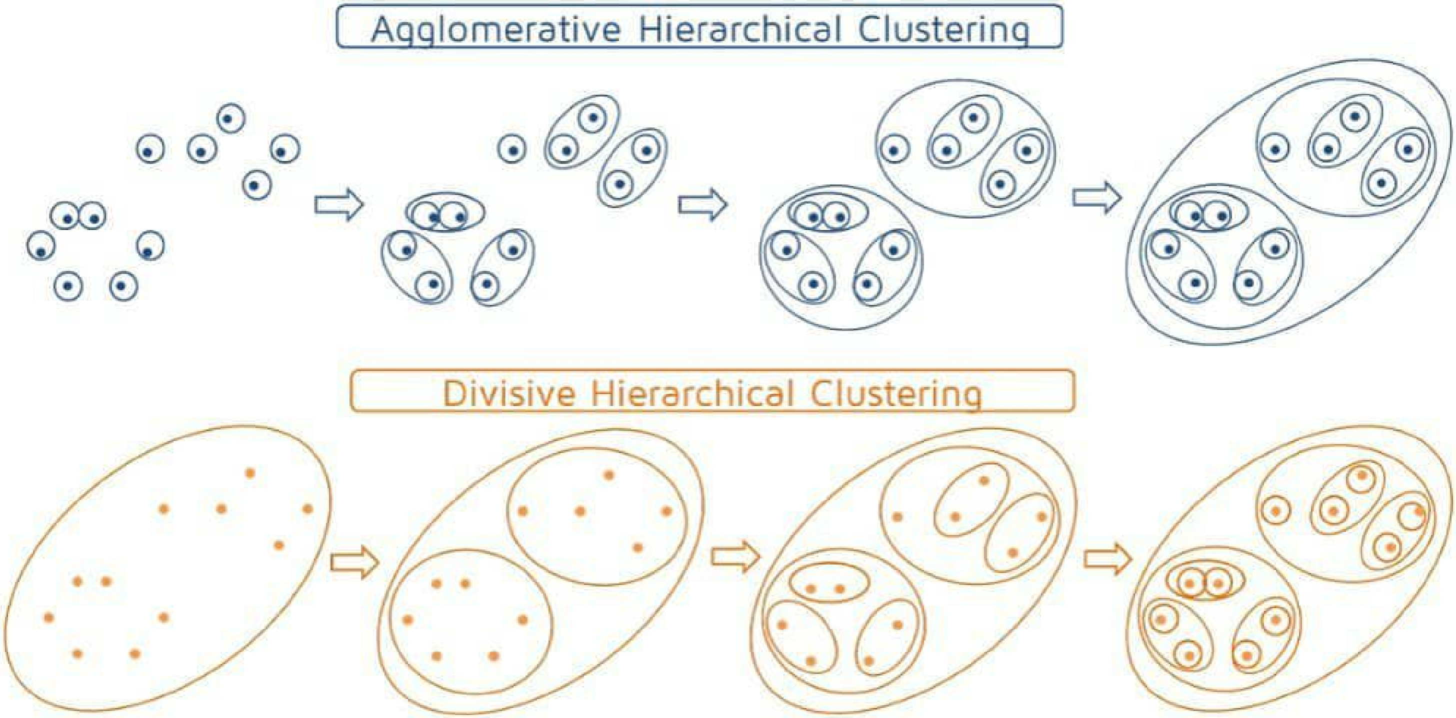
# Hierarchical clustering

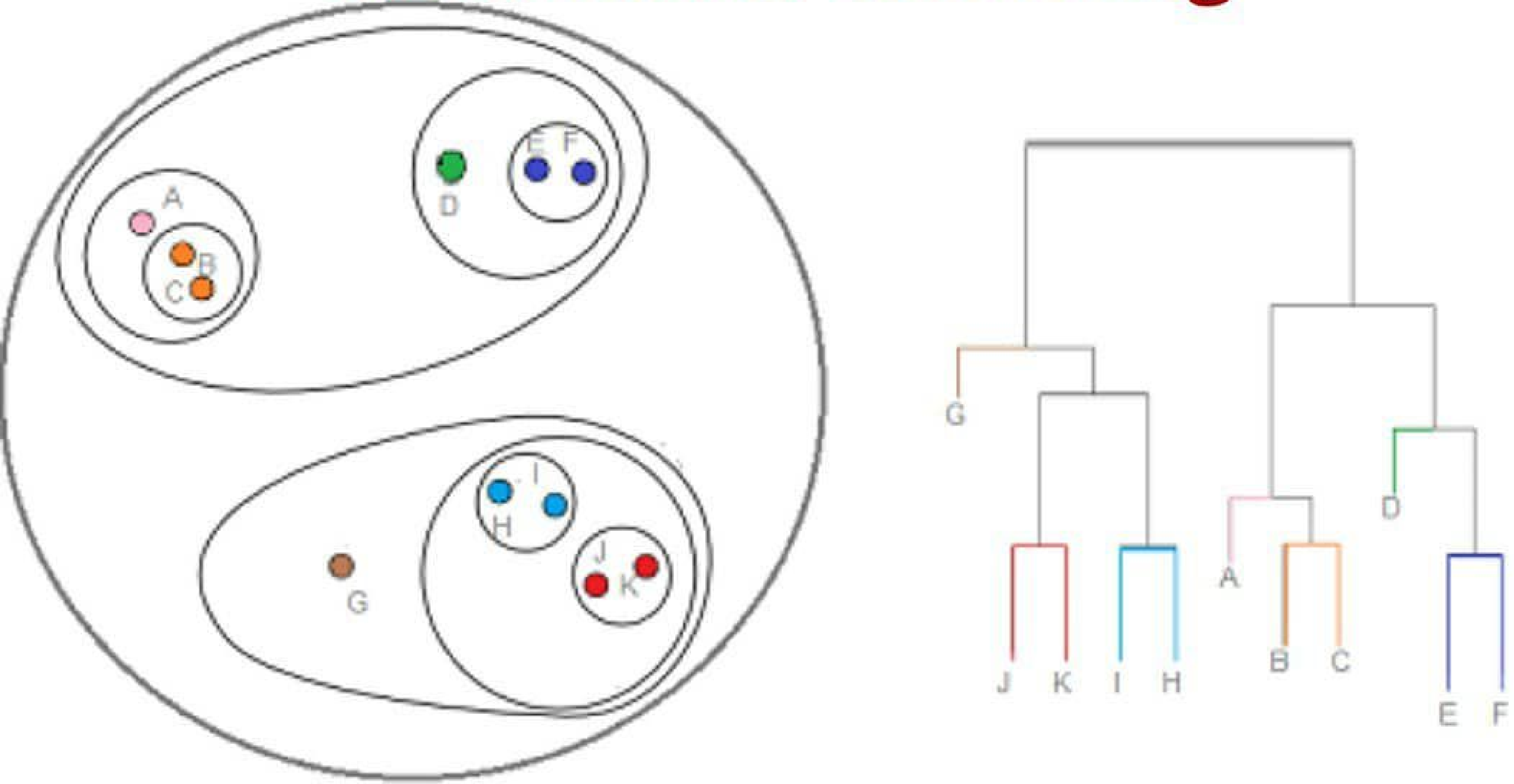
- Two types of hierarchical clustering
- **Agglomerative**: It starts by considering each data point as an individual cluster and in each iteration a pair of similar clusters are merged together until we get a single cluster or K clusters.
- **Divisive**: It is quite opposite to agglomerative, start with one cluster and in each iteration we split the cluster until we get each point as a cluster.
- Traditional hierarchical algorithms use a similarity or distance matrix.
- The main output of Hierarchical Clustering is a dendrogram, which shows the hierarchical relationship between the clusters

# Hierarchical clustering




@learn.machinelearning

# Hierarchical clustering




- **Dendrogram**: A tree like diagram that records the sequences of merges or splits.
- The heights in dendogram shows how similar they are. You can see E and F are at same height and both are similar compared to other.

