Bayesian nonparametric methods for clustering: Practicals 2

Anaïs Rouanet & Boris Hejblum

07/04/2021

Package installation

Install the following packages and load them:

```
install.packages("PReMiuM", "lcmm", "NormPsy", "coda", "ggplot2")
```

```
library("PReMiuM")
library("lcmm")
library("NormPsy")
library("coda")
library("ggplot2")

source("functions_to_load.R")
```

In your working directory, create 5 folders with names: nMMSE, alpha, pred, multiple_chains, labelswitching.

Paquid Dataset

In this practical, we will analyse a dataset from the French Paquid propective study (Letenneur et al., 1994) that aimed study normal and pathological brain ageing. Load the Paquid dataset, from the lcmm package, that contains 500 participants with the following variables:

- ID: participant identifier
- MMSE: Mini Mental State Examination - psychometric test [0-30]
- BVRT: Benton Visual Retention Test - psychometric test [0-15]
- IST: Isaac's Set Test - psychometric test [0-40]
- HIER: physical dependency [0, 1, 2, 3]
- CESD: depression sympatomatology [0-52]
- age: age at each observation
- agedem: age at dementia diagnosis
- dem: denmentia diagnosis status
- age_init: age at baseline
- CEP: 1 if primary school diploma obtained, 0 otherwise
- male: 1 if male 0 otherwise

```
data(paquid)
head(paquid)
```

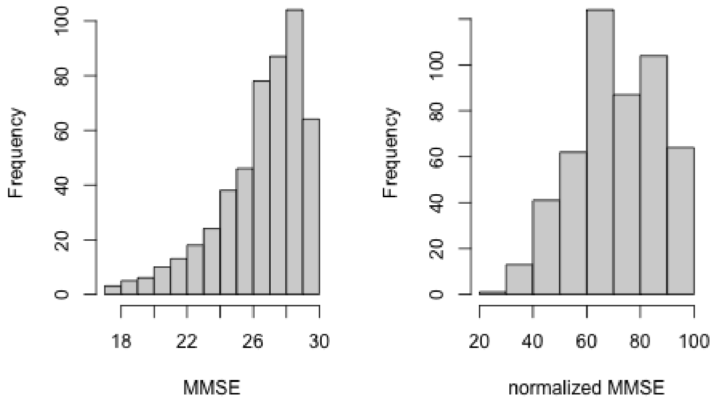
```
##   ID MMSE BVRT IST HIER CESD      age  agedem dem age_init CEP male
## 1  1   26   10  37    2   11 68.50630 68.5063   0  67.4167   1    1
## 2  2   26   13  25    1   10 66.99540 85.6167   1  65.9167   1    0
## 3  2   28   13  28    1   15 69.09530 85.6167   1  65.9167   1    0
## 4  2   25   12  23    1   18 73.80720 85.6167   1  65.9167   1    0
## 5  2   24   13  16    3   22 84.14237 85.6167   1  65.9167   1    0
## 6  2   22    9  15    3   NA 87.09103 85.6167   1  65.9167   1    0
```

We will focus on baseline observations only. Create a dataset with baseline observations and characteristics.

```
data <- paquid[order(paquid$ID, paquid$age),]
baseline <- data[sapply(unique(paquid$ID), function(x) which(paquid$ID==x)[1]),]
```

Let's have a look at the MMSE distribution at baseline. We will use the normalising function for the cognitive test MMSE, proposed by Philipps et al. (2014), to obtain a Gaussian outcome [0-100].

```
baseline$nMMSE <- sapply(baseline$MMSE, normMMSE)
par(mfrow=c(1,2))
hist(baseline$MMSE, xlab = "MMSE", main ="")
hist(baseline$nMMSE, xlab = "normalized MMSE", main ="")
```



Now let's have a look at the covariates: age at baseline (age_init), education (CEP) and sex (male).

```
summary(baseline$age_init)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   65.25   68.42   73.83   74.23   78.42   92.33
```

```
as.data.frame.matrix(table(baseline$CEP, baseline$male),
                     row.names = c("CEP-", "CEP+"))
```

```
##        0   1
## CEP- 101  44
## CEP+ 187 168
```

Outcome-guided analysis

Missing values for the outcome are not handled. Remove the observations where nMMSE is missing.

