Universität
Zürich UZH

Soziologisches Institut

Data Analysis – Advanced Statistics with Python

Dr. Julia Jerke

jerke@soziologie.uzh.ch

Thursday, 12.15pm – 13.45pm, AND 2.46

Session 7 – Cluster analysis

Agenda

1. Cluster analysis basics
2. K-means clustering method
3. Hierarchical clustering method
4. Hands on

1. Cluster analysis basics

Motivation

- So far, we focused on methods that check for a hypothesized structure in the data. Therefore, it is necessary to have an idea about the relationship between a dependent variable Y and a set of given variables $X_1, X_2, \dots, X_p$

- We will now turn towards methods that discover structure within a presumably unstructered data set. In this case, we do not make assumptions about (directed) relationships between variables.

We will discuss two methods in this context:

1. **Cluster analysis** (this session)
2. **Principal component analysis** (next session)

Overview

What is a cluster analysis?

- An exploratory method to identify structures (clusters) of similarity within data
- Cluster analysis is an *unsupervised machine learning* method (There is no Y that we are trying to predict!)

Basic idea

- Observations are clustered along their values for a set of relevant variables $X_1, X_2, \ldots, X_p$
- Groups are identified in such way that subjects are more similar within the groups than between groups
- In principle, the procedure involves two steps:
    1. Calculation of the distances between the individual observations
    2. Formation of clusters based on the distances
- There are usually multiple cluster solution and we rely on heuristics to pick the optimal one

Distances and similarity

- **Distance = a measure for dissimilarity**
- Basis of a cluster analysis is the distance d_{ij} between each pair of observations $i, j \in \{1, \dots, n\}, i \neq j$
- The distance are collected in a $n \times n$ distance matrix
- There are various ways to calculate the distance between observations, however, the general procedure remains the same:
  - Choose a measure for the distance $d_{ij,p}$ between two observations i and j with respect to a specific variable $X_k \in \{X_1, \dots, X_p\}$ (these measures usually depend on the scale of the variable)
  - Calculate the distances between i and j for all variables $X_1, \dots, X_p$ and aggregate them to a measure for the total distance d_{ij} between i and j (in most cases, we will compute the sum)

Distances and similarity

- **Continuous variables**

$$x_i = \begin{pmatrix} 12 \\ 3 \\ 38 \\ 20 \end{pmatrix} \text{ and } x_j = \begin{pmatrix} 10 \\ 7 \\ 56 \\ 45 \end{pmatrix}$$

- *City block distance* (also manhattan distance or taxicab distance): the sum of the absolute differences between the values for $X_1, \dots, X_p$

$$d_{ij} = \sum_{l=1}^{p} |x_{i,l} - x_{j,l}| = 49$$

- *Euclidian distance*: the square root of the sum of squared differences between the values for $X_1, \dots, X_p$

$$d_{ij} = \sqrt{\sum_{l=1}^{p} (x_{i,l} - x_{j,l})^2} = 969$$

Distances and similarity

➢ **Categorical variables**

$$x_i = \begin{pmatrix} male \\ East \\ yes \\ Teacher \end{pmatrix} \text{ and } x_j = \begin{pmatrix} female \\ East \\ no \\ Lawyer \end{pmatrix}$$

- The distance is usually determined by evaluating whether the values for x_i and x_j are the same or not:

$$d_{ij,l} = \begin{cases} 1, & \text{if } x_{i,l} = x_{j,l} \\ 0, & \text{if } x_{i,l} \neq x_{j,l} \end{cases}$$

- The total distance is then calculated as the proportion of values that are not the same (M coefficient):

$$d_{ij} = 1 - \frac{\sum_{l=1}^{p} d_{ij,l}}{p} = 0.75$$

➢ **Ordinal variables**

- The distance is usually based on the differences between the respective ranks

Goodness of a cluster solution

Note: The scikit-learn library uses the term *inertia* for the SSE

Once we have found a reasonable cluster solution (hence a partition of our N observations in K clusters), how can we determine the quality of our cluster solution?

