0.84\ \ 0.88\ \ 0.76\ \ 0.72])$

$= [0.84\ \ 0.88\ \ 0.76\ \ 0.72] + 0.6\,[-0.84\ \ 0.12\ \ 0.24\ \ -0.72]$

$= [0.84\ \ 0.88\ \ 0.76\ \ 0.72] + [-0.5\ \ 0.07\ \ 0.14\ \ -0.43] = [0.34\ \ 0.95\ \ 0.9\ \ 0.29]$

$$\begin{bmatrix}\text{Unit 1}\\ \text{Unit 2}\end{bmatrix}:\begin{bmatrix}0.89 & 0.08 & 0.35 & 0.03\\ 0.34 & 0.95 & 0.9 & 0.29\end{bmatrix}$$

Best mapping units for each of the sample taken are:

x_1: (1, 0, 1, 0) → Unit 1

x_2: (1, 0, 0, 0) → Unit 1

x_3: (1, 1, 1, 1) → Unit 2

x_4: (0, 1, 1, 0) → Unit 2

This process is continued for many epochs until the feature map does not change.

EM ALGORITHM

- In the real-world applications of machine learning, it is very common that there are many relevant features available for learning but only a small subset of them are observable.
- The ***Expectation-Maximization algorithm*** can be used for the latent variables (variables that are not directly observable and are actually inferred from the values of the other observed variables).
- This algorithm is actually the base for many unsupervised clustering algorithms in the field of machine learning.

Algorithm:

1. Given a set of incomplete data, consider a set of starting parameters.
2. **Expectation step (E – step):** Using the observed available data of the dataset, estimate (guess) the values of the missing data.
3. **Maximization step (M – step):** Complete data generated after the expectation (E) step is used in order to update the parameters.
4. Repeat step 2 and step 3 until convergence.

Usage of EM algorithm –

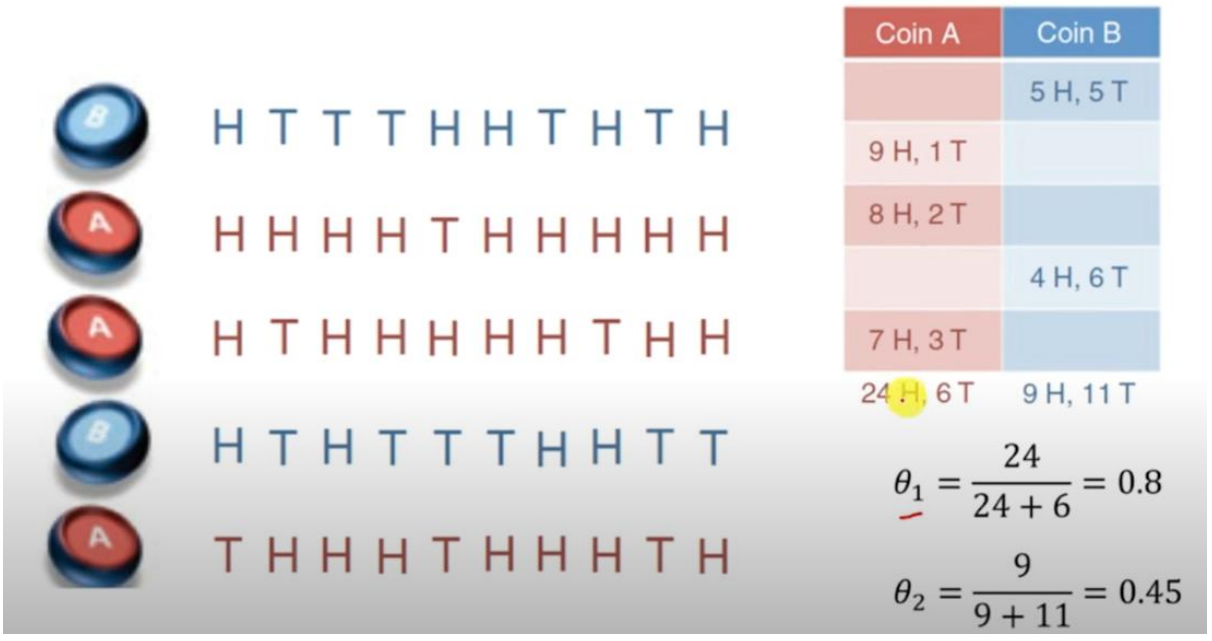
- It can be used to fill the missing data in a sample.
- It can be used as the basis of unsupervised learning of clusters.
- It can be used for the purpose of estimating the parameters of Hidden Markov Model (HMM).
- It can be used for discovering the values of latent variables.

