Vignette MultiVarSel

Marie Perrot-Dockès, Céline Lévy-Leduc, Julien Chiquet

5 avril 2018

Introduction

This vignette explains how to use the package **MultiVarSel** which is dedicated to the variable selection in high-dimensional general linear models by taking into account the dependence that may exist between the columns of the observations matrix. The model can be describe as follow :

$$Y = XB + E,$$

where Y is a $q \times n$ matrix of responses, X a $p \times n$ matrix of covariables, B a sparse matrix of coefficients and E a random error matrix such that $\forall i \in (1, \cdots, n); E_i = (E_{i,1}, \ldots, E_{i,q}) \sim \mathcal{N}(0, \Sigma_q)$. We propose to estimate Σ_q in a first hand and then use it to apply a Lasso criterion to the model to estimate the position of the non nulle value of B. For further details on the methodology we refer the reader to [1].

The package has to be install and then load as follow :

```
#install.packages('MultiVarSel')
require(MultiVarSel)
```

```
## Loading required package: MultiVarSel
```

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
require(Matrix)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
require(glmnet)
```

```
## Loading required package: glmnet
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded glmnet 2.0-13
```

```
require(parallel)
```

```
## Loading required package: parallel
```

Numerical experiment

We first show an application of our methodology to a simulated data set. We start by generate a random error matrix E as describe in the Introduction.

```
n <- 30
q <- 200
p <- 5
rho <- 0.9
sparsity <- 0.01
Generate_sigma_sqrt <- function(q,rho) {
    diag <- sqrt(1-rho^2) * rho^(0:(q-2))
    diags <- lapply(0:(q-1), function(k) {
        return(c(rho^k, rep(diag[k+1],q-k-1)))
    })
    return(bandSparse(q,k=0:(q-1),diag=diags))
}

Sigma_sqrt <-  Generate_sigma_sqrt(q, rho)

white.noise   <- matrix(rnorm(q*n),n,q)
E <- as.matrix(white.noise %*% Sigma_sqrt )
```

We then generate a sparse matrix B of coefficients and a matrix of covariables.

```
s  <- round(sparsity*p*q)
ij <- arrayInd(sample(1:(p*q), size = s), c(p,q))
B <- sparseMatrix(i = ij[, 1], j = ij[, 2],
                  x = runif(s) * sample(c(-1,1),s,rep=T),
                  dims = c(p,q))

X <- matrix(rnorm(n*p),n,p)

Y <- X %*% B  + E
```

To apply our methodology we start by estimate the matrix E by calculate the residuals independamently on all the collumns of Y:

```
residual <- lm(as.matrix(Y) ~ X - 1)$residuals
```

We then test use a Portemanteau test to check if each row of this matrix is a white noise.

```
whitening_test(residual)
```

```
## [1] 0
```

The $p - value$ is really small we reject the hypothesis that each row of the residual is a white noise.

We then try to remoove the dependance among the residuals by estimate the covariance matrix of the lines of E. To estimate it we try different prior on the structure of this covariance. The simplest assumption is that each row of E follows an $AR(1)$ process, we also propose a modelisation where each row is an $ARMA(p,q)$ process and a nonparametric one where Σ is assumed to be Toeplitz. To compare this different estimation we perform a Portmenteau test on the matrix "whithened" $residuals\hat{\Sigma_q}^{-1/2}$.

```
result = whitening_choice(residual, c("AR1", "nonparam", "ARMA"), pAR = 1, qMA = 1)
result
```

```
##           Pvalue    Decision
## AR1        0.319 WHITE NOISE
## nonparam   0.699 WHITE NOISE
## ARMA 1 1   0.303 WHITE NOISE
```

We then select the easiest model that whithened the data, in that case the $AR(1)$ modelling. We compute the square root of the inverse of the estimator of the covariance matrix of each row of the residuals matrix using the $AR(1)$ modelling as follows :