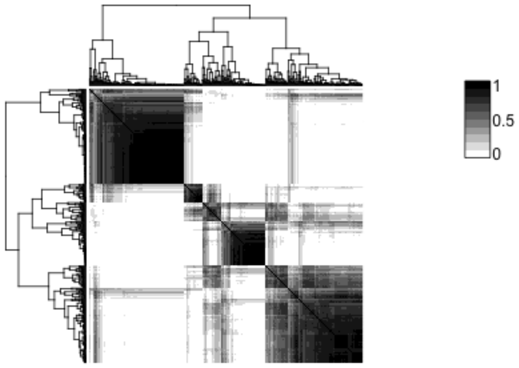
```
baseline <- baseline[which(!is.na(baseline$nMMSE)),]
(N1= length(unique(baseline$ID)))
```

```
## [1] 496
```

Specify the profile regression model to run a semi-supervised clustering analysis considering the normalized MMSE as outcome and age at baseline, education and sex as profile covariates. Once the model is estimated:

- plot the posterior similarity matrix
- identify the best partition
- plot the cluster-specific risks (outcome patterns) and covariate profiles.

```
mod_nMMSE_1<-profRegr(yModel="Normal",   # model type for the outcome
                      xModel="Mixed",    # model type for the covariates
                      nSweeps=10000,     # Number of sweeps
                      nClusInit=20,      # Initial number of clusters
                      nBurn=2000,        # Number of burn in iterations
                      data=baseline,     # database
                      output="nMMSE/output",   # Path and name of the output files
                      covNames = c("age_init", "CEP", "male"), # Profile covariate names
                      discreteCovs  = c("CEP", "male"), # Names of discrete profile covariates
                      continuousCovs = c("age_init"),    # Names of continuous profile covariates
                      outcome = "nMMSE",# Outcome name
                      useHyperpriorR1=FALSE, # No extra hyperparameters for normal profile variables
                      seed=554248199)

# Create dissimilarity matrix
dissimObj <- calcDissimilarityMatrix(mod_nMMSE_1)
# Plot the posterior similarity matrix
PSM1 <- myheatDissMat(dissimObj)
```



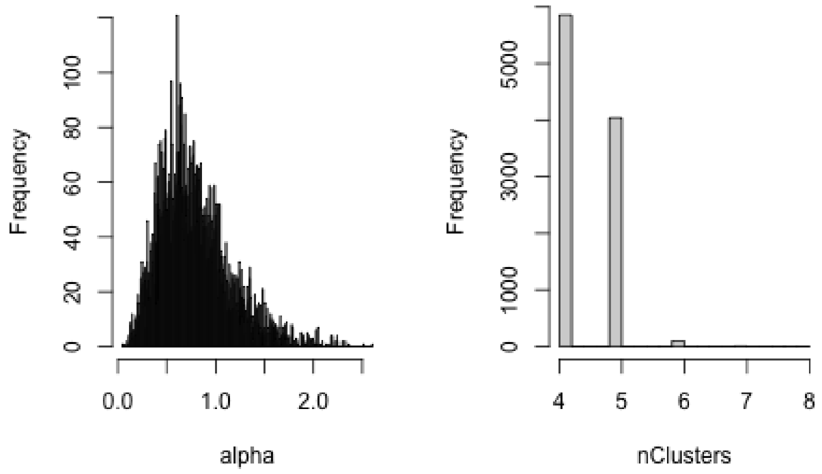
```
# Compute final partition
clusObj1 <- calcOptimalClustering(dissimObj)
# Compute risk and covariate profiles
riskProfileObj <- calcAvgRiskAndProfile(clusObj1)
#plot risk and covariate profiles
clusterOrderObj <- plotRiskProfile(riskProfileObj, "nMMSE/Cluster_profiles.png")
```

In the output folder, you will find a list of text files, including nMMSE1_log.txt that records the model specification and MCMC steps.

