How to Optimize Gower Distance Weights for the k-Medoids Clustering Algorithm to Obtain Mobility Profiles of the Swiss Population

Alperen Bektas and René Schumann

HES-SO Valais / Wallis

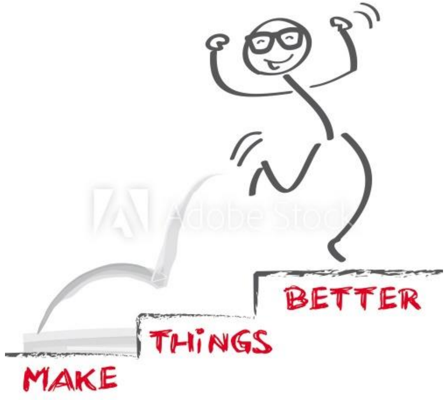
The 6th Swiss Conference on Data Science

Bern, 14th of June 2019

Content

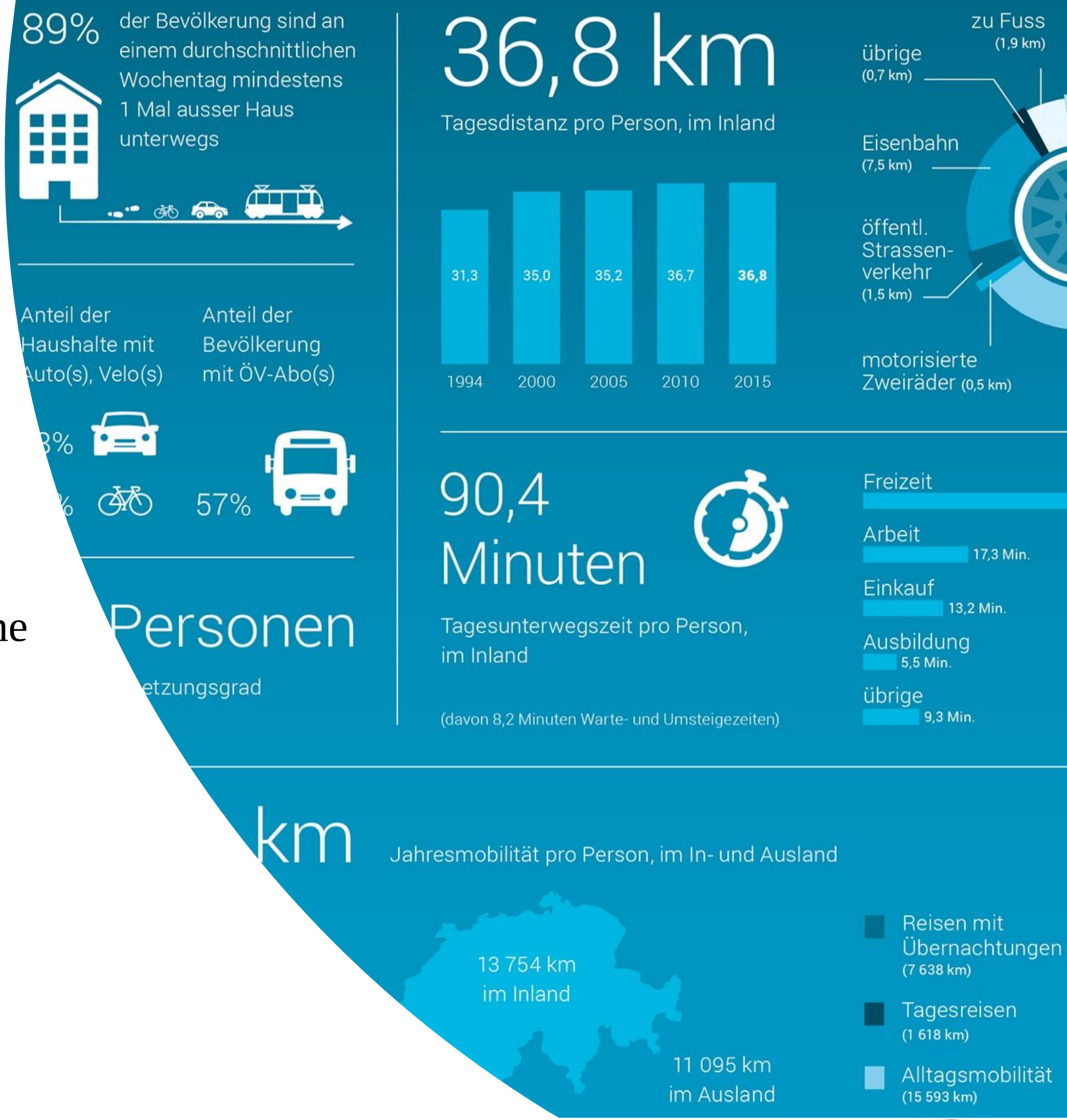
Introduction

- The goal: Obtaining mobility profiles of the Swiss population
- Respondents of empirical data (Census)
  - Mobility-related features of the respondents are ex-ante selected
- Clustering as methodology
  - Respondents who have similar mobility characteristics are placed in the same cluster
- Why not having better clusters? Can we improve quality?
  - Higher inter-cluster heterogeneity (separation)
  - Lower intra-cluster homogeneity (cohesion/similarity)



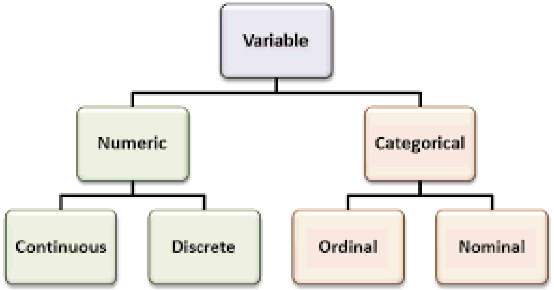
Empirical Data

- Mobility and Transport Micro-Census 2015
- Ex-ante feature selection (active/descriptive)
  - Mobility-related features are chosen
- Eliminating some active features
  - Remove highly correlated variables (measure the same thing)