Advantages of EM algorithm –

- It is always guaranteed that likelihood will increase with each iteration.
- The E-step and M-step are often pretty easy for many problems in terms of implementation.
- Solutions to the M-steps often exist in the closed form.

Disadvantages of EM algorithm –

- It has slow convergence.
- It makes convergence to the local optima only.
- It requires both the probabilities, forward and backward (numerical optimization requires only forward probability).



$$P(E \mid Z_A) = P(HHHHHHHHHT \mid \text{A chosen}) = \binom{n}{x} \theta_A^x (1-\theta_A)^n$$

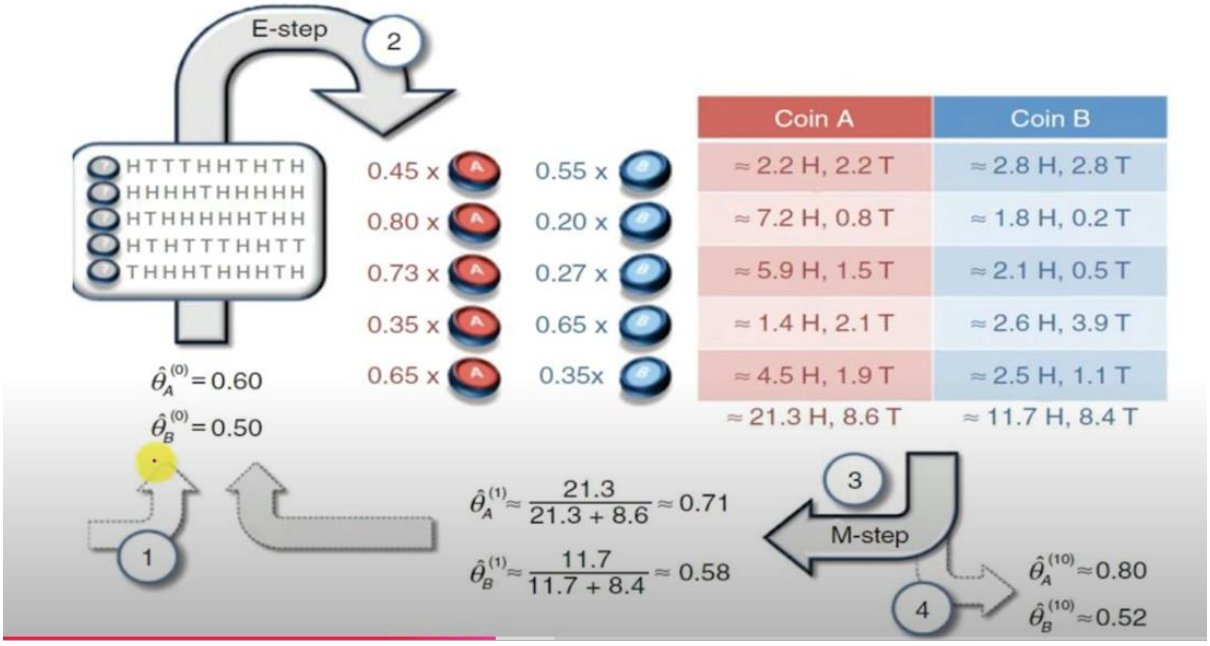
$$P(E \mid Z_B) = P(HHHHHHHHHT \mid \text{B chosen}) = \binom{n}{x} \theta_B^x (1-\theta_B)^n$$