Posterior distributions

Plot the posterior distribution of the concentration parameter and the number of clusters:

```
alphaChain <- mcmc(read.table("./nMMSE/output_alpha.txt")[, 1])
nClus <- plot_trace(mod_nMMSE_1, "nClusters", plot=FALSE)

par(mfrow=c(1,2))
hist(alphaChain, breaks=500,  xlab="alpha", main="")
hist(nClus, main="", xlab="nClusters")
```



```
getmode(alphaChain)*log(1+500/getmode(alphaChain))
```

```
## [1] 4.092896
```

We can define pseudo covariate profiles to obtain predictions of the outcome:

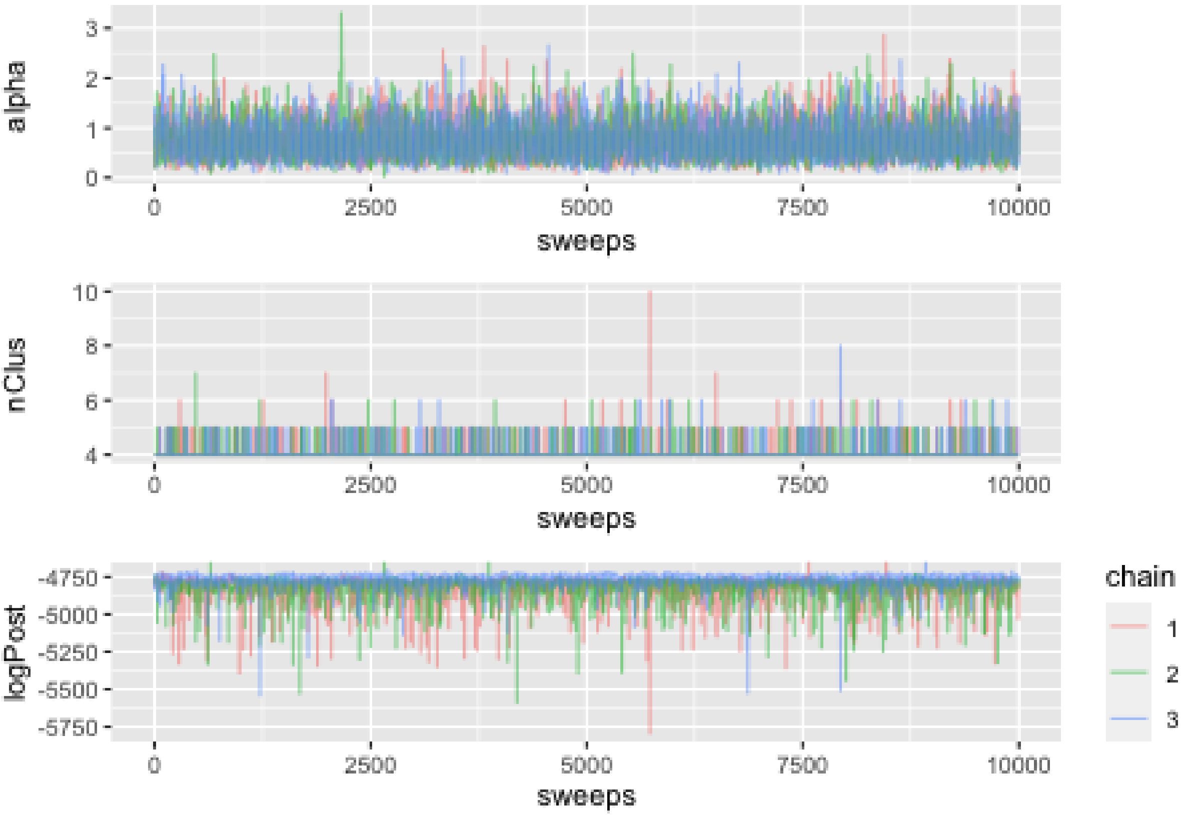
```
preds <- data.frame(matrix(c(75, 0, 0, 75, 0, 1), ncol = 3,  byrow = TRUE))
colnames(preds) <- c("age_init", "CEP", "male")

mod_nMMSE_1p<-profRegr(yModel="Normal",
                     xModel="Mixed",
                     nSweeps=10000,
                     nClusInit=20,
                     nBurn=2000,
                     data=baseline,
                     output="pred/output",
                     covNames = c("age_init", "CEP", "male"),
                     discreteCovs  = c("CEP", "male"),
                     continuousCovs = c("age_init"),
                     outcome = "nMMSE", useHyperpriorR1=FALSE,
                     seed = 554248199,
                     predict = preds)

dissimObj <- calcDissimilarityMatrix(mod_nMMSE_1p)
clusObj <- calcOptimalClustering(dissimObj)
riskProfileObj <- calcAvgRiskAndProfile(clusObj)
predictions <- calcPredictions(riskProfileObj,
                               fullSweepPredictions = TRUE, fullSweepLogOR = TRUE)
```

```
plotPredictions(outfile = "./pred/predictiveDensity.pdf",
           runInfoObj = mod_nMMSE_1p, predictions = predictions, logOR = TRUE)
```

