How to perform mixture multigroup factor analysis with 'MixtureMG_FA' in R

1) Download the R-codes from https://github.com/KimDeRoover/MixtureMG_FA

2) Set your working directory and make sure the data and MixtureMG_FA R-codes are in that directory. Source the MixtureMG_FA, MixtureMG_FA_loadings, MixtureMG_FA_intercepts, and MixtureMG_FA_loadingsandintercepts codes.

3) Load data, for example:

```
normdata<-read.csv("emotienormendata_exclcontEgypt.csv",header=TRUE)
```

Data should contain numerical group ID in the first columns, and no more variables than the ones you want to include in the MixtureMG_FA. Missing data can currently not be dealt with and should be removed or imputed.

4) **If you want to use exploratory factor analysis (EFA)**, specify no design matrix (use 'design = 0' in what follows) and install the GPArotation package (if you want rotated factors):
install.packages("GPArotation")

If you want to use confirmatory factor analysis (CFA), you should specify a 'design' matrix containing zeros for the positions of the zero-loading restrictions and ones for the remaining entries; for example:

```
nfactors=2
IM=diag(nfactors)
design=c(2,2,1,1,2,1,1,1,2,1,1,2)
design=IM[design,]
```

5) Create room for plots in your plots pane (in Rstudio) – to avoid an error – and specify the inputs to the MixtureMG_FA function as follows:

For example, to perform mixture multigroup factor analyses with two factors ('nfactors=2') and 1 to 6 clusters ('nsclust=c(1,6)'), where the clusters of groups are based on equivalence of loadings ('cluster.spec="loadings" ') and EFA is used ('design=0') with oblimin rotation for each cluster ('rotation="oblimin" '):

```
OutputObject <-
MixtureMG_FA(normdata,cluster.spec="loadings",nsclust=c(1,6),nfactors=2,Maxiter = 5000,nruns
= 50,design=0,rotation="oblimin",preselect=10)
```

For example, to perform mixture multigroup factor analyses with two factors ('nfactors=2') and 1 to 6 clusters ('nsclust=c(1,6)'), where the clusters of groups are based on equivalence of intercepts ('cluster.spec="intercepts" ') and CFA is used ('design=design'):

```
OutputObject <-
MixtureMG_FA(normdata,cluster.spec="intercepts",nsclust=c(1,6),nfactors=2,Maxiter =
5000,nruns = 10,design=design,preselect=10)
```

The available options for 'cluster.spec' are currently:

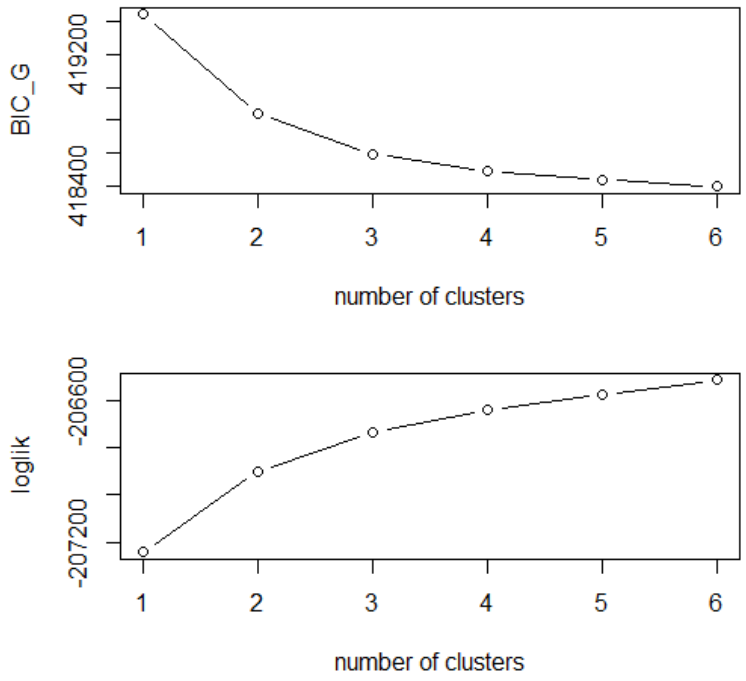
- cluster.spec="loadings"
- cluster.spec="intercepts"
- cluster.spec=c("loadings", "intercepts")