$$P(E|Z_A) = \binom{9}{1} * (0.6)^9 * (0.4)^1 = 0.036$$

$$P(E|Z_B) = \binom{9}{1} * (0.5)^9 * (0.5)^1 = 0.009$$

$$P(Z_A|E) = \frac{0.036}{0.036 + 0.009} = 0.80$$

$$P(Z_B|E) = \frac{0.009}{0.036 + 0.009} = 0.20$$

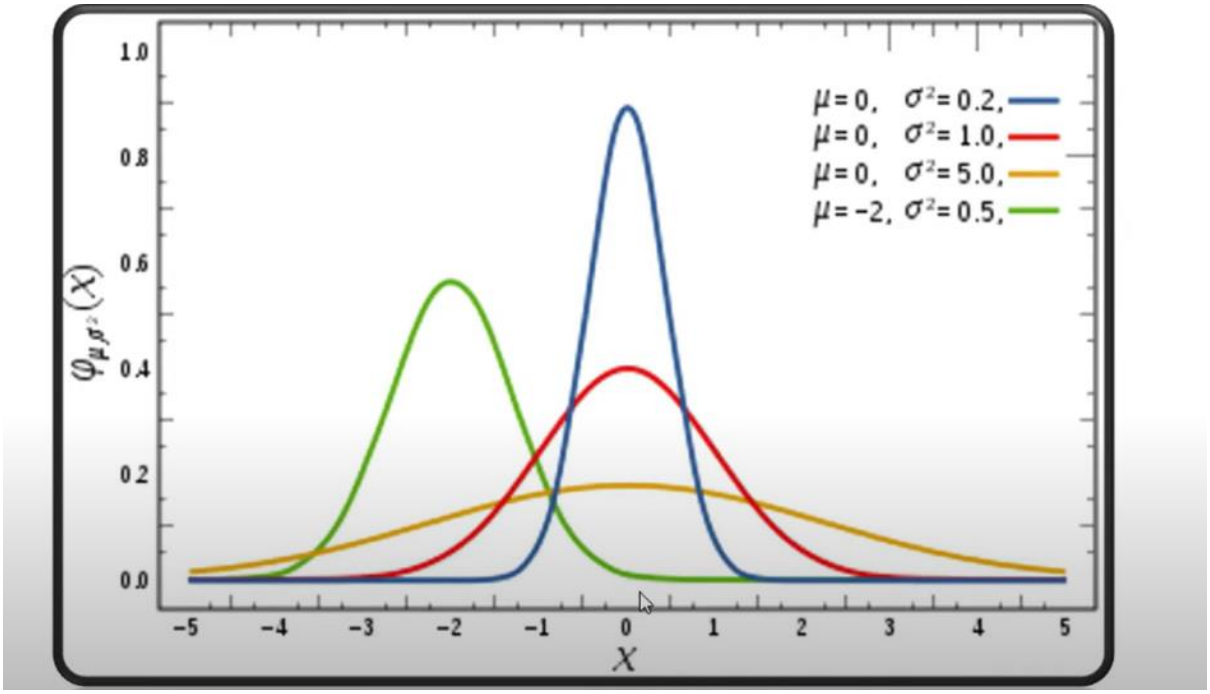


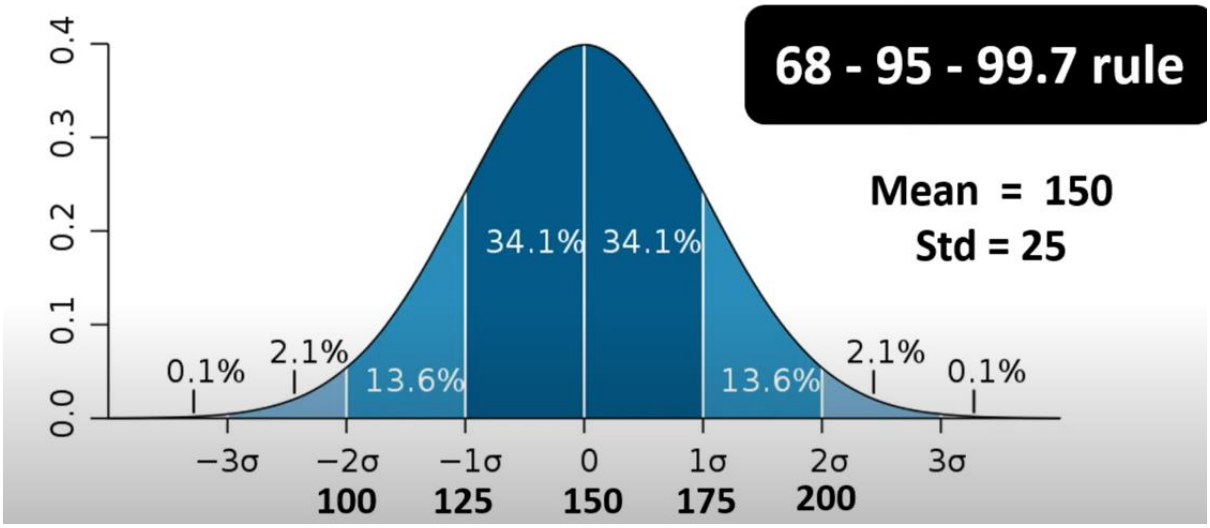
Gaussian mixture model (GMM)

A Gaussian mixture model (GMM) is a machine learning method used to determine the probability each data point belongs to a given cluster. The model is a soft clustering method used in unsupervised learning.