Multiple chains

Run different chains and compare the posterior similarity matrices or traces across chains.

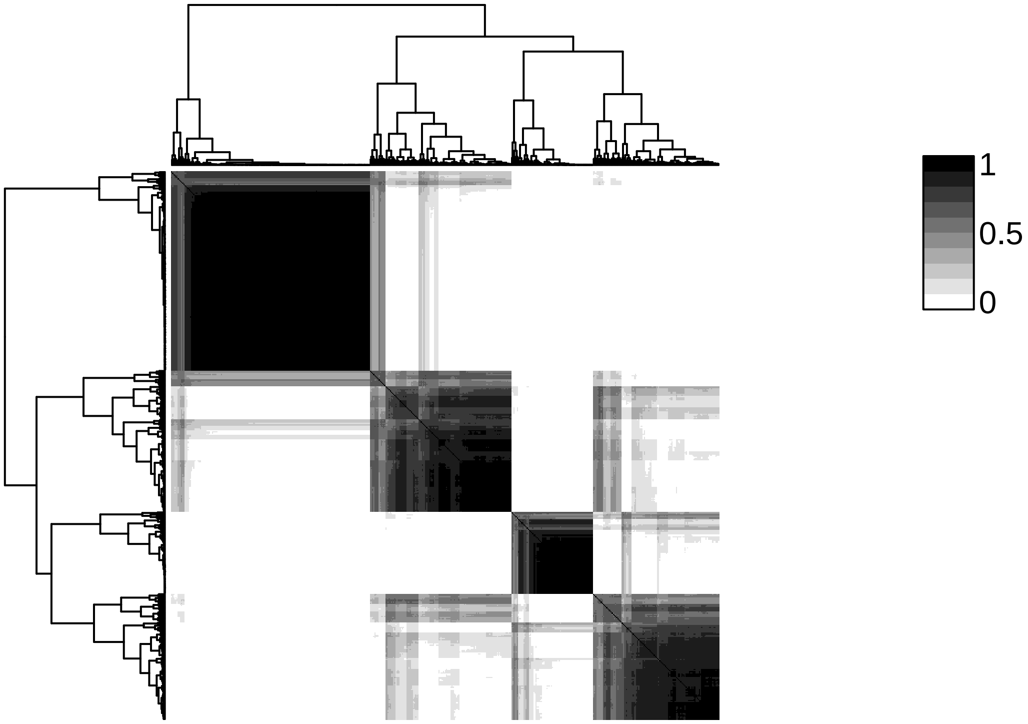
```
seeds <- c(3953863617,9934436348,2665894220)

PReMiuM_fun <- function(seed){
  profRegr(yModel="Normal",
                     xModel="Mixed",
                     nSweeps=10000,
                     nClusInit=20,
                     nBurn=2000,
                     data=baseline,
                     output=paste("multiple_chains/output_",seed,sep=""),
                     covNames = c("age_init", "CEP", "male"),
                     discreteCovs  = c("CEP", "male"),
                     continuousCovs = c("age_init"),
                     outcome = "nMMSE",
                     useHyperpriorR1=FALSE,
                     seed=seed)
  }

mod_list<-parallel::mclapply(seeds, PReMiuM_fun)

for(i in 1:length(seeds)){
  png(paste("./multiple_chains/PSM_chain_",seeds[i],sep=""))
  myheatDissMat(calcDissimilarityMatrix(mod_list[[i]]), order = PSM1$rowInd)
  dev.off()
}

nSweeps <- mod_nMMSE_1$nSweeps
alpha <- matrix(0, nSweeps,length(seeds))
nClus <- matrix(0, nSweeps,length(seeds))
logPost <- matrix(0, nSweeps,length(seeds))

for(i in 1:length(seeds)){
  alpha[,i] <- plot_trace(mod_list[[i]], "alpha", plot=FALSE)
  nClus[,i] <- plot_trace(mod_list[[i]], "nClusters", plot=FALSE)
  logPost[,i] <- plot_trace(mod_list[[i]], "logPost", plot=FALSE)
}

datag <- data.frame("alpha"=c(alpha), "nClus"=c(nClus), "logPost"=c(logPost), "sweeps"=1:nSweeps, "chai

p_alpha <- ggplot(datag) +
  geom_line(aes(x=sweeps,y=alpha, group=chain, colour=chain),size=0.5, alpha=0.5) +
  labs(y="alpha") + labs(x="sweeps")

p_nclus <- ggplot(datag) +
  geom_line(aes(x=sweeps,y=nClus, group=chain, colour=chain),size=0.5, alpha=0.5) +
  labs(y="nClus") +  labs(x="sweeps")

p_logP <- ggplot(datag) +
  geom_line(aes(x=sweeps,y=logPost, group=chain, colour=chain),size=0.5, alpha=0.5) +
  labs(y="logPost") +  labs(x="sweeps")

(p_final <-  (p_alpha + guides(color="none", fill="none") +
               p_nclus + guides(color="none", fill="none") +
               p_logP)  +
    patchwork::plot_layout( nrow=3))
```