```
square_root_inv_hat_Sigma = whitening(residual, "AR1", pAR = 1, qMA = 0)
```

To perform a variable selection we will transform our data to be abble to use the Lasso criterion introduce by Tibshirani in 1996, and available in the glmnet package.

In a linear model $\mathcal{Y} = \mathcal{X}\mathcal{B} + \mathcal{E}$, the Lasso estimator of $\mathcal{B}$ is

$$\widehat{\mathcal{B}}(\lambda) = \text{Argmin}_{\mathcal{B}} \left\{ \|\mathcal{Y} - \mathcal{X}\mathcal{B}\|_2^2 + \lambda \|\mathcal{B}\|_1 \right\},$$

where $\mathcal{Y}$ and $\mathcal{B}$ are vector, $\mathcal{E}$ is a white noise and $\mathcal{X}$ is a matrix.

In order to be abble to use the Lasso criterion we will use the operator vec to $\boldsymbol{Y}\widehat{\boldsymbol{\Sigma}}_q^{-1/2} = \boldsymbol{X}\boldsymbol{B}\widehat{\boldsymbol{\Sigma}}_q^{-1/2} + + \boldsymbol{E}\widehat{\boldsymbol{\Sigma}}_q^{-1/2}$

$$\begin{aligned}
\mathcal{Y} = vec(\boldsymbol{Y}\widehat{\boldsymbol{\Sigma}}_q^{-1/2}) &= vec(\boldsymbol{X}\boldsymbol{B}\widehat{\boldsymbol{\Sigma}}_q^{-1/2}) + vec(\boldsymbol{E}\widehat{\boldsymbol{\Sigma}}_q^{-1/2}) \\
&= ((\widehat{\boldsymbol{\Sigma}}_q^{-1/2})' \otimes \boldsymbol{X}) vec(\boldsymbol{B}) + vec(\boldsymbol{E}\widehat{\boldsymbol{\Sigma}}_q^{-1/2}) \\
&= \mathcal{X}\mathcal{B} + \mathcal{E}.
\end{aligned}$$

We are back to $\mathcal{Y}$ and $\mathcal{B}$ are vector, $\mathcal{E}$ is a white noise and $\mathcal{X}$ is a matrix. So we can apply the lasso criterion to estimate the non null position of $\mathcal{B} = vec(\boldsymbol{B})$ and deduct from it the non null position of B. In order to avoid the False positif we add a stability selection step. This different step are iplemented in the function variable selection of the MultiVarSel package.

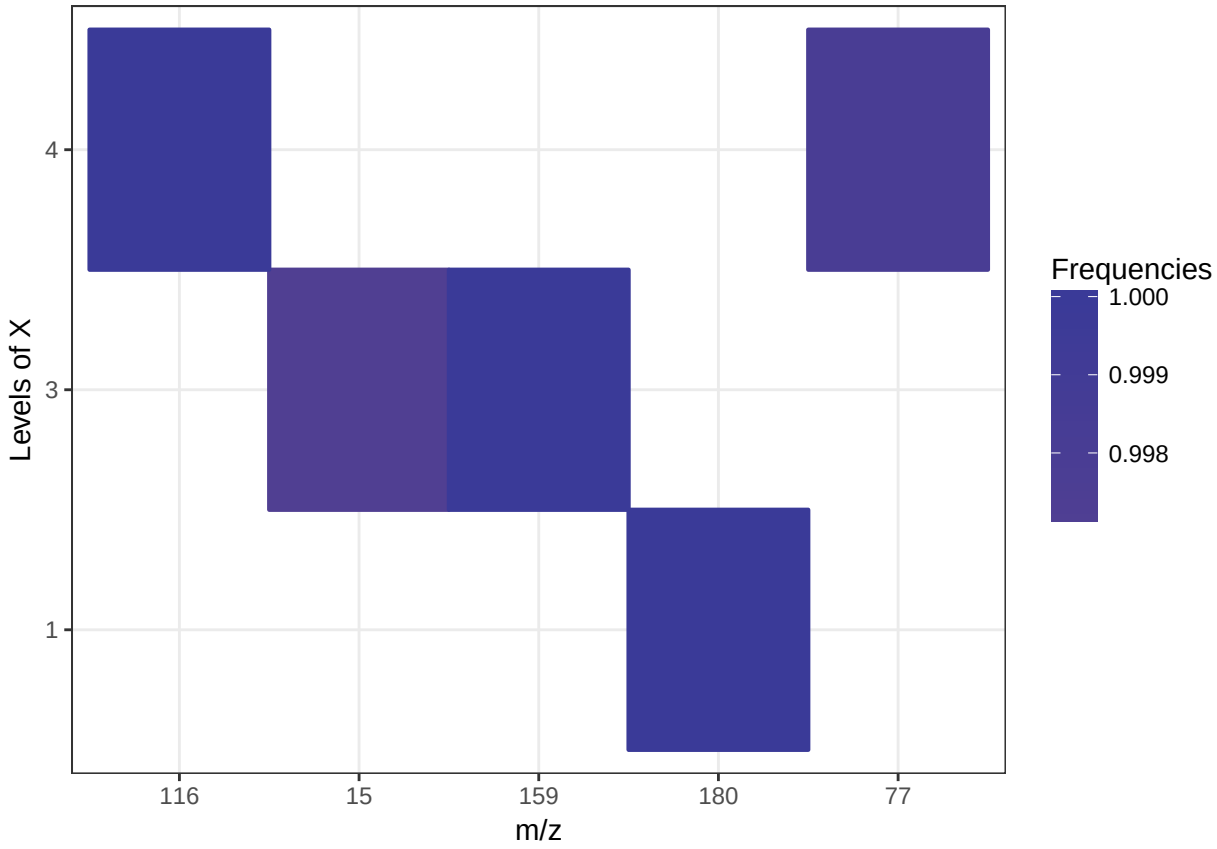
```
source('~/Documents/Multivar_selec/Multivar_selec/MultiVarSel/R/variable_selection.R', echo=TRUE)
```

