# K-Mean Clustering

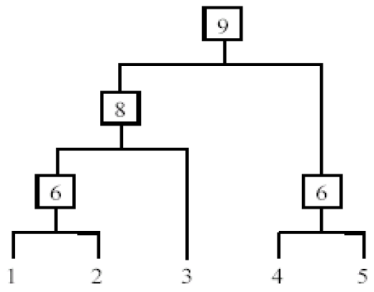
## Clustering

- Finding similarity groups in data, called **clusters**. i.e.,
  - Data instances that are similar to (near) each other are in the same cluster
  - Data instances that are very different (far away) from each other fall in different clusters.

## Different ways for clustering:

- **Hierarchical** approach: Create a hierarchical decomposition of the set of data (or objects) using some criterion (Wald)
- **Partitioning** approach: Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors (K-means, Spectral clustering)
- **Model-based** methods: A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other (EM)

### Hierarchical Clustering




Decomposes data objects into several levels of nested partitioning (tree of clusters).

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

## Agglomerative Clustering (Hierarchical)

- Assign each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item.
- Merge most similar clusters into a single cluster, so that now you have one less cluster.
- 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.

# K-Mean

- K-means is a partitioning based clustering algorithm as it partitions the given data into *k* clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - *k* is specified by the user

The k-means algorithm belongs to the category of **prototype-based clustering**. Prototype-based clustering means that each cluster is represented by a prototype, which can either be the **centroid** (*average*) of similar points with continuous features, or the **medoid** (the most *representative* or most frequently occurring point) in the case of categorical features.

In K mean clustering our goal is to group the samples based on their feature similarities, which can be achieved using the k-means algorithm that can be summarized by the following four steps:

1. Randomly pick *k* centroids from the sample points as initial cluster centres.
2. Assign each sample to the nearest centroid $\mu^{(j)}, j \in \{1, \ldots, k\}$.
3. Move the centroids to the centre of the samples that were assigned to it.
4. Repeat steps 2 and 3 until the cluster assignments do not change or a user-defined tolerance or the maximum number of iterations is reached.

Now, the next question is *how do we measure the similarity between objects*?

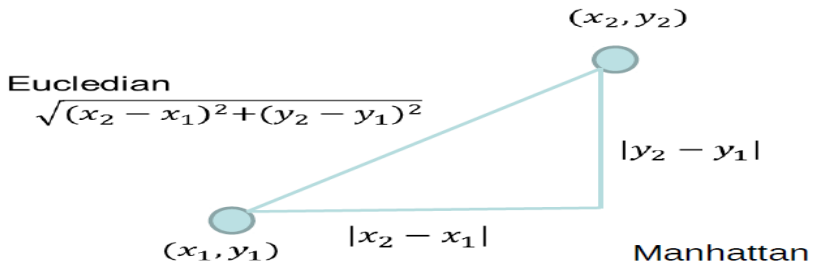
Note:

- If $d_1$ is near $d_2$, then $d_2$ is near $d_1$.
- If $d_1$ near $d_2$, and $d_2$ near $d_3$, then $d_1$ not far from $d_3$.
- No document is closer to $d$ than $d$ itself.

## What do we mean by distance *between* clusters?

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: average distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $dis(K_i, K_j) = dis(C_i, C_j)$

Mostly used ways to calculate distance are as follow:




## Some Other Common Metrics

- Weighted distances
  - More important variables get higher weights
- Minkowski distance
- The maximum distance amongst all attributes
- Correlation between rows

## Stopping/Convergence Criterion

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

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

$C_i$ is the $j$th cluster, $\mathbf{m}_j$ is the centroid of cluster $C_j$

# Stability Check of the Clusters

- To check the stability of the clusters take a random sample of 95% of records.
- Compute the clusters.
- If the clusters formed are very similar to the original, then the clusters are fine.