6) Inspect the overview output and the plots to select the best number of clusters for your data set:

```
> OutputObject<-MixtureMG_FA(normdata,cluster.spec="loadings",nsclust=c(1,6),nfactors,Maxiter =
5000,nruns = 25,design=0,rotation="oblimin",preselect=50)
[1] Fitting MMG-FA with 1 cluster ...
[1] Fitting MMG-FA with 2 clusters ...
[1] Fitting MMG-FA with 3 clusters ...
[1] Fitting MMG-FA with 4 clusters ...
[1] Fitting MMG-FA with 5 clusters ...
[1] Fitting MMG-FA with 6 clusters ...

  nr of clusters    loglik nrpars    BIC_G screeratios convergence nr.activated.constraints
1              1 -207241.6   1289 419446.0    0.000000           1                       0
2              2 -206899.9   1310 418843.4    2.060709           1                       0
3              3 -206734.0   1331 418592.6    1.814598           1                       0
4              4 -206642.7   1352 418490.7    1.376817           1                       0
5              5 -206576.3   1373 418438.9    1.082555           1                       0
6              6 -206515.0   1394 418397.1    0.000000           1                       0

  >> Choose the best number of clusters ('K_best') based on the plots and access the corresponding
 cluster memberships and parameter estimates by using, for example, OutputObject$clustermembership
s[[K_best]] and OutputObject$clusterspecific.loadings[[K_best]].
  >> The parameter sets are further subdivided in group- and/or cluster-specific parameter sets.
```



In the 'BIC_G' plot (i.e., the upper plot), you can see for which number of clusters the BIC_G is minimal (or, at least, an elbow in the decrease is visible, since BIC_G tends to keep decreasing with additional clusters for many empirical datasets; see De Roover, Vermunt, & Ceulemans, 2020).

In the 'screeratios' column of the overview, you can see for which number of clusters the scree ratio is maximal (indicating that the scree plot, i.e., the lower plot, levels off). Be aware of potential artificially inflated scree ratio's when the increase in fit with additional clusters really approaches zero, which causes the scree ratio to become very large even when no elbow point (but rather a straight line) is visible. Such a solution should NOT be selected. Therefore, look at both the scree ratio's AND the plot. Looking at the plot also helps to decide whether you would want to inspect the 'second best' solution (with the second highest scree ratio).

(Note that if no local maxima are found (and, with regard to the plot, if the number of activated constraints are zero), the scree ratio's and scree plot are identical to the CHull method. You can choose to use the multichull package for the CHull method, in which case you should lower the value for the PercentageFit parameter, related to the artificial inflation of scree ratio's mentioned above, to 0.001 or lower).

7) Access the parameter values corresponding to the selected number of clusters in, for example, OutputObject$MMGFA_solutions$"2.clusters" or OutputObject$MMGFA_solutions[[2]]:

For example, the cluster memberships are accessed as follows:

```
round(MSoutput$MMGFA_solutions$"2.clusters"$clustermemberships,6)
```

```
      Cluster_1 Cluster_2
 [1,] 0.000000 1.000000
 [2,] 0.000000 1.000000
 [3,] 1.000000 0.000000
 [4,] 0.999986 0.000014
 [5,] 0.000000 1.000000
 [6,] 1.000000 0.000000
 [7,] 1.000000 0.000000
 [8,] 0.000000 1.000000
 [9,] 0.000000 1.000000
[10,] 0.000676 0.999324
[11,] 0.000000 1.000000
[12,] 0.000000 1.000000
[13,] 0.000000 1.000000
[14,] 0.000000 1.000000
[15,] 0.000000 1.000000
[16,] 0.000000 1.000000
[17,] 0.000000 1.000000
[18,] 1.000000 0.000000
[19,] 0.002719 0.997281
[20,] 0.989188 0.010812
[21,] 1.000000 0.000000
[22,] 0.000000 1.000000
[23,] 1.000000 0.000000
[24,] 0.986320 0.013680
[25,] 0.000000 1.000000
[26,] 0.000000 1.000000
[27,] 1.000000 0.000000
[28,] 0.000001 0.999999
[29,] 0.999768 0.000232
[30,] 0.000000 1.000000
[31,] 0.000753 0.999247
[32,] 0.998844 0.001156
[33,] 0.000000 1.000000
[34,] 0.000692 0.999308
[35,] 0.113884 0.886116
[36,] 0.000189 0.999811
[37,] 0.000000 1.000000
```

```
[38,] 0.999997 0.000003
[39,] 0.999367 0.000633
[40,] 0.999934 0.000066
[41,] 1.000000 0.000000
[42,] 0.000000 1.000000
[43,] 0.000000 1.000000
[44,] 1.000000 0.000000
[45,] 0.000002 0.999998
[46,] 0.000021 0.999979
[47,] 0.000000 1.000000
```

