Clustering concepts

Clustering of Data

Welcome to week 3

- By the end of this week you will know:
  - Distance metrics and their usage in machine learning.
  - Clustering concept and it’s use for revealing patterns from unlabelled data

Distance Metrics

- Measuring similarity or distances between different data points is fundamental to many machine learning algorithms
  - unsupervised learning problems (i.e. K-means method in clustering)
  - supervised learning methods (i.e. K-Nearest-Neighbor)
- Distance measures are functions that define a distance $d(x_i, x_j)$ between any two data instances x_i and x_j for measuring how similar the instances are.

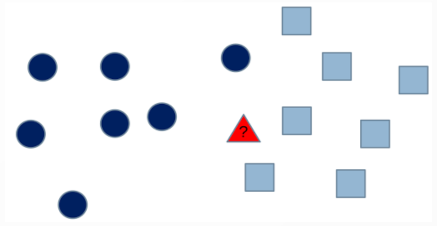
Distance Metrics...

- Distance measures satisfy the following three properties:
  - For any instance x_i ,distance with itself is zero, $d(x_i, x_i) = 0$
  - For an instance pairs x_i and x_j ,the distance is non-negative and symmetric, $d(x_i, x_j) \geq 0$ and $d(x_i, x_j) = d(x_j, x_i)$
  - Distance measure follows triangular inequality $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k)$

- Distance measures satisfying above properties are also known as Distance Metrics

Distance Metrics...

- Example 1: Nearest Neighbour Classification
  - Using distance to find the label of the new data point (the red triangle - Square or circle?)



- Example 2: Image retrieval
  - Animal types in NUS Wide Animal dataset

  - given a new image like the image of a cat, can we fetch all cat images from the dataset? (yes! with the help of distance measurements)

Distance Measurement Types

- Euclidean distance
  - Straight-line distance between two points in Euclidean space
  - For any two data instances, represented by d-dimensional feature vectors x_i ,x_j their Euclidean distance is

$$d_{Euclidean}(x_i, x_j) = \left(\left(x_{i,1} - x_{j,1}\right)^2 + \dots + \left(x_{i,D} - x_{j,D}\right)^2\right)^{1/2}$$

  - For example, consider these two vectors:

$$x_1 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix} \text{ and } x_1 = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 0 \\ 2 \end{bmatrix}$$

$$d(x_1, x_2) = ((1-0)^2 + (1-2)^2 + (2-2)^2 + (1-0)^2 + (0-2)^2)^{\frac{1}{2}}$$

$$= \sqrt{(1+1+0+1+4)} = \sqrt{7} = 2.65 \text{ (approx)}$$

Distance Measurement Types

- Cosine distance
  - For any two data instances, represented by d−dimensional feature vectors x_i ,x_j

$$d_{Cosine}(x_i, x_j) = 1 - \frac{x_i^T x_j}{\|X_i\|_2 . \|X_j\|_2}$$

  - Lets see an example

$$d(x_1, x_2) = 1 - \frac{(1*0+1*2+2*2+1*0+0*2)}{\sqrt{1^2+1^2+2^2+1^2+0^2} * \sqrt{0^2+2^2+2^2+0^2+2^2}}$$

$=0.3453$

$x_1 = \begin{bmatrix}1\\1\\2\\1\\0\end{bmatrix}$ and $x_2 = \begin{bmatrix}0\\2\\2\\0\\2\end{bmatrix}$

Distance Measurement Types

- Cityblock/Manhattan distance
  - For any two data instances, represented by d−dimensional feature vectors x_i ,x_j ,their Cityblock distance is computed as:

$$d_{Cityblock}(x_i, x_j) = |x_{i,1} - x_{j,1}| + \dots + |x_{i,D} - x_{j,D}|$$

  - In most cases, this results similar to the Euclidean distance
  - However, the effect of a large difference in a single dimension is dampened (since the distances are not squared)

$$d(x_1, x_2)_{CB} = |1-0| + |1-2| + |2-2| + |1-0| + |0-2| = 5$$

$x_1 = \begin{bmatrix}1\\1\\2\\1\\0\end{bmatrix}^T$ and $x_2 = \begin{bmatrix}0\\2\\2\\0\\2\end{bmatrix}^T$

Distance Measurement Types

- Minkowski distance
  - Defines a distance between two points in a normed vector space.
  - Euclidean and Cityblock distances are 2nd and 1st normed distance between x_i and x_j.
  - Minkowski distance is a generalization of these distances defined for any p-norm

$$d(x,y) = \left(\sum_{i=0}^{n-1} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