Prior for alpha

Let's look closer at the priors, especially the concentration parameter. By default, α is random. Run the model setting α to 3.

```
mod_nMMSE_alpha<-profRegr(yModel="Normal", # model type for the outcome
                          xModel="Mixed", # model type for the covariates
                          nSweeps=10000,
                          nClusInit=20,
                          nBurn=2000,
                          data=baseline,
                          output="alpha/output",
                          covNames = c("age_init", "CEP", "male"),
                          discreteCovs  = c("CEP", "male"),
                          continuousCovs = c("age_init"),
                          outcome = "nMMSE",
                          useHyperpriorR1 = FALSE, seed = 554248199,
                          alpha = 3,
                          dPitmanYor = 0) # equivalent to Dirichlet process prior

     dissimObj <- calcDissimilarityMatrix(mod_nMMSE_alpha)
     myheatDissMat(dissimObj, order = PSM1$rowInd)
```



```
clusObj_alpha <- calcOptimalClustering(dissimObj)
riskProfileObj <- calcAvgRiskAndProfile(clusObj_alpha)
clusterOrderObj <- plotRiskProfile(riskProfileObj, "./alpha/Cluster_profiles.png")
```

```
table(clusObj1$clustering, clusObj_alpha$clustering)
```

```
##
##       1   2   3   4
##   1 160   6   0   0
##   2   0  96   2  82
##   3   0   0  70  16
##   4  21  31   0  12
```

