Marketing Analytics

Cluster Analysis

Daniel Winkler

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION OF MBAS AMBA ACCREDITED

Objectives

Learn

1. The basic concept of cluster analysis
2. Popular clustering algorithms
   - Core idea
   - Determining the number of clusters
   - Visualization

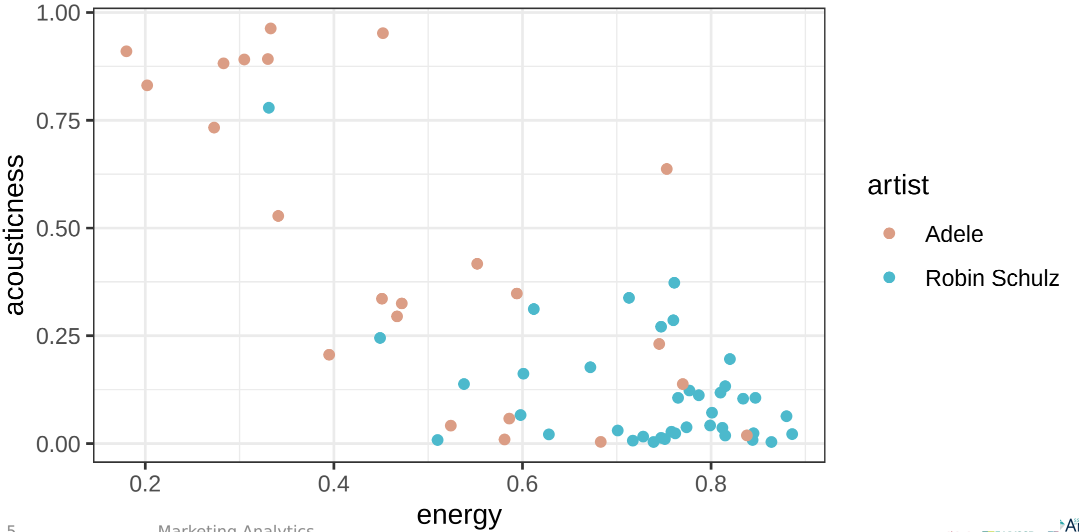
Basic concept of cluster analysis

- Goal: Group observations into **clusters** such that those in the same cluster are more “similar” than those of other clusters.
- Reduction in number of *rows*
- No distinction between dependent and independent variables.
- What exactly constitutes a cluster is not clear and many different concepts exist.
- We are going to discuss two popular concepts:
  - Centroid based: K-Means
  - Connectivity based: Hierarchical clustering

K-Means

- Assigns each observation to one of K clusters.
- Iterative procedure repeated until cluster assignments no longer change:
  1. Assign each observation to the cluster with the closest mean
  2. Re-calculate the cluster means taking into account the changed assignments
- The number of clusters K is a priori unclear (more later)