Sum of the Squared Error (SSE)

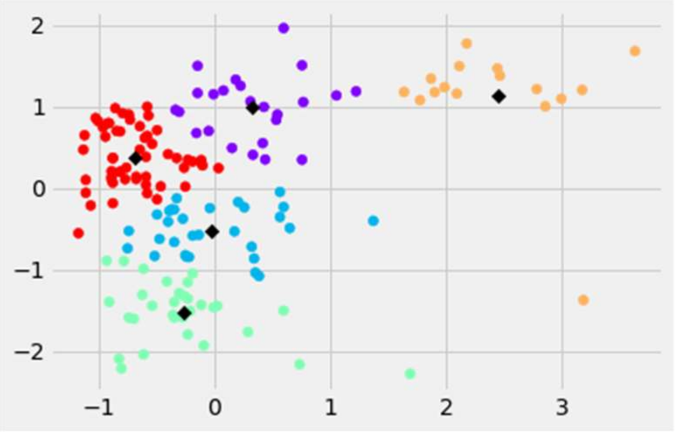
- The SSE is defined as the total sum over all clusters of the squared Euclidian distance between all observations in the clusters and the respective cluster means (also called centroids)

$$\sum_{k=1}^{K} \sum_{\forall i \in C_k} d_{i,\bar{x}_k}$$

whereas:

- C_k is cluster number k
- $d_{i,\bar{x}_k}$ is the squared Euclidian distance between an observation in cluster C_k and the respective cluster centroid:

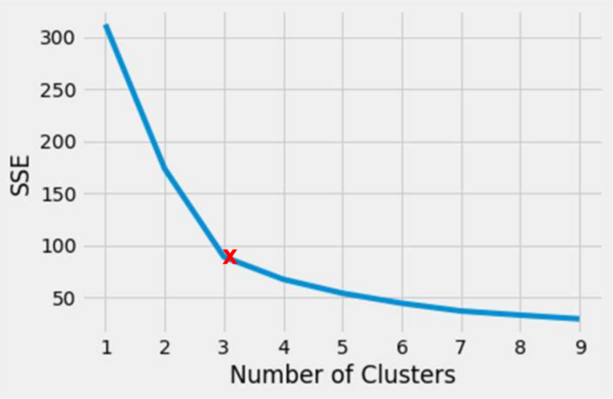
$$d_{i,\bar{x}_k} = \sum_{l=1}^{p} (x_{i,l} - \bar{x}_l)^2$$



Goodness of a cluster solution

Sum of the Squared Error (SSE)

- The SSE is a relative value, it is not limited to a fixed range
- The goal is to find a cluster solution that minimizes the SSE or at least has a low SSE
- *Note*:
  - The SSE always increases with the number clusters since the number of centroids increases as well
  - The SSE will be minimal (=0), when $N = K$, hence when each observation builds an own cluster
- We have to find a trade-off between the SSE and the number of clusters
- Often, an *elbow plot* can help
- The elbow plot plots the number of clusters and the respective SSE
- Heuristic: choose the cluster solution at the «elbow»



Overview of clustering methods

1. Partitional clustering
   - **K-means clustering**
   - K-medoids clustering
2. Hierarchical clustering
   - **Agglomerative clustering (bottom-up)**
   - Divisive (top-down)
3. Density-based clustering
   - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
   - OPTICS (Ordering Points To Identify the Clustering Structure)

2. K-means clustering

K-means clustering

Belongs to the partitional clustering strategies

Basic idea

- Start with an arbitrary cluster solution
- Replacement method: Try to successively improve the solution by changing the assignment of the individual observations to clusters
- Goal:
  - Maximize heterogeneity between clusters
  - Maximize homogeneity within clusters

K-means clustering

Algorithm

1. Determine k as the number of clusters that your solution should have
2. Start with a random cluster solution by randomly determining k centroids (cluster means) and assigning each observation $\mathrm{i} \in \{1, \dots, n\}$ to the closest cluster
3. Calculate the new centroids for each cluster
4. Reorganise your clusters in such way that each observation is now assigned to the cluster for which the centroid is closest
5. Repeat step 4 and 5 until the cluster solution does not change anymore (i.e. converges)

Finding the optimum solution

Problem

- The final solution always fulfills *"minimum distance"*:

  The squared euclidean distances of the observations to their own cluster centroids are always smaller than the distances to other cluster centroids.
- *But*: the solution does not necessarily minimize the variance within the clusters

Reason… the procedure is *nondeterministic*!

- The solution is a local optimum on the basis of a random initial cluster solution
- With different initial cluster solutions different local optimum will be identified

Solution

- Repeat the procedure with different initial cluster solutions
- Select the solution with the smallest inner-class variance

Strengths and weaknesses of *K*-means clustering

Strengths

- Perform well for large sample sizes
- They work well when clusters have a spherical shape
- They're scalable with respect to algorithm complexity

Weaknesses

- Necessary to specify the number of clusters in advance
- They're not well suited for clusters with complex shapes and different sizes
- They break down when used with clusters of different densities

3. Hierarchical (bottom-up) clustering

Hierarchical clustering

Basic idea

- A large number of hierarchically sequential cluster solutions are generated (altogether n solutions for n observations in the sample)
- Using certain heuristics one of these cluster solutions is then selected