  - Remove categorical variables in which a category is very dominant
  - Remove categorical features with too many levels



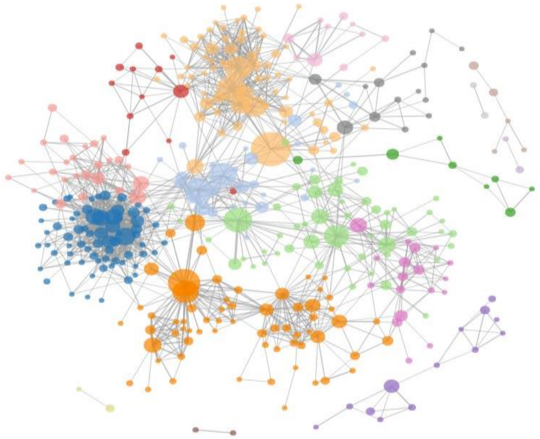
Empirical Data

- 6 active variables are used to determine positions in the latent space
  - Number of cars (in the household)
  - Has half-fare travel card (binary)
  - Number of daily trips
  - Daily distance (kilometers)
  - Modal-choice (car, train, walking, etc.)
  - Multi-modality (binary)
- Active variables are mixed-type (numeric/categorical)



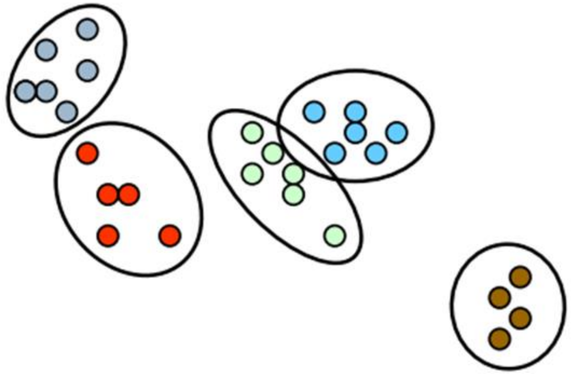
Multi Dimensional Social Space

- Respondents are placed in a Latent Space
- Distance (Dissimilarity) Matrix functions as the latent space
- Various metrics can handle it e.g. Euclidean
- Gower distance metric
  - Can handle mixed-type data sets
  - All variable has a weight (default all equals 1)
  - Weights can be tuned
  - Distances are normalized between 0-1
- Peer-wise distances (symmetric) determine the closeness
- According to the positions in this space, a clustering algorithm partitions them