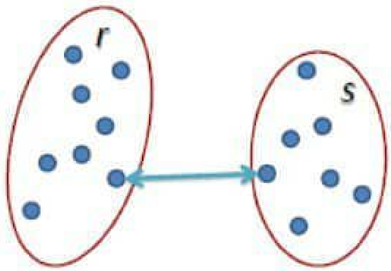
# Hierarchical clustering

- **Steps**
- Compute the proximity matrix (This stores the distances between each point)
- Let each data point be a cluster
- Repeat: Merge the two closest clusters and update the proximity matrix
- Until only a single cluster remains

# Hierarchical clustering

- How do we calculate the similarity between the points? There are many ways.
- **Complete linkage (Max):** similarity of the farthest pair. One drawback is that outliers can cause merging of close groups later than is optimal.
- **Single-linkage (Min):** similarity of the closest pair. This can cause premature merging of groups with close pairs, even if those groups are quite dissimilar overall.
- **Group average:** similarity between groups.
- **Centroid similarity:** each iteration merges the clusters with the most similar central point.
- **Ward's Method**: This approach of calculating the similarity between two clusters is exactly the same as Group Average except that Ward's method calculates the sum of the square of the distances

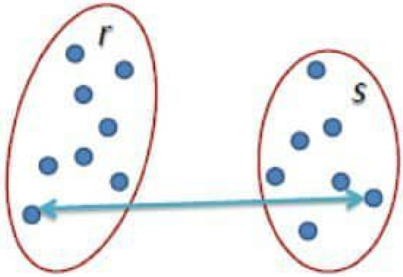
# Hierarchical clustering




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

- **Single linkage (min): The distance between two clusters is defined as the shortest distance between two points in each cluster.**
- **For example, the distance between clusters "r" and "s" to the left is equal to the length of the arrow between their two closest points.**
- **Pros: This approach can separate non-elliptical shapes if the gap between two clusters is not small.**
- **Cons: MIN approach cannot separate clusters properly if there is noise between clusters.**

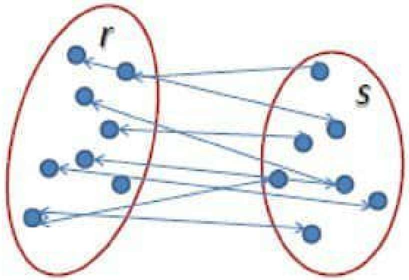
# Hierarchical clustering




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

- **Complete linkage (max)**: The distance between two clusters is defined as the longest distance between two points in each cluster.
- For example, the distance between clusters “r” and “s” to the left is equal to the length of the arrow between their two furthest points.
- **Pros**: MAX approach does well in separating clusters if there is noise or outliers.
- **Cons**: Max approach is biased towards globular clusters and tends to break large clusters.

# Hierarchical clustering

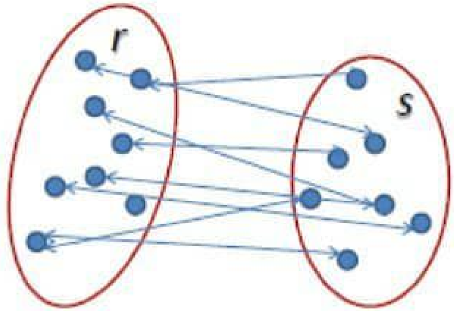



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

- **Group average (AVG):** The distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.
- For example, the distance between clusters “r” and “s” to the left is equal to the average length each arrow between connecting the points of one cluster to the other.
- **Pros:** The group Average approach does well in separating clusters if there is noise between clusters.
- **Cons:** The group Average approach is biased towards globular clusters.

