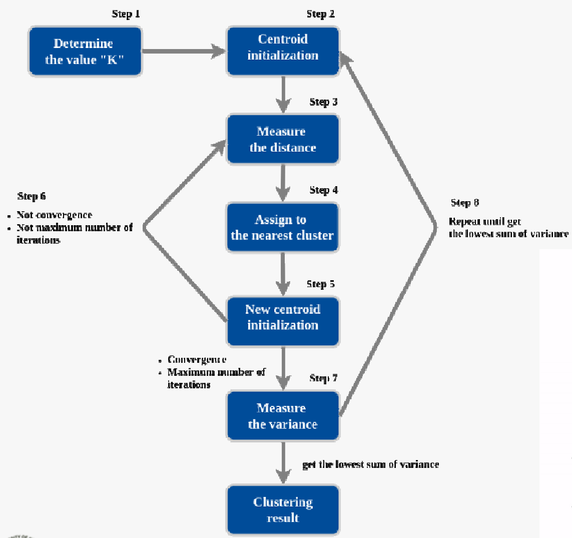
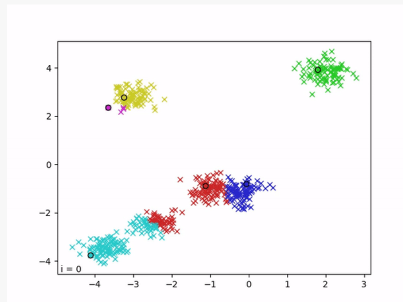
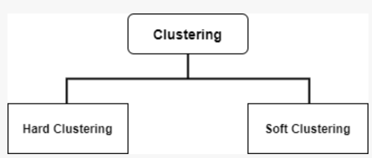
k-Means Clustering

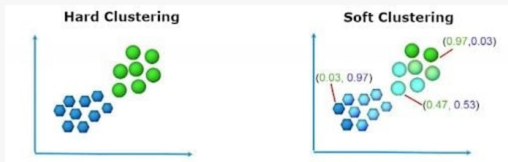




Fuzzy C-Means Clustering



Fuzzy C-Means Clustering



- **Step 1:** Given the data points based on the number of clusters required initialize the membership table with random values.
- Suppose the given data points are **{(1, 3), (2, 5), (6, 8), (7, 9)}**

Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.8	0.7	0.2	0.1
2	0.2	0.3	0.8	0.9

Fuzzy C-Means Clustering

- **Step 2: Find out the centroid.**
- The formula for finding out the centroid (V) is:
- $V_{ij} = \frac{\sum_{k=1}^{n} \gamma_{ik}^{m} * x_k}{\sum_{k=1}^{n} \gamma_{ik}^{m}}$
  - γ: Fuzzy membership value
  - m: Fuzziness parameter generally taken as 2 and
  - x_k is the data point
- $V_{11} = \frac{(0.8^2 * 1 + 0.7^2 * 2 + 0.2^2 * 4 + 0.1^2 * 7)}{(0.8^2 + 0.7^2 + 0.2^2 + 0.1^2)} = 1.568$
- $V_{12} = \frac{(0.8^2 * 3 + 0.7^2 * 5 + 0.2^2 * 8 + 0.1^2 * 9)}{(0.8^2 + 0.7^2 + 0.2^2 + 0.1^2)} = 4.051$

Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.8	0.7	0.2	0.1
2	0.2	0.3	0.8	0.9

- **Step 2: Find out the centroid.**
- The formula for finding out the centroid (V) is:
- $V_{ij} = \frac{\sum_{k=1}^{n} \gamma_{ik}^{m} * x_k}{\sum_{k=1}^{n} \gamma_{ik}^{m}}$
  - γ: Fuzzy membership value
  - m: Fuzziness parameter generally taken as 2 and
  - x_k is the data point
- $V_{21} = \frac{(0.2^2 * 1 + 0.3^2 * 2 + 0.8^2 * 4 + 0.9^2 * 7)}{(0.2^2 + 0.3^2 + 0.8^2 + 0.9^2)} = 5.35$
- $V_{22} = \frac{(0.2^2 * 3 + 0.3^2 * 5 + 0.8^2 * 8 + 0.9^2 * 9)}{(0.2^2 + 0.3^2 + 0.8^2 + 0.9^2)} = 8.215$

Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.8	0.7	0.2	0.1
2	0.2	0.3	0.8	0.9