$$A = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 & \dots & d_{1n}^2 \\ d_{21}^2 & 0 & d_{23}^2 & \dots & d_{2n}^2 \\ d_{31}^2 & d_{32}^2 & 0 & \dots & d_{3n}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & d_{n3}^2 & \dots & 0 \end{bmatrix}$$

k-Medoids (partitioning around medoids, PAM)



- Unsupervised partitioning algorithm
- Robust to outliers
- Finds a medoid (exemplar, representative) of each cluster
- Gets a latent space (distance matrix) and the number of clusters (k) as input
- Based on positions in the space, respondents are partitioned into k clusters
- In the end, clusters, intra-cluster distributions, and medoids are obtained

Average Silhouette Width (ASW)

- The number of clusters (k) should be pre specified
- How well an instance is matched with its own cluster
- A fitness measure that reflects how maximized intra-cluster homogeneity and inter-cluster dissimilarity
- K-value that has the highest ASW score is assigned as the optimal number of cluster

Optimization

- Tune default Gower weights
- **Optim** function in R language
- Function B minimizes the return of Function A
- Best weight combination that maximizes the ASW value of k clusters is obtained

---

Algorithm 2 Function A to find the ASW value of the clustering with the input weights

---

Input: weights
Output: ASW value

1: Calculate a Gower distance matrix with the weights
$Gower\_dist \leftarrow daisy(data,\ weights = weights)$
2: Partition the cases into k clusters
$clusters \leftarrow pam(Gower\_dist, k)$
3: Calculate the ASW value of the cases in the k clusters
$ASW\ value \leftarrow\ clusters\$silinfo\$avg.width$
4: **return** $-ASW\ value$

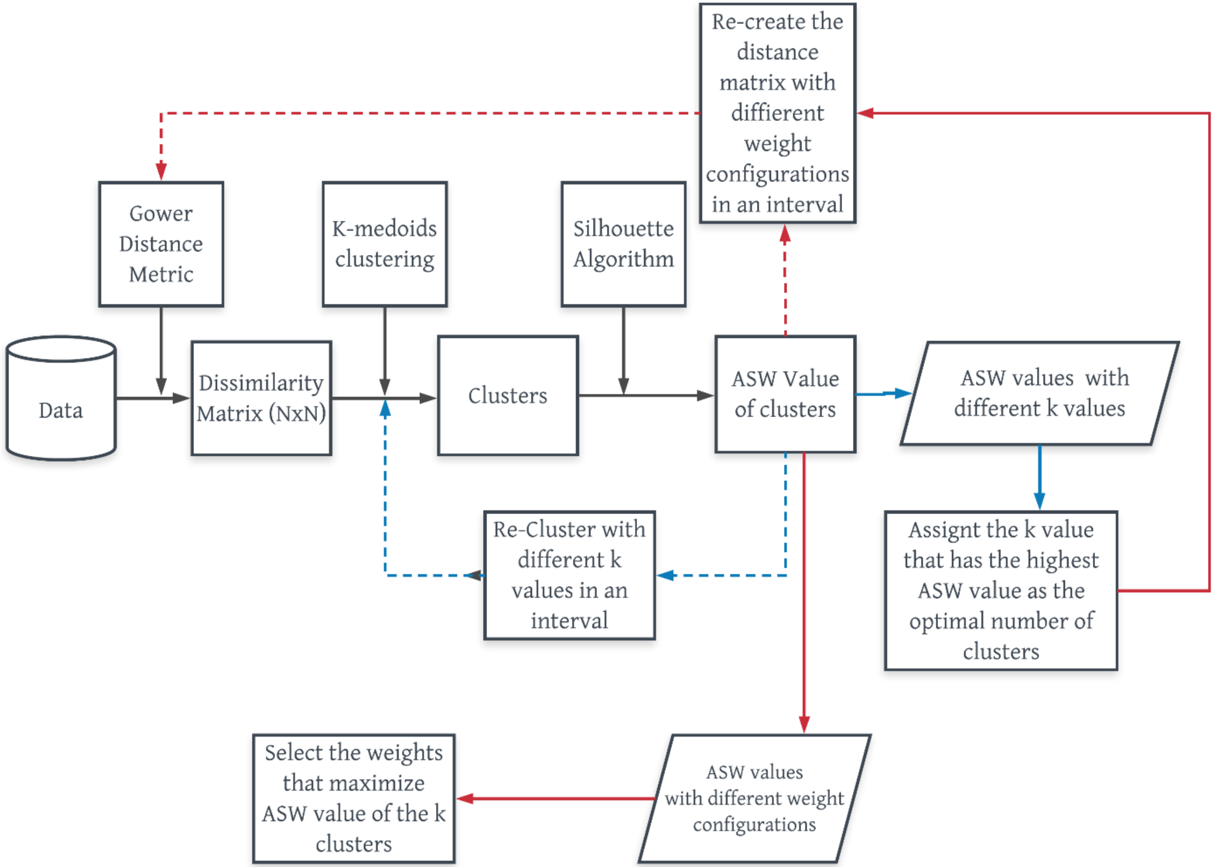
Algorithm 3 Function B to optimize the default weights