And the oblimin rotated loadings of the first cluster as follows:

```
OutputObject$MMGFA_solutions$"2.clusters"$clusterspecific.loadings[[1]]
          Cluster_1       Cluster_2
 [1,]  0.06005637 -1.282187845
 [2,] -0.04357160 -1.218860479
 [3,] -1.31267720 -0.180582904
 [4,] -1.30796200  0.001253857
 [5,] -0.02180187 -1.145670828
 [6,] -1.48229120 -0.168922786
 [7,] -1.73177014  0.196774877
 [8,] -1.54428115  0.111805856
 [9,] -0.04740498 -0.962115516
[10,] -1.27101356 -0.142071133
[11,] -1.20299713 -0.081355405
[12,]  0.06862353 -1.035055041
```

And the counter-rotated factor (co)variance matrix of group 2 in cluster 1 as follows:

```
OutputObject$MMGFA_solutions$"2.clusters"$group.and.clusterspecific.factorcovariances[[2,1]]
          [,1]          [,2]
[1,] 1.0714420 0.3521687
[2,] 0.3521687 0.7032381
```

Or, similarly, in case of clusters based on the intercepts, access the cluster-specific intercepts as follows:

```
> OutputObject$MMGFA_solutions$`2.clusters`$clusterspecific.intercepts
                [,1]        [,2]        [,3]       [,4]        [,5]        [,6]        [,7]        [,8]        [,9]
Cluster_1  7.372984 7.333838 4.219937 4.62205 7.187141 5.184071 4.635324 4.206072 6.636692
Cluster_2  7.152304 7.151708 4.611682 4.70016 6.835172 4.942833 4.912570 4.309714 5.528593
               [,10]       [,11]       [,12]
Cluster_1  4.148917 4.047005 6.595535
Cluster_2  4.548368 4.568226 6.783675
```

Further recommendations and suggestions:

- If nfactors=1, there is no difference between EFA and CFA.

- If the overview shows **lack of convergence** (i.e., zeros in the 'convergence' column), increase the value for 'Maxiter'.

- If the scree plot looks a bit erratic, it could be due to **local maxima** and you should increase the value of 'nruns'.

- If your data is very large (e.g., very large number of groups and/or observations per groups), the **analysis may become slow for the higher number of clusters**. You can speed up the analysis by bringing the 'preselect' number closer to 100 and 'nruns' to 10 or 25. I will perform more speed-ups in the near future. In the meantime, when waiting for your results, you can comfort yourself with the thought that it would take up to 300 times longer in Mplus (*if* specifying a mixture multigroup factor analysis is possible with the syntax options currently available).

- Look at the papers listed below for more recommendations on sample sizes, number of starts, model selection and how to move on from the results of these analyses.

Thank you for using Mixture Multigroup Factor Analysis and citing the papers below!

Kim De Roover

References

De Roover, K. (2021). Finding clusters of groups with measurement invariance: Unraveling intercept non-invariance with mixture multigroup factor analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 1-21. Advance online publication. https://doi.org/10.1080/10705511.2020.1866577

De Roover, K., Vermunt, J. K., & Ceulemans, E. (2020). Mixture multigroup factor analysis for unraveling factor loading non-invariance across many groups. *Psychological Methods*. Advance online publication. https://doi.org/10.1037/met0000355