    - when p=1, d is Manhattan distance
    - when p=2, d is Euclidean distance

Distance Measurement Types

- Mahalanobis distance
  - MD is the distance between two points in multivariate space, represented by d−dimensional feature vectors x_i ,x_j

    $$d_{Mahalanobis}(x_i, x_j) = (x_i - x_j)M^{-1}(x_i - x_j)^T$$

    where, M is the covariance matrix of the data.
  - Intuitively, the covariance matrix generalizes the notion of variance to multiple dimensions
  - MD can be thought of scaling each data dimension by its variance and adjusting for their relationships
  - When data are independent, i.e. M=I (identity matrix), Mahalanobis distance becomes same as Euclidean distance.

Distance Measurement Types

- Jaccard distance
  - Is a distance used to measure diversity of any two sets
  - Consider any two instances x_i and x_j as binary vectors indicating presence or absence of features
  - Jaccard distance between x_i and x_j is defined as $d_{Jaccard}(x_i, x_j) = 1 - \frac{|x_i \cap x_j|_1}{|x_i \cup x_j|_1}$
  - Where ∩ denotes logical 'AND' and ∪ denotes logical 'OR' operators. The $|x|_1$ is 1-norm
  - Jaccard distance for $x_i = [1,0,1]$, and $x_j = [1,1,0]$

$$d_{Jaccard}(x_i, x_j) = 1 - \frac{1}{3} = \frac{2}{3}$$

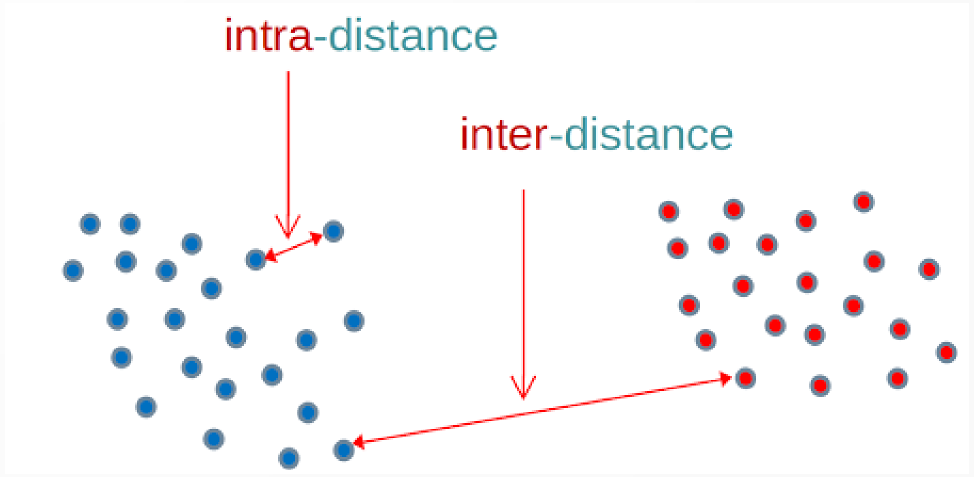
Clustering and It's Applications

- Humans are encoded to see patterns in everything. (related to ML ?!)
- Did I just saw a huge puffy white duck in the sky?

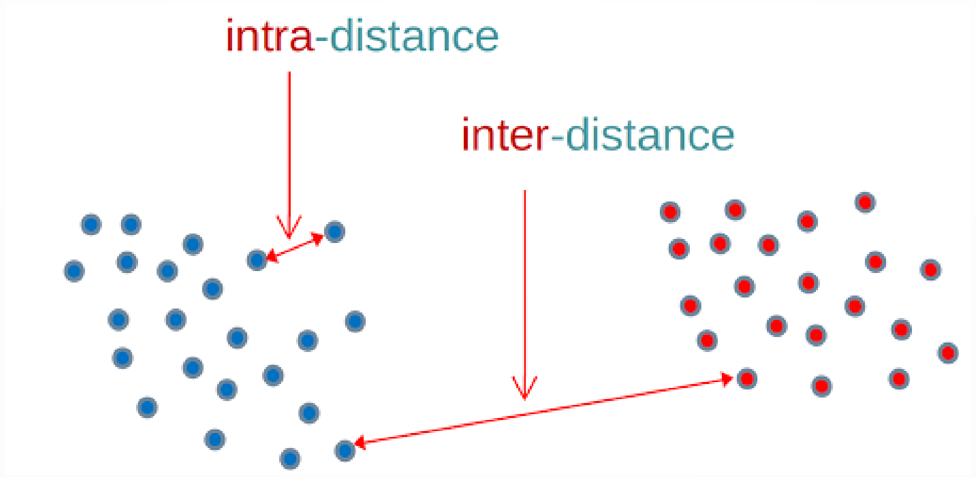
- You are probably right! Our brain prefers patterns and we always look for patterns!
- It looks that our brains do clustering unconsciously