Input: default weights,upper bound(u), lower bound(l)
Output: optimized weights

1: Calculate the optimized weights through the optim function
$optimized\_weights \leftarrow optim(par = weights, fn = function\ A, lower = 1, upper = u, method =' L - BFGS - B')$
2: **return** $optimized\_weights$

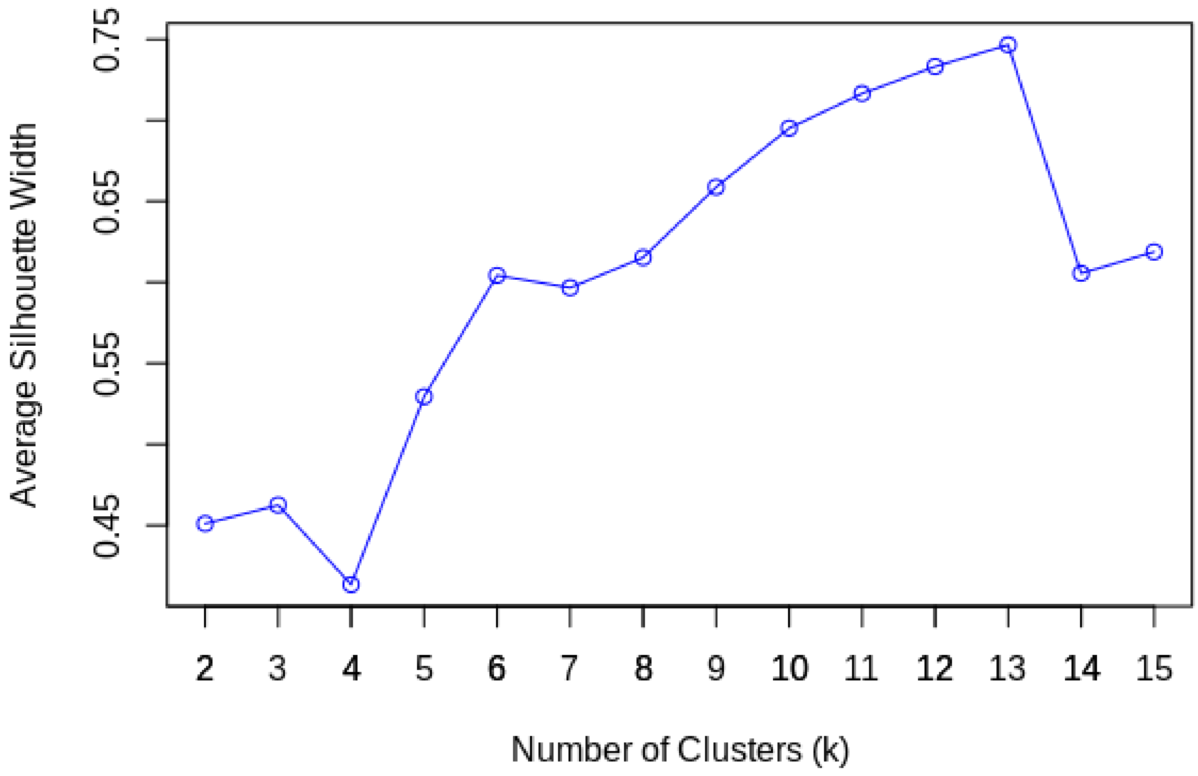
Overall Concept

- 1st step: the optimal number of clusters is obtained
- 2nd step: The ASW value of the optimal number of clusters (obtained in the first step) is improved through optimizing the default Gower weights.



Results-(1st step)

- The optimal number of clusters: 13 (ASW=0.7465)
- The second best: 12 (ASW=0.7300)
- Interval [2-15]