A first example



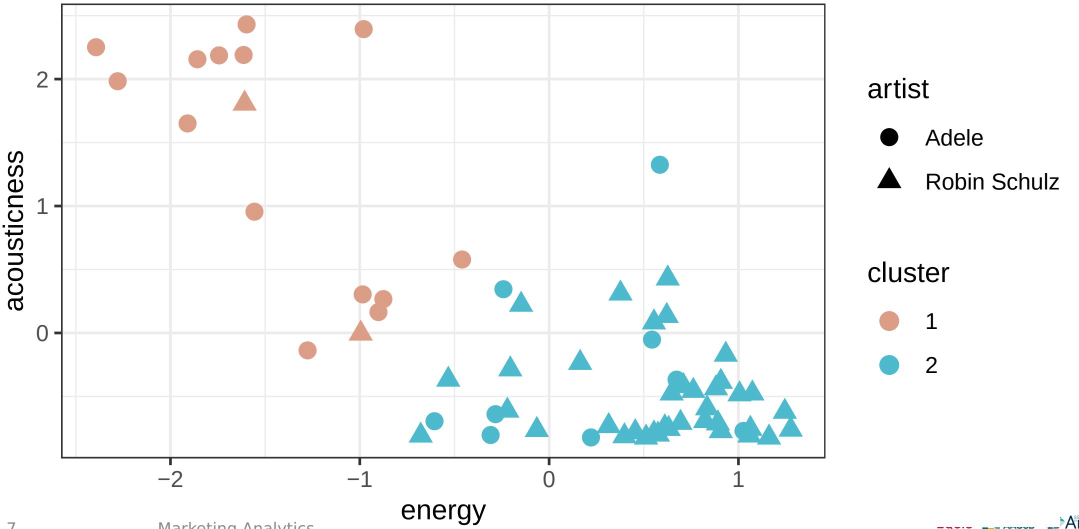
A first example

- Example: two artists and two variables, $K = 2$.
- Important: scale all variables before clustering to ensure equal contribution to distance

```
tracks_scale <- data.frame(
  artist = example_tracks$artist,
  energy = scale(example_tracks$energy),
  acousticness = scale(example_tracks$acousticness))
kmeans_clusters <- kmeans(tracks_scale[-1], 2)
kmeans_clusters$centers
```

```
     energy acousticness
1 -1.439466    1.3234653
2  0.500684   -0.4603358
```

A first example



Choosing the number of clusters

- If we extend the sample to the more interesting case of multiple artists the optimal K is unclear
- We can calculate varying indices for the optimal K and use the one that is optimal for the most indices
- In this case 3 is the best number of clusters according to the majority rule, chosen by 13 indices

```
library(NbClust)
opt_K <- NbClust(famous_tracks_scale,
                 method = "kmeans", max.nc = 10)
```

```
table(opt_K$Best.nc["Number_clusters", ])
```

```
 0  2  3  4  8 10
 2  5 13  1  1  4
```

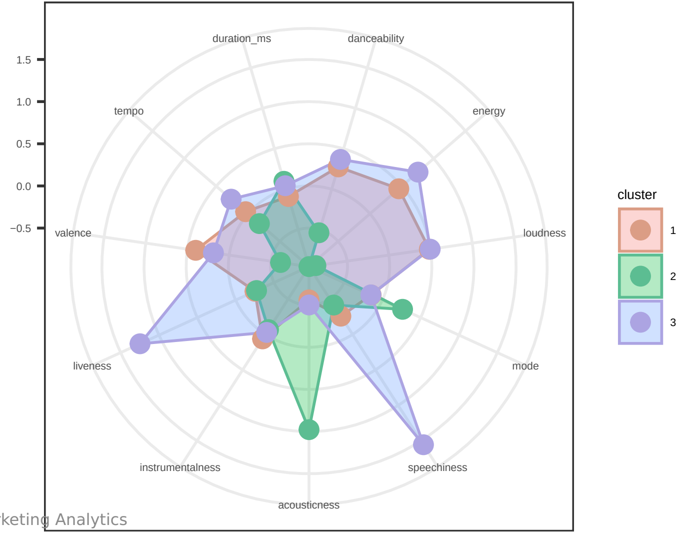
Extended example

```
kmeans_tracks <- kmeans(famous_tracks_scale, 3)
kmeans_tracks$centers
```

```
  danceability     energy   loudness       mode speechiness
1    0.2758301  0.4526214  0.4853302 -0.1461378  -0.2576401
2   -0.5385548 -0.9566495 -0.8742383  0.2632910  -0.4147683
3    0.3678342  0.7543900  0.4954862 -0.1492974   1.5546155
  acousticness instrumentalness   liveness    valence
1   -0.5618965       0.06266324 -0.2524251  0.4012080
2    0.9772843      -0.06816719 -0.2728148 -0.6158085
3   -0.5015270      -0.02849344  1.2469257  0.1885188
        tempo duration_ms
1  0.03789976 -0.09047459
2 -0.17761023  0.09754255
3  0.26482862  0.04295692
```