Clustering Algorithms

- How do we teach a computer to do this?
- Goal of clustering algorithms are:
  - Group objects of similar properties together
  - Discover interesting clusters and groups in the data
  - Find valid organisation of the data
- In other words, we can define two algorithmic goals:
  - Minimise intra-distance (distance between points in the same cluster)
  - Maximise inter-distance (distance between points from different clusters)



Clustering Algorithms...



- Now we can define a generic set-up based on our current understanding from clustering methods:
  - Step 1: define a distance metric between objects
  - Step 2: define an objective function that gets us to our clustering goal
  - Step 3: devise an algorithm to optimise the objective function

How Kmeans Works

- The most popular clustering algorithm; simple and fast
- was independently discovered in 60s and 70s by Steinhaus (1955), Lloyd (1957), Ball and Hall (1965) and McQueen (1967)
- Kmeans
  - stores k centroids
  - A point is considered to be in a particular cluster if it is closer to that cluster’s centroid than any other centroid.
  - KMeans searches for the best centroids by alternating between two methods:
    - Assigning data points to clusters based on the current defined centroids (points which are the centre of a cluster).
    - Choosing centroids based on the current assignment of data points to clusters

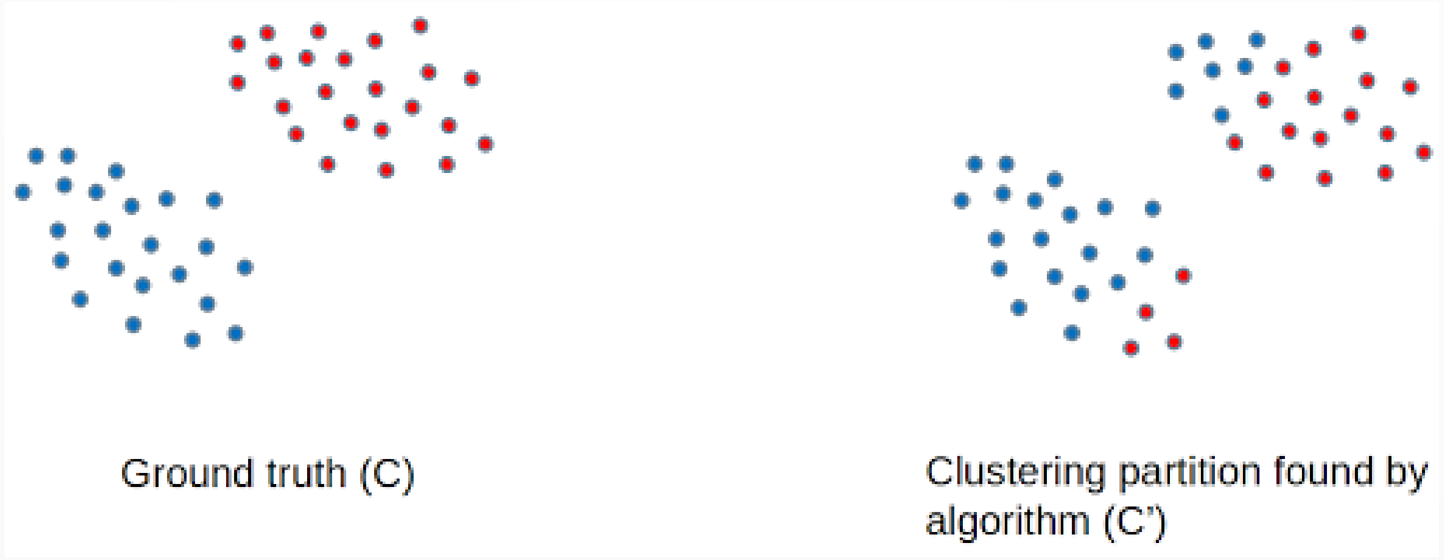
N.B. Please see the video presented in “3.5 How Kmeans works” for better understanding

Evaluation of Clustering

- Generally there are two main categories of evaluation methods for clustering
- External assessment:
  - compare clustering performance against a known clustering (often called Ground truth or Gold standard)
- Internal assessment:
  - determine if clustering follows certain intrinsic assumptions (e.g. cluster-to-cluster distance or cluster size etc.),
  - Examples: Silhouette coefficient, Dunn index etc.

