Lab 4 - Machine Learning for Social Science

To be handed in no later than September 24th, 13:00. The submission should include code, relevant output, as well as answers to questions. We recommend the use of RMarkdown to create the report.

Part 1: Taste clustering and influence

In the first part of this lab, we will consider a (simulated) data set which contains information about a sample of (fictive) individuals' music tastes as well as a measure of their influence on others.

1. Begin by importing the file "`taste_influence.csv`". Report the number of rows and columns of the data set, and the genres contained in it. Create a scatter-plot of two combinations of genres of your choice. Based on this, do you get any indication that the data is clustered along musical tastes?

2. Now you shall do some clustering. To prepare the data, do the following: (*i*) store/copy the data to a new `R` object, and subset it so that it only contains the three "taste columns"— these are the columns you will cluster based upon, (*ii*) standardize this data table (hint: you can e.g., use `scale()` for this purpose), (*iii*) transform it into a matrix (hint: e.g., by using `as.matrix()`).

3. Having formatted the data according to #2, you shall now use the *kmeans* algorithm to cluster your data. Recall that a requisite for running *kmeans* is that the parameter k has been specified. In practice—and as is the case here—we often do not know the appropriate number of clusters a priori. Therefore, you shall implement a loop that, at every iteration, runs kmeans with a different number of clusters, and extracts the *total within cluster sum of squares* (hint 1: which can be extracted using `$tot.withinss` | hint 2: set the argument `nstart=100` to ensure robustness of the local optima you find). Consider no. clusters ranging from 1 to 20, with an interval of 1. Plot k against `tot.withinss`. Which number of clusters do you find appropriate? Motivate.

4. For the specification (of k) that you decided on in #3, extract the *centroids* and interpret each cluster in terms of what distinguishes it from the rest. Do the clusters seem meaningfully distinct?

5. To get a feeling for the role that the choice of k plays, estimate another *kmeans* model but this time with $k = 2$. Inspecting the centroids, how does your clustering change; how does it alter your understanding of the population?

6. Clustering provides a tool for discovering underlying structures in our data. Once these structures have been discovered, they can be studied in separate analyses. That is what you shall do now. We want to examine whether different "taste types" have differential degree of influence on others. To do so, (*i*) create a new column in your original data set storing the the retrieved cluster assignments (hint: you find the cluster assignments using `$cluster`). Then (*ii*) estimate a linear regression with the *influence score* (`infuence`) as the outcome variable, and the clusters (formatted as a factor) as predictors. Interpret the results: are there any difference in influence between the clusters?

7. Now that you have merged the cluster assignments to the original data, produce the same plots as you did in #1, but now colored by the cluster assignments. Does it look like *kmeans* have picked up on the patterns you observed in #1? Further—what you think of the separation between the clusters? Is there clear spacing betweeen the clusters, or are the borders almost touching each other (note that there will be certain overlap due to plotting the data in 2D)?

8. Repeat step 3–7 (but skip #5) using now instead a *Gaussian mixture model*. For this, you may use `mclust`'s function `Mclust()` (specifying the number of components with the argument `G`). For #3: Note that, because this is a probabilistic model, we retrieve a *likelihood score* (or, more specifically *BIC* which is based upon the likehliood score but also penalizes for complexity) to measure its performance instead of *total within cluster sum of squares* (hint: you can extract the BIC by `$bic` on the model object). For #4, you can use `$parameters$mean` to extract the means/centroids of each cluster. For #6, you shall extract the hard cluster assignments (which you can do using `$classification`).[1]

9. Something which `Mclust()` also provides is a score for each observation how *uncertain* we are about its assignment. As mentioned during the lecture, "border-observations" can sometimes be substantively meaningful to study. You shall do so here. Extract the vector `$uncertainity` from the Gaussian mixture model fit, and store it in the original data. Then yet again fit a linear regression (together with the taste variables), but this time additionally with the `uncertainity` variable.