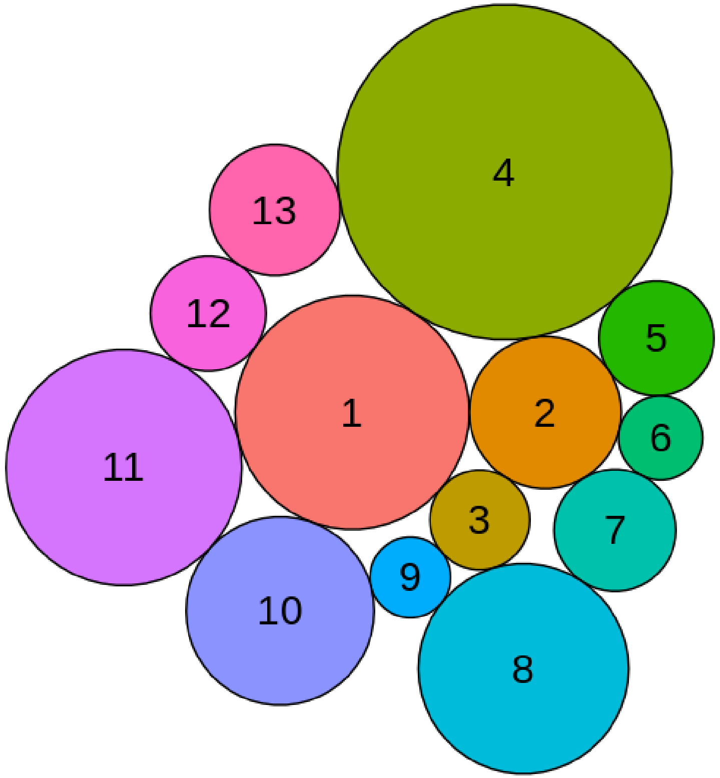
Results-(2nd step)

- Optimized Gower Weights
- New ASW value of 13 clusters 0.8458 (ex - 0.7465)
- New ASW value of the control 0.8349 (ex – 0,7300)

Features	**Optimized Weights**
Number of cars	1,000000
Has half-fare travel card	2,469693
Daily trips	1,000000
Daily distance	1,000000
Modal Choice	3,000000
Multimodality	2,640402

Results-(Clusters and Medoids)

- Private car: 4, 11, 8, 2
- Walker: 1, 10
- Train: 5, 2
- Bike / E-bike : 12, 7
- Bus: 9, 6
- Tram: 3



Limitations / Future Work

- Interval of k-values [2-15]
- Upper bound of the weights

- Challenging limitations
- Synthetic population generation
- Policy extractions (messages) over medoids / profiles

Questions