Two types

- *Agglomerative clustering (or bottom-up approach)*:

  Starting with all observations in individual clusters, subsequently merge the two observations that are the most similar until all observations have been merged into a single cluster
- *Divisive clustering (or top-down approach):*

  Starting with all observations in one cluster, subsequently split the least similar clusters at each step until only the individual observations remain

Hierarchical clustering - procedure

Procedure

1. n clusters: each observation creates its own cluster
2. n-1 clusters: find those two clusters with the smallest distance to each other and combine them to a new cluster
3. n-2 clusters: select from the n-1 clusters those with the smallest distance to each other and combine them
4. n-3 clusters: ...

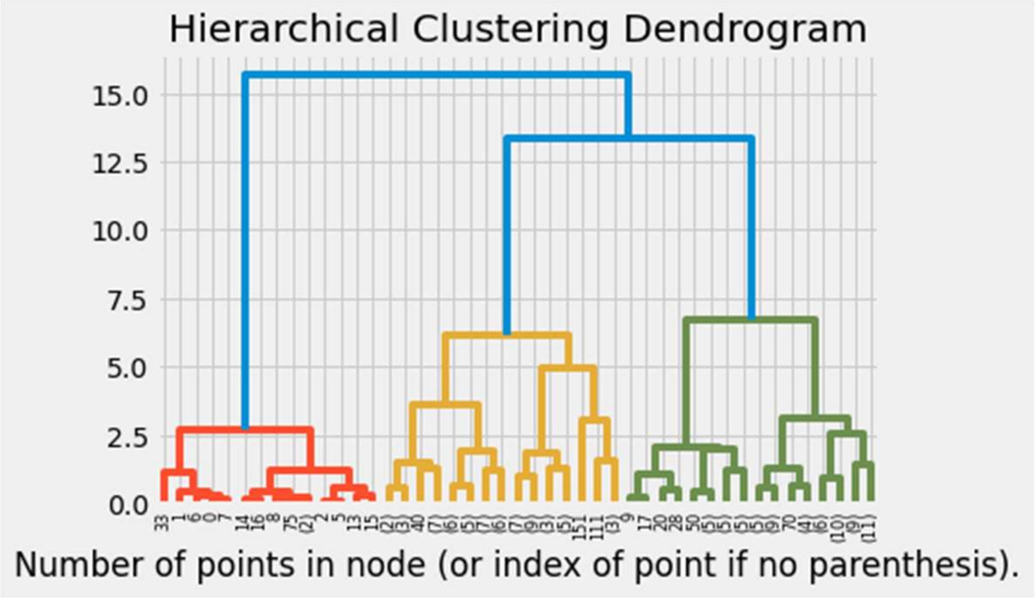
... (in each step the number of clusters is reduced by 1)

n. 1 cluster: Combine the last two large clusters into one, all cases are now in one cluster

- ➢ With each step the heterogeneity within the respective clusters increases
- ➢ **Determining the number of clusters: avoid the largest increase in heterogeneity**

Hierarchical clustering - dendogram

- Hierarchical clustering methods produce a tree-like hierarchy
- A **dendogram** is the visual illustration of the sequentially created cluster solutions
- In contrast to partitional clustering, hierarchical clustering is *deterministic*: running the analysis several times will always result in the same cluster solution
- To decide for the optimal cluster solution, the dendrogram is visually inspected
- A common heuristic to determine the number of clusters: **avoid the largest increase in heterogeneity**



Hierarchical clustering – measuring the distance

How is the distance between two clusters defined?

- **Single linkage**: calculate the smallest distance among all pairs of observations from each of the two clusters ("*Nearest Neighbor*”)
- **Complete linkage**: calculate the largest distance among all pairs of observations from each of the two clusters ("Furthest Neighbor”)
- **Average-Linkage**: calculate the mean value over all distance pairings of the two clusters

For metric variables:

- **Centroid method:**
  - Calculate the mean vector ("*centroid*") within each cluster
  - The distance of two clusters is then defined as the squared euclidean distance of their centroids
- **Ward method**: like centroid method, but with optimizing correction term

Strengths and weaknesses of hierarchical clustering

Strengths

- They often reveal the finer details about the relationships between data objects
- They provide an interpretable dendrogram
- No need to specify the number of clusters in advance

Weaknesses

- They're computationally expensive with respect to algorithm complexity
- Better suited for smaller sample size due to the visual inspection of cluster solutions
- Sensitive to noise and outliers

4. Hands on

… Open *Session_7_cluster_analysis.ipynb* in jupyter notebook