Modern Statistics and Big Data Analysis, Exercises 7

Deadline: End of Friday 1 December.

1. (3.5 points) The `flexmixedruns` function used to fit the latent class categorical mixture on the course slides is in package fpc. There is another R-function to fit that model, namely `poLCA` in package poLCA. I'd have expected the `poLCA` function to be as good as `flexmixedruns` or better (because I wrote the `flexmixedruns` function myself and did it pretty quickly), however the `poLCA` function couldn't handle the Veronica example properly, probably because there are too many variables.

   The package poLCA has an example data set for latent class clustering with categorical variables that are not just binary called "election", to be loaded by `data(election)` (requiring `library(poLCA)`); also look up the election help page for some documentation about these data.

   The categorical variables to be clustered are variables 1-12 (to be extracted before clustering; this can be done by `election12 <- election[,1:12]`).

   The data set has missing values. If you just provide the data set to the `poLCA`-function as done in the examples on the `election`-help page, all observations with missing value (474 out of 1785) will be omitted. There is however a more elegant way of dealing with missing value if there are categorical variables. You can define a new category for the missing values, i.e., replacing all missing values NA by a category called "NA". Note that all the variables are of type "factor", and this requires to define a new factor level:

```
electionwithna <- election12
for (i in 1:12){
  levels(electionwithna[,i]) <- c(levels(election12[,i]),"NA")
  electionwithna[is.na(election12[,i]),i] <- "NA"
}
```

   This is the data set you are asked to work with in parts (a)-(c), and (e) (see part (d) for what to use there). Run the following clusterings and compare them using MDS plots based on the simple matching distance. Comment on which clusterings seem more or less convincing.

   (a) Compute a latent class clustering with 3 clusters using `poLCA` (read its help page and decide about potentially useful parameter settings).

   (b) Compute a latent class clustering with 3 clusters using `flexmixedruns`.

   (c) Compute a distance-based clustering of your choice with 3 clusters based on the simple matching distance (you can also choose here between computing the simple matching distance on `electionwithna` or on `election12`, which will compute the distance just taking variables into account that are non-missing on both observations, using `daisy` as explained earlier in class - actually in general then it's a dissimilarity and not a distance as missing values can spoil the triangle inequality).

(d) Compute a latent class clustering using `flexmixedruns` with estimated number of clusters. (`poLCA` will not estimate the number of cluster automatically, although it gives out the BIC, so this could in principle be implemented easily.)

(e) The original `election` data set also has the variables AGE and EDUC (these are the variables number 14 and 15). Define a data set `election14` that has the 12 variables already used above and AGE and EDUC. Assuming that these can be treated as continuous variables (which is rather questionable at least for EDUC, but you can ignore this here), use `flexmixedruns` to compute a clustering based on a latent class mixture model, including estimation of the number of clusters, were the continuous variables are assumed Gaussian within clusters independently of the categorical variables. In order to use all observations for this, impute the missing values of AGE and EDUC with the mean of these variables. You can use the MDS already computed above for visualising this, but if you are curious, you can also run a new MDS for these data using the Gower coefficient.

2. (1.5 points) For one of the latent class clusterings computed in question 1 on the `electionwithna`-data produce a heatmap as on slide 289 of the course notes. Note that this requires a different colour choice for the heatmap "entries" than in the course notes, as the data now have more than 2 categories (it is part of the exercise to find out how to do this). Comment on the plots. Do you find the clusters convincing? Why or why not? Is there evidence against local independence?

3. (1 points) Assume a situation with 10 categorical variables. Five variables are binary, three variables have three categories, and two variables have five categories. What is the number of free parameters for

   (a) a general categorical model that models all possible probabilities,

   (b) a latent class mixture model with 4 mixture components?

   Remark: Note that the value computed in (a) plus one is the number of possible different observations! If this number is low, it also implies a strong restriction for the possible numbers of clusters to be fitted, which cannot be larger and should normally be substantially smaller, regardless of the number of observations.

4. (2 points) Consider the COVID data set analysed in Chapter 7 of the course slides. Consider Italy, Haiti, and the USA (country 79, 69, and 164). Produce residual plots, i.e., plots with the time points on the x-axis and the residuals (difference between data and fit) on the y-axis) for these countries for

   (a) the fit by a B-spline basis with $p = 100$,

   (b) the fit by a 5-dimensional principal components basis as shown on p. 310 of the slides.

   Comment on potential model assumption violations from these plots. (It is part of the exercise to figure out how to produce these plots.)

5. (2 points) Representing all countries in the COVID data set by the first functional principal component scores only, run a one-way analysis of variance to test whether there is evidence that the scores from different continents have different means. Also visualise the scores so that the continents can easily be compared, and interpret plot and result (try to figure out what larger or smaller/positive or negative scores on the first principal component actually mean). This is a simply method to run a test comparing groups of functional data objects (obviously relying on the information represented by the first principal component only).