Evaluation of Clustering...

- The following figure illustrates a sample of ground truth (C) and the clustering partition found by a clustering algorithm (C′)



Evaluation of Clustering...

- Rand Index
  - Is a measure of the similarity between two data clusters
  - Rand index is a function that measures the similarity of the two assignments C and C′, ignoring their permutations.
  - Rand index is computed as $R = \frac{a + \mathrm{b}}{\binom{n}{2}} = \frac{a + \mathrm{b}}{a + b + c + d}$
    - a = the number of pairs of data instances that are in the same cluster in both C,C′
    - b = the number of pairs of data instances that are in the different clusters in C and in different clusters in C′
    - c = the number of pairs of data instances that are in the same cluster in C but in different clusters in C′.
    - d = the number of pairs of data instances that are in the different clusters in C but in the same clusters in C′

Evaluation of Clustering...

- Purity
  - Purity is a way of quality measurement in clustering methods
  - As the name suggests, we would like to measure the purity for all clusters in terms of class labels of the data in each cluster i.e.,

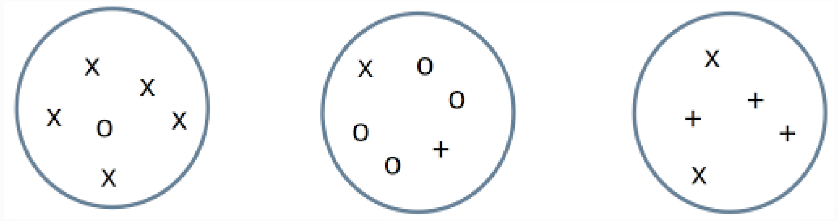
$$Purity = \frac{\sum_{i=1..q} A_i}{\sum_{i-1..q} n_i}$$

where q is the number of clusters, A_i is the number of correctly assigned elements in i^{th} cluster and n_i is the number of elements in each cluster.

  - This means the purity is measured by counting the number of correctly assigned instances and dividing by the number of total instances

Evaluation of Clustering...

- Purity Example
  - Let’s we have three clusters as shown in the figure.



  - Each cluster is assigned to the class label which has the majority in the cluster i.e., first cluster labelled as Cross, second as Circle and last one as Plus.
  - Based on the figure, 5 crosses, 4 circles, 3 pluses were correctly assigned. So, the purity of this clustering is:

$$Purity = \frac{1}{17} * (5 + 4 + 3) \approx 0.71$$

Evaluation of Clustering...

- Mutual Information
  - Mutual information is a function that measures the agreement of the two clustering assignments C and C′ in terms of how informative one is about the other, ignoring permutations.
  - To put it simple, how informative is C about C′
  - Let us assume that clustering partition C has K clusters and the partition C′ has K′ clusters

$$MI(C,C') = \sum_{i=1}^{K}\sum_{j=1}^{K\prime} P\,(i,j) log\frac{P(i,j)}{P(i)P'(j)}$$

where $P(i)$ denotes the probability of randomly selected instance to belong to i^{th} cluster of the partition C, similarly, $P(i,j)$ denotes the probability of a randomly selected instance to belong in i^{th} cluster of the partition C and j^{th} cluster of the particition C′. $P'(j)$ is defined similarly as $P(i)$.

  - So if our C′ clustering is highly informative based on C, we can conclude that the C clustering assignment is doing good

Evaluation of Clustering...