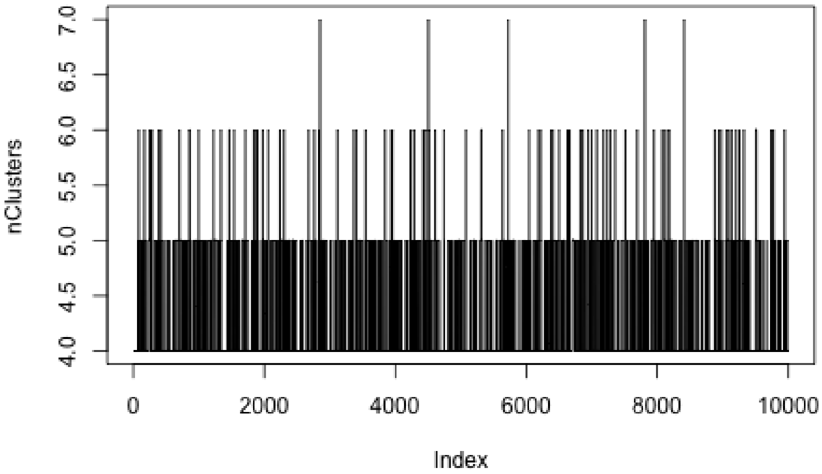
```
nClusChain <- mcmc(read.table("./alpha/output_nClusters.txt")[, 1])
getmode(nClusChain)
```

```
## [1] 12
```

```
3*log(1+500/3)
```

```
## [1] 15.36593
```

```
nClus <- plot_trace(mod_nMMSE_alpha, "nClusters", plot=TRUE)
```



Different label switching modes are available, in order to ensure a good mixing of the orderings:

- Move 1: swap labels of 2 randomly selected non-empty clusters
- Move 2: swap labels of 2 randomly selected neighbouring clusters, also swapping the v at the same time (Papaspiliopoulos and Roberts, 2008)
- Move 3: the idea is to simultaneously propose an update of the new cluster weights so they are something like their expected value conditional upon the new allocations (Hastie, Liverani, and Richardson, 2014).

```
whichLabelSwitch <- c("12", "3")
mod_LS <- list()
clus_LS <- list()

for(i in 1:length(whichLabelSwitch)){
  mod<-profRegr(yModel="Normal",
                xModel="Mixed",
                nSweeps=10000,
                nClusInit=20,
                nBurn=2000,
                data=baseline,
                output=paste("labelswitching/output",whichLabelSwitch[i],sep=''),
                covNames = c("age_init", "CEP", "male"),
                discreteCovs  = c("CEP", "male"),
                continuousCovs = c("age_init"),
                outcome = "nMMSE",
                useHyperpriorR1 = FALSE, seed = 554248199,
                whichLabelSwitch=whichLabelSwitch[i])

  dissimObj <- calcDissimilarityMatrix(mod)

  png(paste("./labelswitching/PSM_",whichLabelSwitch[i],sep=""))
  myheatDissMat(calcDissimilarityMatrix(mod), order = PSM1$rowInd)
  dev.off()

  clusObj <- calcOptimalClustering(dissimObj)
  riskProfileObj <- calcAvgRiskAndProfile(clusObj)
  clusterOrderObj <- plotRiskProfile(riskProfileObj, paste("./labelswitching/Cluster_profiles_", whichLa
  mod_LS <- c(mod_LS, list(mod))
  clus_LS <- c(clus_LS, list(clusObj))
}
```

Compare the clusterings obtained with the very first one (model mod_nMMSE_1):

```
table(clusObj1$clustering, clus_LS[[1]]$clustering)
```

```
##
##       1   2   3   4
##   1 160   6   0   0
##   2   0  93   3  84
##   3   0   0  71  15
##   4  20  32   0  12
```

```
table(clusObj1$clustering, clus_LS[[2]]$clustering)
```

```
##
##       1   2   3   4   5
##   1  24   6 136   0   0
##   2   3 111   0   0  66
##   3   0   5   0  69  12
##   4  37   0   0   0  27
```