Part 2: Regional variation

In the second part of the lab, we will consider another simulated data set. This time, containing information about both the (fictive) individuals themselves, but also their social environments.

1. Begin by importing the file "`neighborhood.csv`". Report the number of rows and columns of the data set, and make a brief note on the types of columns contained in it.

2. Based on the types of variables we find, we have some suspicion that there may exist considerable correlation between different variables in this data set. To explore whether we can capture key aspects of our data using fewer dimensions, we will use *PCA* and its extensions. Begin by estimating a *principal components* model *without* doing any standardization (hint: to estimate a PCA, use `prcomp()`). Why is this problematic (hint: examine the principal loadings)

3. Now, standardize your data, and then fit a PCA on this standardized data set. Plot the *proportion variance explained*. Interpret and decide on an appropriate number of principal components.

4. Interpret the retrieved principal components based on their loadings. Do they provide easy and substantively expected interpretations?

5. Because of the conclusions in #4, we will now consider the *sparse PCA*. Use the same number of principal components that you did for the standard PCA in #2. In comparison to the standard PCA, we have an additional parameter λ in the sparse PCA. Us the *IS* index to determine an appropriate λ. Inspect the principal loadings for the resulting configuration. Interpret each dimension. Which do you think was easier to interpret; the sparse PCA or the standard PCA? Are there any downsides to sparse PCA?

6. As a last exercise for today, you shall simulate your own data. Generate a dataset of 50 observations and 50 independent variables using the function provided below:

```
gen_data <- function(n,p){
  df <- c()
  for(i in 1:p){
    ith_var <- rnorm(n = n, mean = 0, sd = 1)
    df <- cbind(df,ith_var)
  }
```

[1] Because soft assignments sum up to exactly 1 for every individual, we cannot include all soft assignment values; it creates multi-colinearity issues (just like we only can estimate p-1 dummies for a p-level categorical variable). Interpretting the remaining p-1 coefficients comes with some caveats (that are not important for the course). I recommend reading the following stackoverflow thread if you want to learn more: https://stats.stackexchange.com/questions/183601/interpreting-proportions-thatsum-to-one-as-independent-variables-in-linear-regr. For these reasons, we here instead work with the hard assignments.

```
  return(df)
}
```

- Once you have generated the data, process your data as you did above for the `neighborhood` data set (standardize, making into a matrix). Then, estimate a standard PCA. What do you find: could the PCA help us effecively redue the dimensionality of our data or not? Why?

Quiz wrap-up:

Provided here are four optional wrap-up questions that help recap some central ideas from the lecture, and which are taken from old exams.

1. Which of the following are true about the relation between *supervised* learning and *unsupervised* learning (1p):

    a. The task of assigning observations to *predefined* categories *is* exclusive to supervised learning.
    b. Discovery of previously unknown patterns/relations is exclusive to unsupervised learning.
    c. Supervised learning problems have a ground-truth. Unsupervised learning problems do not.
    d. The problem of models picking up noise and spurious patterns in data is exclusive to supervised learning.

2. We can use PCA on a matrix X in order to:

    a. Discover latent organizations of the variables in X.
    b. Partition observations into distinct clusters.
    c. Predict some outcome variable Y.
    d. Reduce the dimensionality of X.

3. When using a quantitative approach to select the number of clusters (or dimensions) in unsupervised learning, which two competing forces do we usually seek to balance?

4. For which of the following scenarios do you have a good reason to make a choice—about the number of clusters/dimensions—that contradicts the decision based on the so-called elbow criterion:

    a. You have no domain knowledge and no hypothesis; you are just interested in exploring the data.
    b. You have substantial domain knowledge and a clear hypothesis.
    c. You do not care about interpretability. Your goal is to use the principal scores in an supervised learning model to predict some outcome as well as possible.
    d. Your purpose for using PCA is to visualize your data.