- Silhouette Coefficient
  - It is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation)
  - This method does not require the ground truth cluster assignments
  - The silhouette coefficient contrasts the average distance between the instances of the same cluster with the average distance between the instances of different clusters

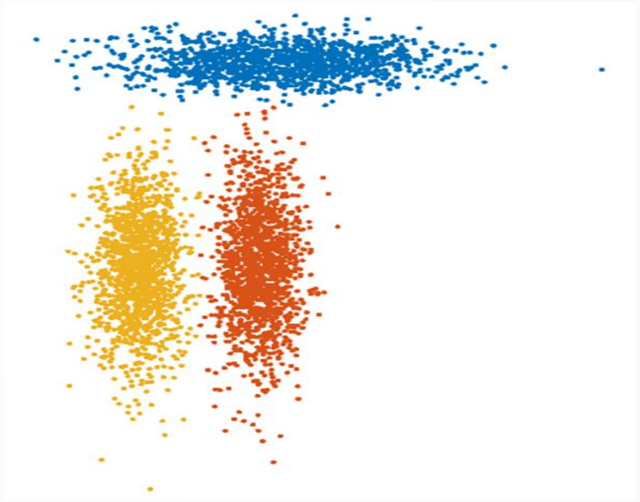
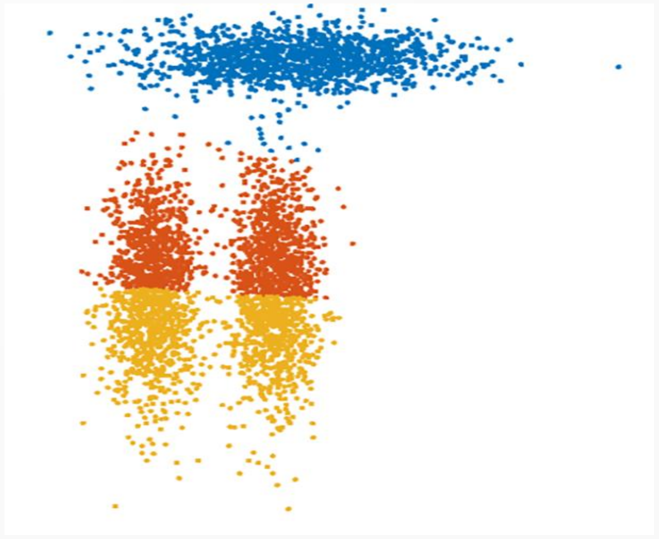
$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Where, $a(i)$ is the average distance of i^{th} instance with all other instances of the same cluster, and $b(i)$ is the lowest average dissimilarity of i^{th} instance with all other clusters.

  - The final value of Silhouette ranges from −1 to +1. High value of $s(i)$ indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

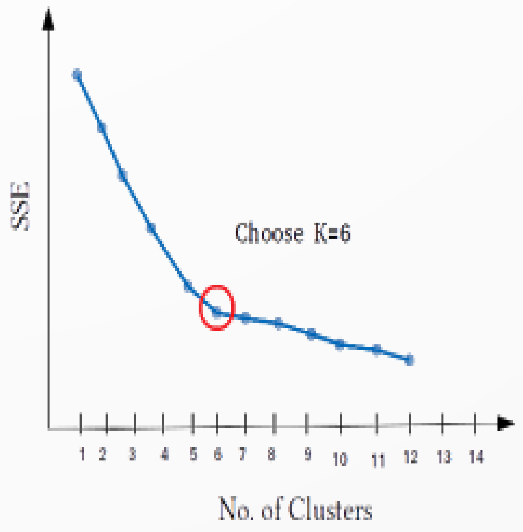
Limitations of Kmeans

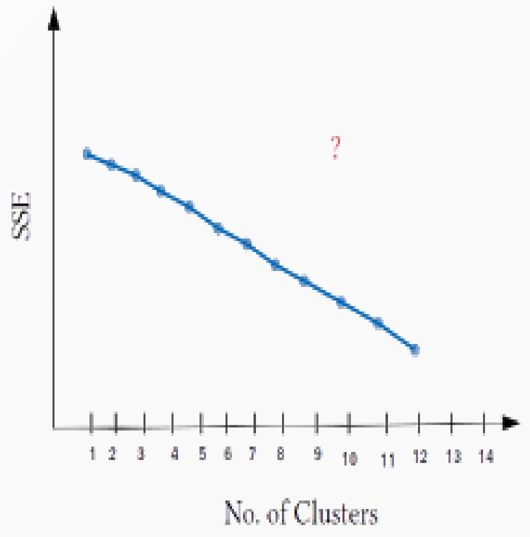
- Most important limitations of Simple Kmeans are:
  - Random initialisation means that you may get different clusters each time. As a solution, we can use Kmeans++ initialisation algorithm to initialise it better.
  - We have to supply the number of clusters beforehand. We can use Elbow method to choose K, but it may not be straightforward.
  - It cannot find clusters of arbitrary shapes.
  - It cannot detect noisy data points, i.e. they should not be taken into account for cluster analysis. K-median is less affected but cannot identify them.