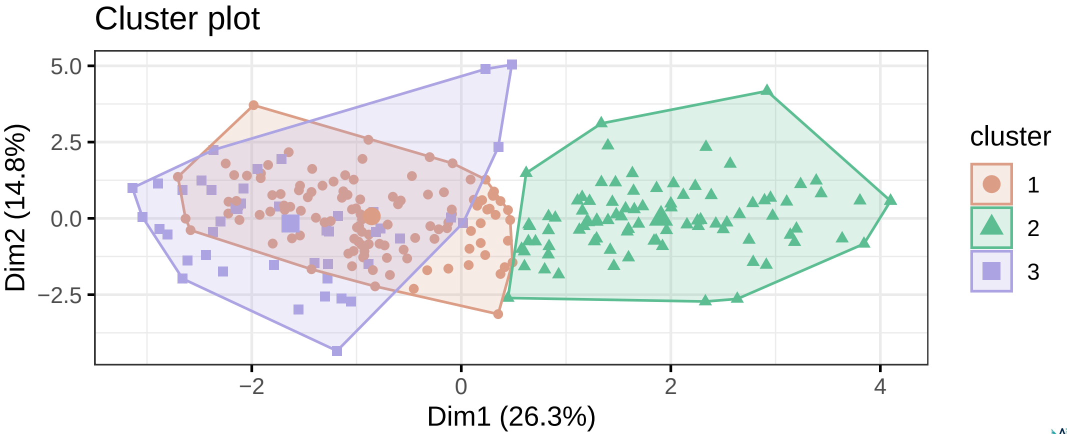
Extended example

- Characterize clusters



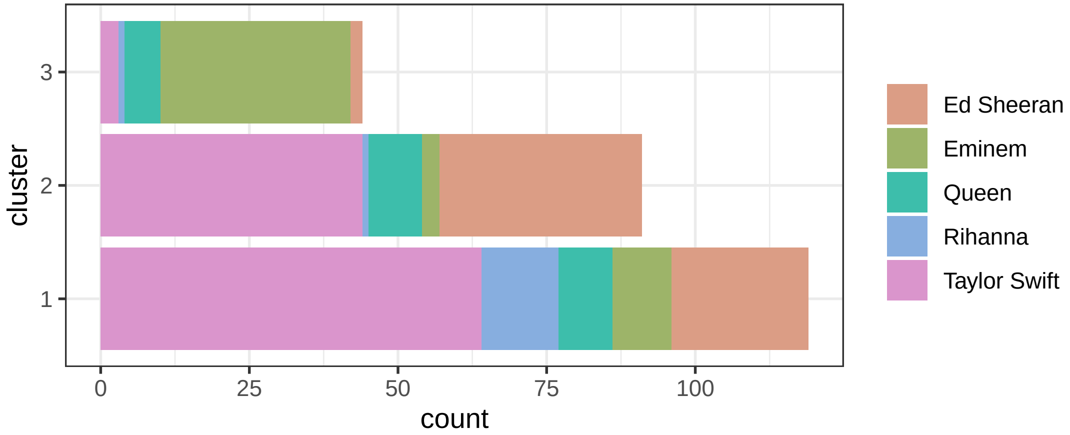
2D-Visualization

- Reduce dimensions with PCA and plot 2 components
- Gives a *partial* picture



Extended example

- Characterize artists



Making a recommendation

- Given a liked song, recommend songs in the same cluster

```
famous_tracks[famous_tracks$trackName=="The Archer","cluster"]
```

```
[1] 2
Levels: 1 2 3
```

trackName	artistName	cluster
Bohemian Rhapsody - Remastered 2011	Queen	2
Photograph	Ed Sheeran	2
The A Team	Ed Sheeran	2
I See Fire - From “The Hobbit - The Desolation Of Smaug”	Ed Sheeran	2
Give Me Love	Ed Sheeran	2
Tenerife Sea	Ed Sheeran	2

Hierarchical clustering

- Idea: similar observations can be “merged” to one cluster
- Starts out merging pairs of the most similar or closest observations
- Iteratively merges the most similar clusters until there is only one cluster left
- Does not require a priori setting of the number of clusters
- Number of clusters is determined post-hoc by “cutting-off” at some iteration

Visualization

Example

- Calculate distances between observations (default: Euclidean) using `dist`
- Use distances in `hclust` to perform a hierarchical cluster analysis

```
hclust_tracks <- hclust(dist(pf_ri_scale))
hclust_tracks
```

```
Call:
hclust(d = dist(pf_ri_scale))

Cluster method   : complete
Distance         : euclidean
Number of objects: 34
```

Characterize clusters

- Decide on cut-off based on dendrogram. Specify the desired number of clusters using `cutree`.
- Calculate summary statistics for cluster

```
hclusters <- cutree(hclust_tracks,4)
pf_ri_hier <- data.frame(pf_ri_scale)
pf_ri_hier$cluster <- as.factor(hclusters)
aggregate(. ~ cluster, pf_ri_hier, mean)[,1:5]
```

```
  cluster danceability     energy   loudness       mode
1       1    0.6301558 -0.8423205  0.4931387 -1.1087884
2       2    0.6793570  0.7636001  0.6540518 -0.2407238
3       3   -0.6149044 -0.7835601 -0.7125465  0.4785297
4       4   -1.1381509  0.1890603 -0.1348845 -1.1087884
```