Fuzzy C-Means Clustering

- **Step 3: Find out the distance of each point from the centroid.**
- $D_{11} = \sqrt{(1 - 1.568)^2 + (3 - 4.051)^2} = 1.2$
- $D_{12} = \sqrt{(1 - 5.35)^2 + (3 - 8.215)^2} = 6.79$
- $D_{21} = \sqrt{(2 - 1.568)^2 + (5 - 4.051)^2} = 1.04$
- $D_{22} = \sqrt{(2 - 5.35)^2 + (5 - 8.215)^2} = 4.64$
- $D_{31} = \sqrt{(4 - 1.568)^2 + (8 - 4.051)^2} = 4.63$
- $D_{32} = \sqrt{(4 - 5.35)^2 + (8 - 8.215)^2} = 1.36$
- $D_{31} = \sqrt{(7 - 1.568)^2 + (9 - 4.051)^2} = 7.34$
- $D_{32} = \sqrt{(7 - 5.35)^2 + (9 - 8.215)^2} = 1.82$

Centroids are:
(1.568, 4.051) and
(5.35, 8.215)

Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.8	0.7	0.2	0.1
2	0.2	0.3	0.8	0.9
	1	1	2	2

Fuzzy C-Means Clustering

- **Step 4: Updating membership values.**
- $\gamma_{ki} = \left(\sum_{j=1}^{n} \left\{ \frac{d_{ki}^2}{d_{kj}^2} \right\}^{\left(\frac{1}{(m-1)}\right)} \right)^{-1}$
- For point 1 new membership values are:
- $\gamma_{11} = \left(\left\{ \frac{(1.2)^2}{(1.2)^2} + \frac{(1.2)^2}{(6.79)^2} \right\}^{\left(\frac{1}{(2-1)}\right)} \right)^{-1} = 0.97$
- $\gamma_{12} = \left(\left\{ \frac{(6.79)^2}{(1.2)^2} + \frac{(6.79)^2}{(6.79)^2} \right\}^{\left(\frac{1}{(2-1)}\right)} \right)^{-1} = 0.03$

$D_{11} = 1.2, \quad D_{12} = 6.79$
$D_{21} = 1.04, \quad D_{22} = 4.64$
$D_{31} = 4.63, \quad D_{32} = 1.36$
$D_{31} = 7.34, \quad D_{32} = 1.82$

Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.97	0.7	0.2	0.1
2	0.03	0.3	0.8	0.9

- **Step 4: Updating membership values.**
- $\gamma_{ki} = \left(\sum_{j=1}^{n} \left\{ \frac{d_{ki}^2}{d_{kj}^2} \right\}^{\left(\frac{1}{(m-1)}\right)} \right)^{-1}$
- For point 4 new membership values are:
- $\gamma_{41} = \left(\left\{ \frac{(7.34)^2}{(7.34)^2} + \frac{(7.34)^2}{(1.82)^2} \right\}^{\left(\frac{1}{(2-1)}\right)} \right)^{-1} = 0.06$
- $\gamma_{42} = \left(\left\{ \frac{(1.82)^2}{(7.34)^2} + \frac{(1.82)^2}{(1.82)^2} \right\}^{\left(\frac{1}{(2-1)}\right)} \right)^{-1} = 0.94$

$D_{11} = 1.2, \quad D_{12} = 6.79$
$D_{21} = 1.04, \quad D_{22} = 4.64$
$D_{31} = 4.63, \quad D_{32} = 1.36$
$D_{31} = 7.34, \quad D_{32} = 1.82$

Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.97	0.95	0.08	0.06
2	0.03	0.05	0.92	0.94

Fuzzy C-Means Clustering

- **Step 5:** Repeat the steps (2-4) until the constant values are obtained for the membership values or the difference is less than the tolerance value

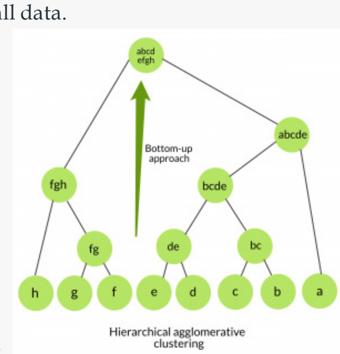
Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.8	0.7	0.2	0.1
2	0.2	0.3	0.8	0.9

Cluster	(1, 3)	(2, 5)	(4, 8)	(7, 9)
1	0.97	0.95	0.08	0.06
2	0.03	0.05	0.92	0.94

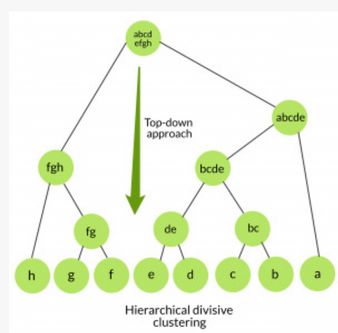
Hierarchical Clustering

Agglomerative Clustering

Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.



Hierarchical agglomerative clustering



Hierarchical divisive clustering

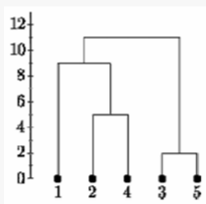
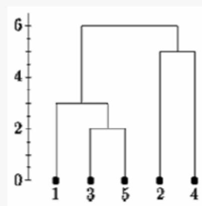
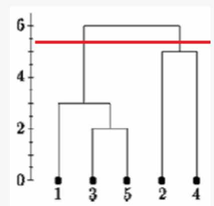
Divisive clustering

Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been split into singleton clusters.