Finding the Number of Clusters

- Elbow Method
  - The idea of elbow method is to run Kmeans clustering algorithm for a range of values of K, and for each value of K, compute the sum of squared error (SSE) as: $SSE_k = \sum \sum z_{ik} \|x_i - \mu_k\|^2$
  - So, $\|x_i - \mu_k\|^2$ finds the distance of each point (x_i) to its corresponding centroid (μ_k) in the cluster
  - z_{ik} is a binary variable
    - 1 when x_i is assigned to the cluster number k and
    - 0 when x_i is not related to cluster number k
  - it looks 6 is the best cluster number





Kmeans with Kmeans++

- Kmeans++ is an algorithm for choosing the initial cluster's centre values or centroids for the Kmeans clustering algorithm
  - K-means++ starts with allocating one cluster centre randomly and then searches for other centres given the first one
    - Choose one centroid μ_1 uniformly at random from dataset
    - Let $D(x)$ be the shortest distance from a data point to the closest centroid we have already chose
    - Choose a new centroid from the dataset with probability of $\frac{D^2(x_i)}{\sum_i D^2(x_i)}$
    - Now repeat previous step until we have initialised K centroids

Kmeans with Kmeans++

- Guarantee of Kmeans++
  - In Kmeans algorithm with random initialisation of centroids, the objective function monotonically decreases with each iteration of the algorithm
  - Let us say, for the best solution, the objective function takes value $j_{optimum}$
  - Let us say, when using Kmeans the objective function converges to $j_{optimum}$
  - For Kmeans with random initialisation there is no theoretical bound on

$$\frac{j_{converged}}{j_{optimum}}$$

  - In contrast, Kmeans++ initialisation has the following theoretical guarantee on convergence

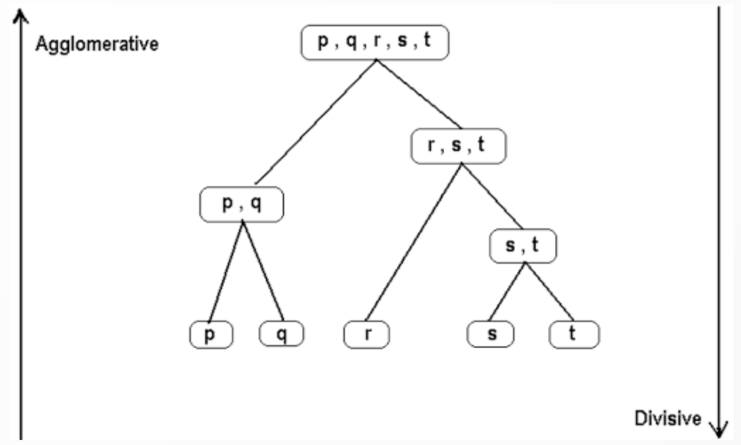
$$\frac{E\left(j_{converged}\right)}{j_{optimum}} \leq 8(logK + 2)$$

Other Clustering Algorithms...

- Hierarchical Clustering:
  - clusters that have a predetermined ordering
  - Two types -
    - **Agglomerative Clustering (Bottom-up)**
      - each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy
    - **Divisive Clustering (Top-down)**
      - all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy

Other Clustering Algorithms...

- Agglomerative Clustering:
  - At the bottom of the tree, at the starting point each of the characters p,q,r,s,t are assigned into a single separate cluster
  - As we go up to the higher levels, the closest characters are formed another cluster. i.e. s,t and p,q.
  - At the next level we can notice r,s,t are making a cluster
  - And finally at the top of the tree, all the characters are in one single cluster p,q,r,s,t
  - How to find the closest cluster pairs? i.e. how to find the distance between two sub-clusters in the middle of the tree?



Other Clustering Algorithms...

- Agglomerative Clustering: four ways to find distance -
  - Single-link: It is a distance between closest points
  - Complete-link: Distance between the furthest points
  - Centroid: Distance between the Centroids
  - Average-link: Average distance between pairs of elements from across cluster pairs

Other Clustering Algorithms...

- Divisive Clustering
  - Same as the Agglomerative Clustering in this type of clustering, initially all data instances are put in the same cluster
  - For splitting, we can use any clustering algorithm that produces at least two clusters (e.g. Kmeans) to find 2 clusters
  - The process is continued until each data instance is separate

Thank You.