Gaussian Mixture Model Defined

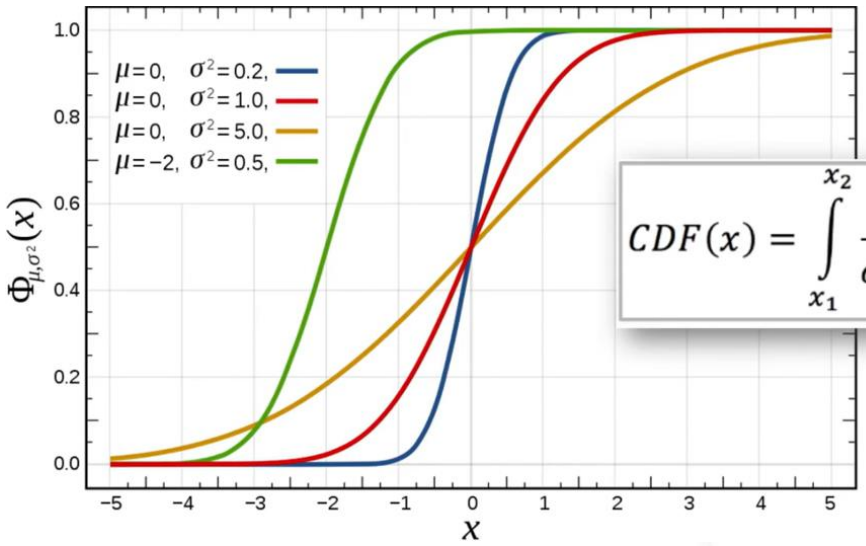
A Gaussian mixture model is a soft clustering technique used in unsupervised learning to determine the probability that a given data point belongs to a cluster. It's composed of several Gaussians, each identified by $k \in \{1,\ldots, K\}$, where K is the number of clusters in a data set.





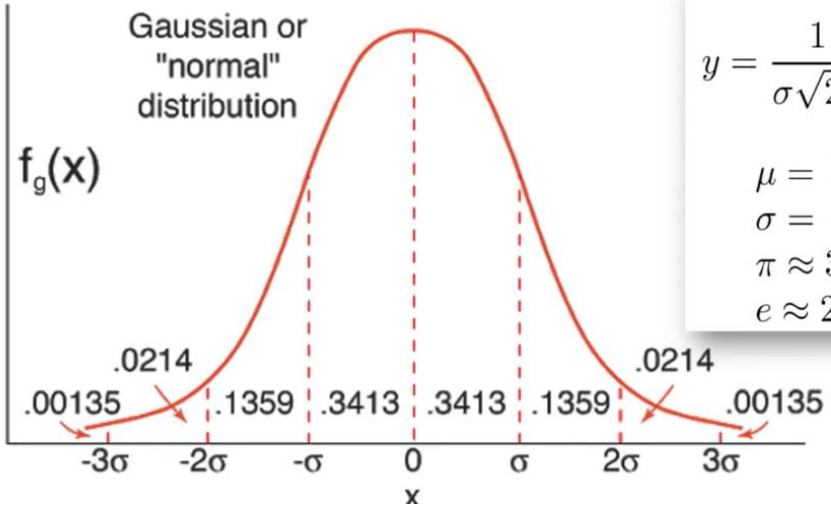
PDF, Probability Distribution Function is a function in mathematics that gives the probability of all the possible outcomes of any event.

CDF, cumulative distribution function, Cumulative distribution functions are also used to specify the distribution of multivariate random variables.



$$CDF(x) = \int_{x_1}^{x_2} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx$$

CDF



$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$\mu =$ Mean
$\sigma =$ Standard Deviation
$\pi \approx 3.14159\cdots$
$e \approx 2.71828\cdots$

PDF

Principal components analysis (PCA).

