TECHNISCHE UNIVERSITÄT DRESDEN

Dresden Database Systems Group

Clustering

Programming for Data Science

Definition

Dresden Database Systems Group

Goal of clustering

- Partition a set of objects/data into groups –so called clusters— so that objects in the same cluster are as similar as possible and objects in different clusters are as dissimilar as possible.

Properties of clusters

- Clusters can vary in size, shape, and density
- Disjunct vs. overlapping
- Deterministic vs. probabilistic
- Hierarchical vs. flat

Clustering by Minimization of Variance

Dresden Database Systems Group

Algorithm characteristics

- converges against a (possibly only local) minimum
- complexity: O(*n*) for one iteration
- Number of iterations is typically small (~5-10)

Further hard to quantify properties

- Sensitivity to noise / outliers
- Convex cluster shape
- Result and runtime strongly depend on initialization (first random segmentation)

K-Means

Dresden Database Systems Group

Variant of variance minimizing algorithm

- Centroids are re-calculated whenever an object changes cluster affiliation
- Two main Versions: Lloyd & MacQueen
- ***Lloyds Algorithm***: see next slide
- ***MacQueen's Algorithm***:
  Incremental adaption, when a point p changes from Cluster C_1 to Cluster C_2

$$\bar{x}_j(C_1') = \frac{1}{n_{C_1} - 1} \cdot (n_{C_1} \cdot \bar{x}_j(C_1) - x_j^p) \quad \text{und} \quad \bar{x}_j(C_2') = \frac{1}{n_{C_2} + 1} \cdot (n_{C_2} \cdot \bar{x}_j(C_2) + x_j^p)$$

- n_C = number of objects in C

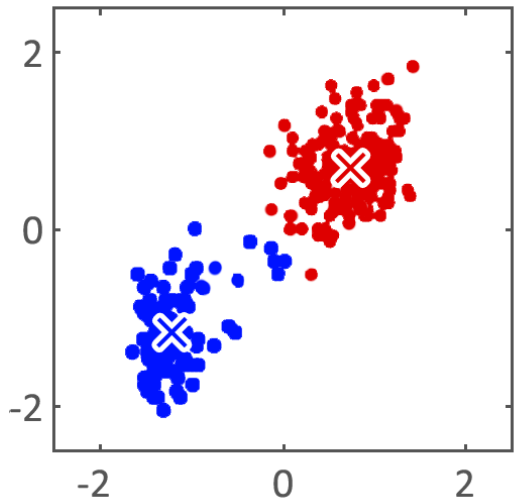
Remarks

- Most common clustering method (by far)
- complexity: O(nkt), n: #objects, k: #clusters, t: #iterations , typically: k<<n and t<<n
- k-means is basically variance minimizing, and strongly depends on the processing order

kMeans - Lloyds Algorithm

Dresden Database Systems Group

```
ClusteringByVarianceMinimization(data set, k)
Create random initial k cluster centroids C'={C1, ..., Ck};
C = {};
repeat until C = C'
        C = C';
        Create k clusters, by assigning each point to its nearest centroid;
        Calculate C'={C'1, ..., C'k} for the newly defined clusters.
return C;
```

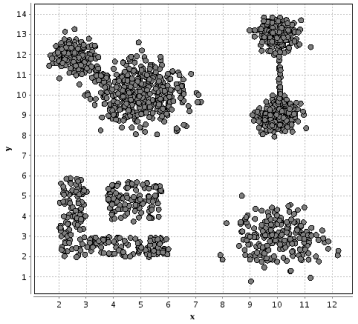


K-Means

Dresden Database Systems Group

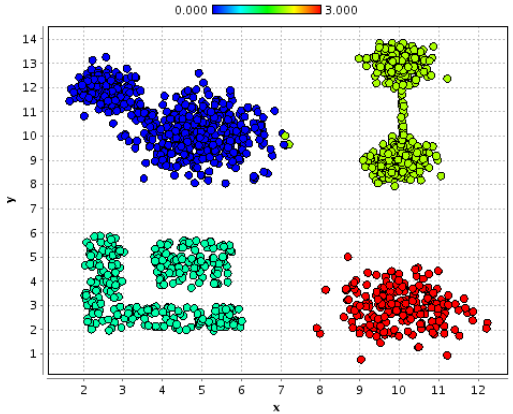
Example data

- d=2 (x-y-coordinates)
- Synthetically generated

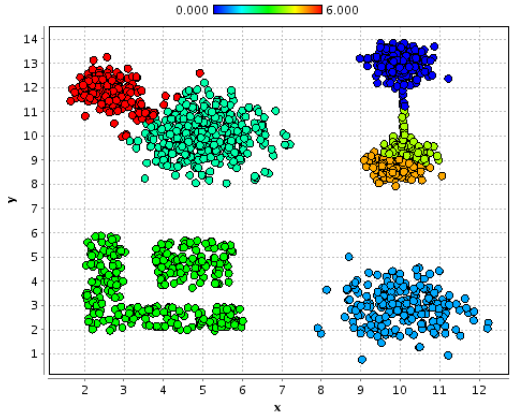


K-Means wit different k

k=4



k=7



Hierarchical clustering

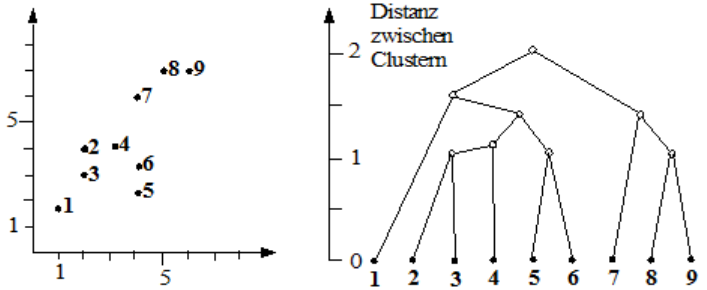
Dresden Database Systems Group

Goal

- Create a representation of the data, from which clusters can be derived

Dendrogram

- Merge clusters with minimal distance
- Tree, whose nodes represent clusters
  - Root contains whole data set
  - Leaves contain individual objects
  - Inner node is union of all objects contained in its child nodes