Agglomerative Clustering

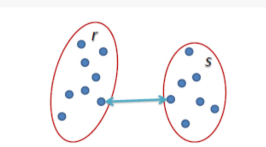
	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	$\boxed{2}$	8	0

	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0

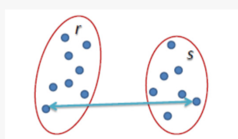




Hierarchical Clustering-Linkage Criteria

Single Linkage: The distance between two clusters is the shortest distance between two points in each cluster



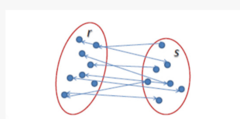
$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$



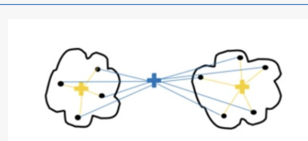
$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Complete Linkage: The distance between two clusters is the longest distance between two points in each cluster

Average Linkage: The distance between clusters is the average distance between each point in one cluster to every point in other cluster



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$



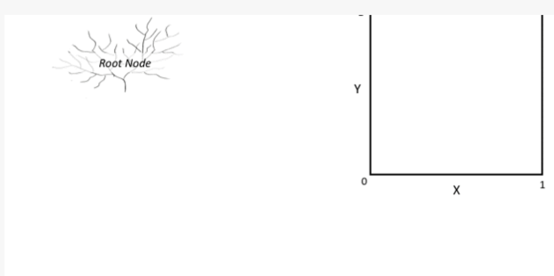
Ward Linkage: The distance between clusters is the sum of squared differences within all clusters

Hierarchical Clustering-Distance Metric

Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan (or city block) distance	$\|a-b\|_1 = \sum_i \lvert a_i - b_i \rvert$
Maximum distance (or Chebyshev distance)	$\|a-b\|_\infty = \max_i \lvert a_i - b_i \rvert$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the Covariance matrix

Decision Trees

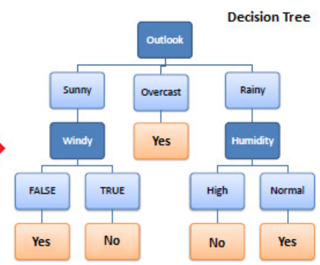
Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. In a decision tree, node represents a feature (attribute), link (branch) represents a decision (rule) and each leaf represents an outcome(categorical or continues value).



Decision Trees - ID3 Algorithm

- The core algorithm, called **ID3** by J. R. Quinlan
- A top-down, greedy search through the space of possible branches with no backtracking.
- ID3 uses *Entropy* and *Information Gain* to construct a decision tree.

Outlook	Temp.	Humidity	Windy	Play Golf
Predictors				Target
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



ID3 Algorithm - Entropy

Entropy

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

Total Entropy

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Entropy of an attribute

$$E(T, X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

ID3 Algorithm

Step 1: Calculate entropy of the target (total entropy)

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

ID3 Algorithm...

Step 2: The entropy for each branch is calculated. The Information Gain (decrease in entropy) is calculated for each attribute.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

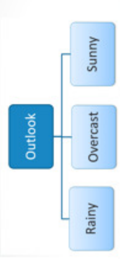
$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)
= 0.940 – 0.693 = 0.247

ID3 Algorithm...

Step 3: Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

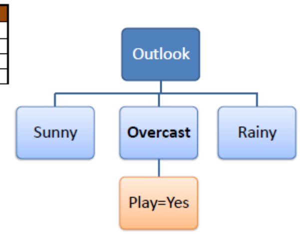
★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

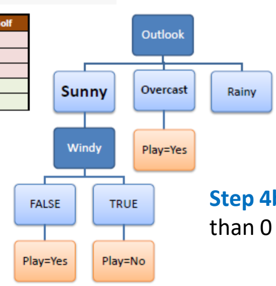
ID3 Algorithm...

Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Step 4a: A branch with entropy of 0 is a leaf node.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Step 4b: A branch with entropy more than 0 needs further splitting.

Step 5: The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Decision Trees

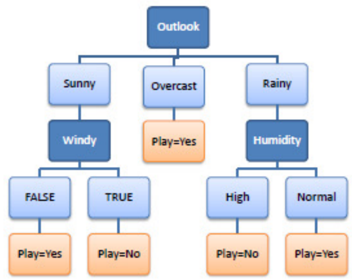
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes



Decision Trees - Issues

- Working with continuous attributes (binning)
- Avoiding overfitting
- Super Attributes (attributes with many unique values)
- Working with missing values