Step 1. Data

- We consider a dataset having n features or variables denoted by $X_1; X_2; \ldots ; X_n$.
- Let there be N examples.
- Let the values of the i^{th} feature X_i be $X_{i1}; X_{i2}; \ldots ; X_{iN}$

Features	Example 1	Example 2	$\cdots$	Example N
X_1	X_{11}	X_{12}	$\cdots$	X_{1N}
X_2	X_{21}	X_{22}	$\cdots$	X_{2N}
$\vdots$				
X_i	X_{i1}	X_{i2}	$\cdots$	X_{iN}
$\vdots$				
X_n	X_{n1}	X_{n2}	$\cdots$	X_{nN}

Step 2. Compute the means of the variables

Features	Example 1	Example 2	$\cdots$	Example N
X_1	X_{11}	X_{12}	$\cdots$	X_{1N}
X_2	X_{21}	X_{22}	$\cdots$	X_{2N}
$\vdots$				
X_i	X_{i1}	X_{i2}	$\cdots$	X_{iN}
$\vdots$				
X_n	X_{n1}	X_{n2}	$\cdots$	X_{nN}

$$\bar{X}_{i\cdot} = \frac{1}{N}(X_{i1} + X_{i2} + \cdots + X_{iN})$$

Step 3. Calculate the covariance matrix

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}

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

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

Step 4. Calculate the eigenvalues and eigenvectors of the covariance matrix

i. **Set up the equation:** This is a polynomial equation of degree **n** in. It has **n** real roots and these roots are the eigenvalues of **S**

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

ii. **If $\lambda = \lambda'$ is an eigenvalue**, then the corresponding eigenvector is a vector

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad \text{such that} \quad (S - \lambda' I)U = 0$$

Step 4. Calculate the eigenvalues and eigenvectors of the covariance matrix

iii. **We now normalize the eigenvectors**. Given any vector X we normalize 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

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

Step 5. Derive new data set

- Order the eigenvalues from highest to lowest.
- The unit eigenvector corresponding to the largest eigenvalue is the first principal component.

i) Let the eigenvalues in descending order be $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ and let the corresponding unit eigenvectors be $e_1, e_2, \ldots, e_n$.

ii) Choose a positive integer p such that $1 \leq p \leq n$.

iii) Choose the eigenvectors corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and form the following $p \times n$ matrix (we write the eigenvectors as row vectors):

Step 5. Derive new data set

iv) We form the following $n \times N$ matrix:

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

v) Next compute the matrix:

$$X_{\text{new}} = FX.$$

Note that this is a p × N matrix. This gives us a dataset of N samples having p features.

PCA

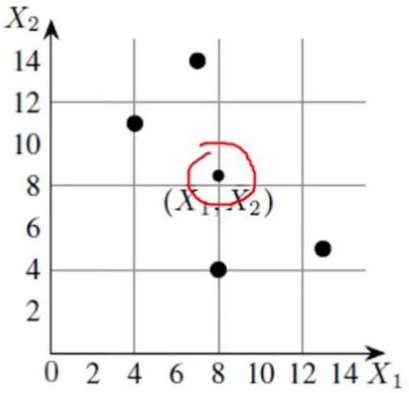
- Given the data in Table, reduce the dimension from 2 to 1 using the Principal Component Analysis (PCA) algorithm.

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

Step 1: Calculate Mean

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

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14



Step 2: Calculation of the covariance matrix.

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

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

$$\begin{aligned} \mathrm{Cov}(X_1, X_1) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{1k} - \bar{X}_1)(X_{1k} - \bar{X}_1) \\ &= \tfrac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right) \\ &= 14 \end{aligned}$$

$$\begin{aligned} \mathrm{Cov}(X_1, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{1k} - \bar{X}_1)(X_{2k} - \bar{X}_2) \\ &= \tfrac{1}{3}((4-8)(11-8.5) + (8-8)(4-8.5) \\ &\quad + (13-8)(5-8.5) + (7-8)(14-8.5) \\ &= -11 \end{aligned}$$

$$\begin{aligned} \mathrm{Cov}(X_2, X_1) &= \mathrm{Cov}(X_1, X_2) \\ &= -11 \end{aligned}$$

$$\begin{aligned} \mathrm{Cov}(X_2, X_2) &= \frac{1}{N-1}\sum_{k=1}^{N}(X_{2k} - \bar{X}_2)(X_{2k} - \bar{X}_2) \\ &= \tfrac{1}{3}\left((11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\right) \\ &= 23 \end{aligned}$$

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

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

F	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

Step 3: Eigenvalues of the covariance matrix

The characteristic equation of the covariance matrix is,

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

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

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

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

$$\overline{X_1} = 8$$

$$\overline{X_2} = 8.5$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$