Dendrogram

Types

- Bottom-Up
  - Agglomerative
- Top-Down
  - Divisive

Agglomerative Hierarchical Clustering

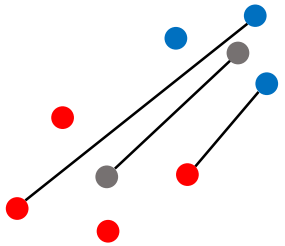
Dresden Database Systems Group

Algorithm

1. Create initial clusters, one object per cluster, and calculate the distance between all cluster pairs
2. Create new cluster by merging the two closest clusters
3. Calculate distances between new and old clusters
4. Stop if one cluster contains all objects: otherwise repeat from 2.

Distance functions for object sets (cluster)

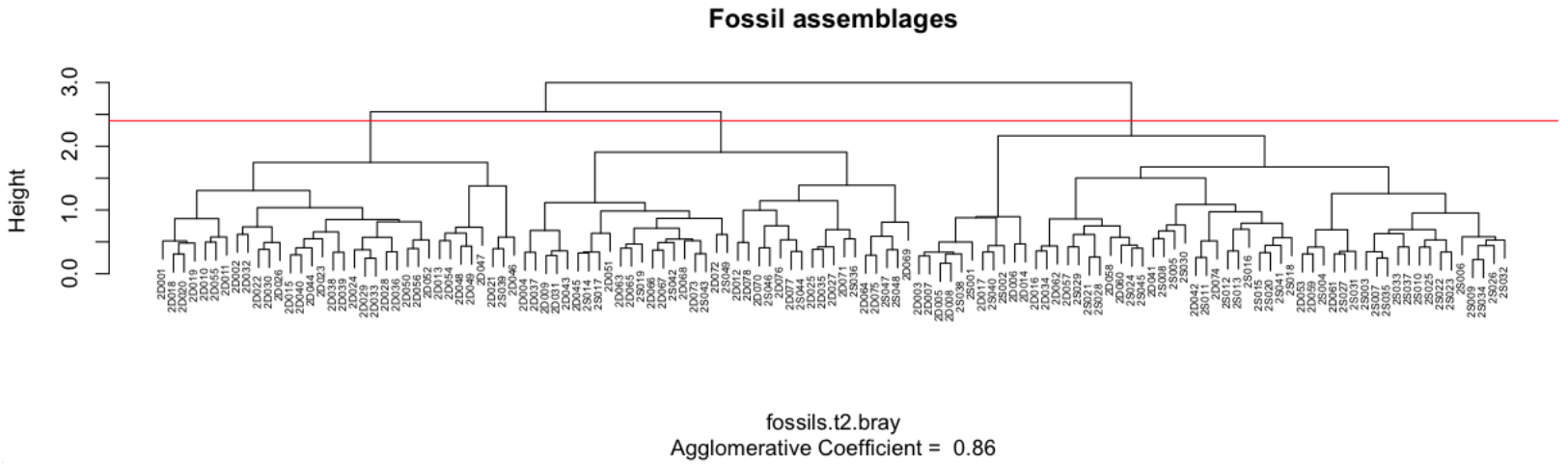
- Single-Link: $dist.sl(X,Y) = \min_{x \in X, y \in Y} dist(x,y)$
- Complete-Link: $dist.cl(X,Y) = \max_{x \in X, y \in Y} dist(x,y)$
- Average-Link: $dist.al(X,Y) = \frac{1}{|X| \cdot |Y|} \cdot \sum_{x \in X, y \in Y} dist(x,y)$
- Centroid-Link: $dist.ctl(X,Y) = dist(c_x, c_y)$



Agglomerative Hierarchical Clustering

Dresden Database Systems Group

Cutting out clusters



Agglomerative Hierarchical Clustering

Dresden Database Systems Group

Build the dendrogram „bottom-up“

Pros

- No knowledge on k
- Identifies hierarchies, not only flat clusters
- A clustering is derived from the dendrogram with an horizontal cut

Cons

- Sensitive to noise (Single-Link)
- Two clusters cannot be identified in the dendrogram if they are connected by a “line” of objects with similar inter-object distances
- Inefficient: runtime complexity $O(kn^2)$ for n objects and k levels
- No incremental update

Task

Dresden Database Systems Group

Step 0

- You will get a csv file from us. Load it in your language/environment.
- Explore the data in it.

Step 1

- Implement kMeans (Lloyd)*.

Step 2

- Implement AGNES*. Choose at least one link criterion.

*use your own implementation

Package suggestions

Dresden Database Systems Group

R

- (data.table)

python3

- numpy
- pandas

Exercise Appointment

Dresden Database Systems Group

We compare and discuss the results

- Tuesday, 08.12.2020,
- Consultation: Please use the forum in Opal.
- Please prepare your solutions! Send us your code!

If you have questions, please mail us:

claudio.hartmann@tu-dresden.de Orga + Task+ Code + R
lucas.woltmann@tu-dresden.de Python