@learn.machinelearning

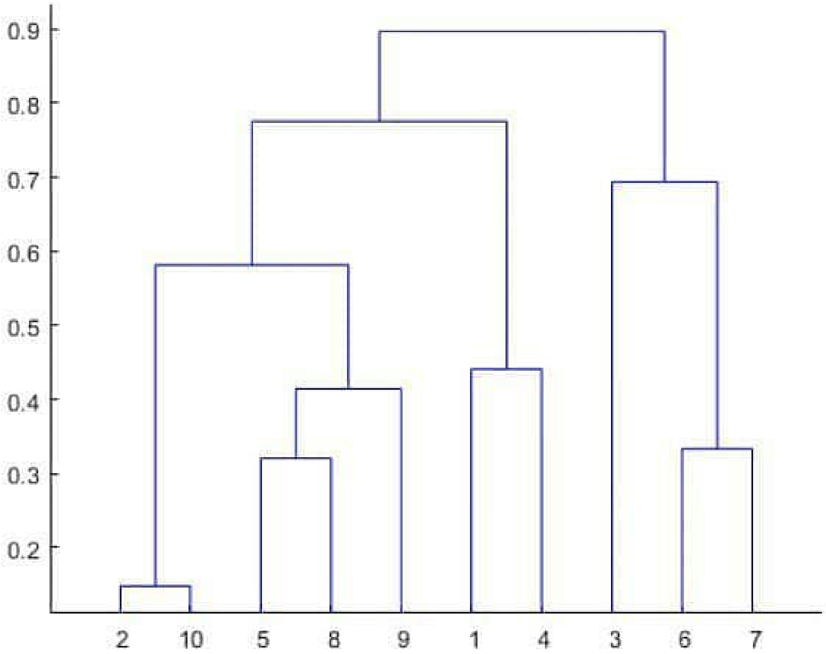
# Hierarchical clustering




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

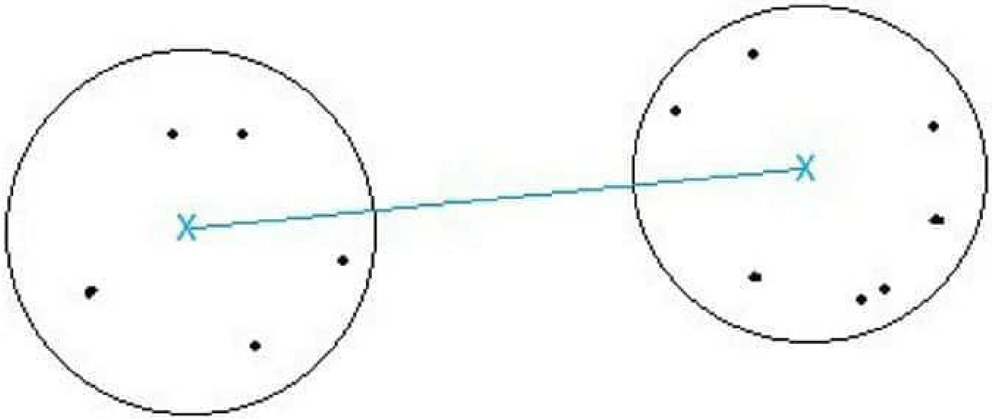
- **Ward's Method:** This approach of calculating the similarity between two clusters is exactly the same as Group Average except that Ward's method calculates the sum of the square of the distances Xi and XJ.
- Similarity(R,S)= ∑ sim(Xi, Xj)/|R|*|S|
- **Pros:** Ward's method approach also does well in separating clusters if there is noise between clusters.
- **Cons:** Ward's method approach is also biased towards globular clusters.

# Hierarchical clustering




- **How to choose no of clusters**
- To get the number of clusters for hierarchical clustering, we make use of an awesome concept called a Dendrogram.
- More the distance of the vertical lines in the dendrogram, more the distance between those clusters.

# Hierarchical clustering



- **Centroid similarity:** Compute the centroids of two clusters R & S and take the similarity between the two centroids as the similarity between two clusters. This is a less popular technique in the real world.

# Hierarchical clustering

- **Limitations**
- There is no mathematical objective for Hierarchical clustering.
- All the approaches to calculate the similarity between clusters has its own disadvantages.
- High space and time complexity for Hierarchical clustering. Hence this clustering algorithm cannot be used when we have huge data.