Cluster analysis

# Partitioning methods

**In general:** Distance-based, heuristic methods(k-means etc), spherical clusters, small/medium data sets.

**Characteristics:** Every data point must be in a cluster. Objective of the algorithms is to minimize the sum of squared distance within each cluster. Need to be specified beforehand how many clusters to be found.

**Methods:** K-means. K-medoids/PAM. Clara. Clarans.

## K-Means

The method

* The quality of a cluster is measured by the squared euclidean distance between the points in the cluster ("within cluster variation")
* Tries to make clusters as compact and separable as possible
* The algorithm iteratively lowers the within clusters variation
* Alogrithm: Assigns objects to clusters according to initial centroids -> Updates mean centroids ->Assigns objects to the right cluster -> Repeats until clusters remains unchanged after the iteration
* Often finds local minima so recommended to test different initial centroids
* Linear complexity
* Works best for compact and well separated clusters

Strengths

* Easy to run
* Finds spherical and convex shaped clusters
* Guarantees to find the local minimum
* Is a relatively scalable method and quite efficient in processing large data sets (naah?)

Weaknesses

* Unable to handle noisy data and outliers since all points must belong to a cluster. The outliers may impact the mean values a lot and the clustering will suffer.
* Only suitable for speherical or convex shaped clusters, or when clusters are of very different size.
* Cannot guarantee that the global minimum is found, in most cases it won´t
* Results depends on the initial centroids
* Sensitive to the choice of *K* and the initial centroids
* Need to specify number of clusters is a clear disadvantage of the method
* Not good for categorical data

## K-medoids

### PAM

The method

* Algorithm: Randomly select medoids -> Swap points and compare which point gives the lowest swapping cost -> These k points are the medoids -> Assigns points to the mediods
* Need to compare TC, swapping cost, for every pair of medoids and non-medoids
* The swapping cost is the difference of the sum for the current medoid and the tested medoid. If the cost is negative a better medoid is found.
* Minimizes the distance within each cluster (K-means minimizes squared distance)

Strengths

* A way to avoid the sensitivity to noise and outliers compared to K-means
* Finds spherical, convex clusters well.

Weaknesses

* Does not scale well, only works well for small and medium data sets
* Requires the number of clusters to be specified beforehand
* A computational costly algorithm for large data sets. The complexity is quadratic.
* Just finds cluster of spherical or convex form and of similar size.

### CLARA

The method

* Chooses medoids by using a sample of the entire data set
* Tests this for five times, then tests the medoids and chooses the best medoids
* Same complexity as before times five, but on a smaller data set.
* The efficiency depends on the sample size
* It is not sure that a sample will return good results when applied on the whole data set
* A way to scale up the PAM method, make it applicable for large data sets

### CLARANS

The method

* Uses the whole data set but is an greedier algorithm than PAM
* Is a trade-off between the computational cost and the effectiveness of using sample to obtain clustering
* Specifies number of local minima to be found to be found and maximum number of neighbors to compare
* If it finds a medoid with a negative swapping cost, it goes there without checking the others. Goes to the first better solution directly.
* Has better scalability -> Works better in high dimensions

# Hierarchical methods

**In general:** Distance-, density or continuity-based. Agglomerative or divisive.

### Distance measures

* Single link. Hierarchical clusters defined by local proximity.
* Complete link. Really useful if true clusters are rather compact and of approximately equal size. Tends to find clusters opting for global closeness.
* Both complete link and single link represents extremes in measuring the distance between clusters. Both tent to be overly sensitive to outliers and noisy data.
* Average link. Can handle both numerical and categorical data.
* Mean link.
* Mean and average link are compromises of single and complete link and are not as sensitive to outliers.

### Agglomerative

* From scratch all points forms their own cluster
* At every step are the closest clusters merged
* Gives a hierarchy of clusters
* Need to specify the distance between the clusters
* Is called AGNES
* Requires at most n iterations

### Divisive

* Starts with all points in one cluster
* Both methods has a time complexity of O()
* There are possible ways to partition a set of n objects into two exclusive subsets. For large data sets it becomes computationally prohibitive to examine all possibilities.
* More common to use agglomerative methods than divisive because of the computational burden and need for heuristic approaches.
* Suffers from not being able to undo a performed split, once a step is done it can never be undone. Is true for both divisive and agglomerative.
* Find the on average most dissimilar point. Compute

### BIRCH

The method

* Designed for clustering a large amount of numerical data
* Integrates hierarchical clustering with other clustering methods
* Overcomes the two main difficulties in agglomerative clustering methods: Scalability and inability to undo what was done in the previous step
* "Clustering feature" -> Summarize a cluster, is a summary of the statistics for a given cluster. For example the centroid, the radius and the diameter is of interest.
* "Clustering feature tree" -> Represent a cluster hierarchy
* Has a structure that makes the method effective for incremental and dynamic clustering of incoming objects.
* R and D (radius and diameter) reflects the tightness of the cluster around the centroid

# Density-based methods

**In general:** Can find clusters of arbitrary shapes.