```
##
## > variable_selection <- function(X, group = NULL, Y,
## +     nb_replis = 1000, nb.cores = 3, typeDep = "AR1", pAR = 1,
## +     qMA = 0) {
## +     if (!is. .... [TRUNCATED]
```

```
Frequencies=variable_selection(Y = Y, X = X, nb_repli = 100)
```

```
p <- ggplot(data = Frequencies[Frequencies$Frequencies >= 0.95, ],
            aes(x = Names_of_Y, y = Names_of_X, color = Frequencies, fill = Frequencies)) +
            geom_tile(size = 0.75) + scale_color_gradient2(midpoint = 0.95, mid = 'orange')  +
```

```
            theme_bw() + ylab('Levels of X') + xlab('m/z')
p
```



If we take a threshold at 0.95, meaning that we keep has non null value only the one that are kept in more than 95% of the times we have a True Positif Rate of 0.5 and a False Positive Rate 0.

An exemple in metabolomic

In this section we study a LC-MS (Liquid Chromatography-Mass Spectrometry) data set made of African copals samples. The samples correspond to ethanolic extracts of copals produced by trees belonging to two genera Copaifera (C) and Trachylobium (T) with a second level of classification coming from the geographical provenance of the Copaifera samples (West (W) or East (E) Africa). Since all the Trachylobium samples come from East Africa, we can use the modeling proposed in Equations (1) and (2) with C = 3 conditions: CE, CW and TE such that n CE = 9, n CW = 8 and n TE = 13. Our goal is to identify the most important features (the m/z values) for distinguishing the different conditions. In order to have a fast and reproducible exemple we focus on this section on the 200 first metaboliotes.

```
data("copals_camera")
Y <- scale(Y %>% as.matrix() %>% as.data.frame() %>% select(1:200))
```

We start by calculate the residuals of the ANOVA models on each of the metabolites independently.

```
residuals=lm(as.matrix(Y) ~ X - 1)$residuals
```

Then we test if the collum of the residuals are independant using the Portmanteau test.

```
whitening_test(residuals)
```

```
## [1] 5.676735e-229
```

The $p - value$ is really small and the fact that each lines of E is a whithe noise is rejected. We will try our different modelisation of the covariance of the residuals and see if one manage to remoove the dependance among the colluymn using a Portemanteau test.

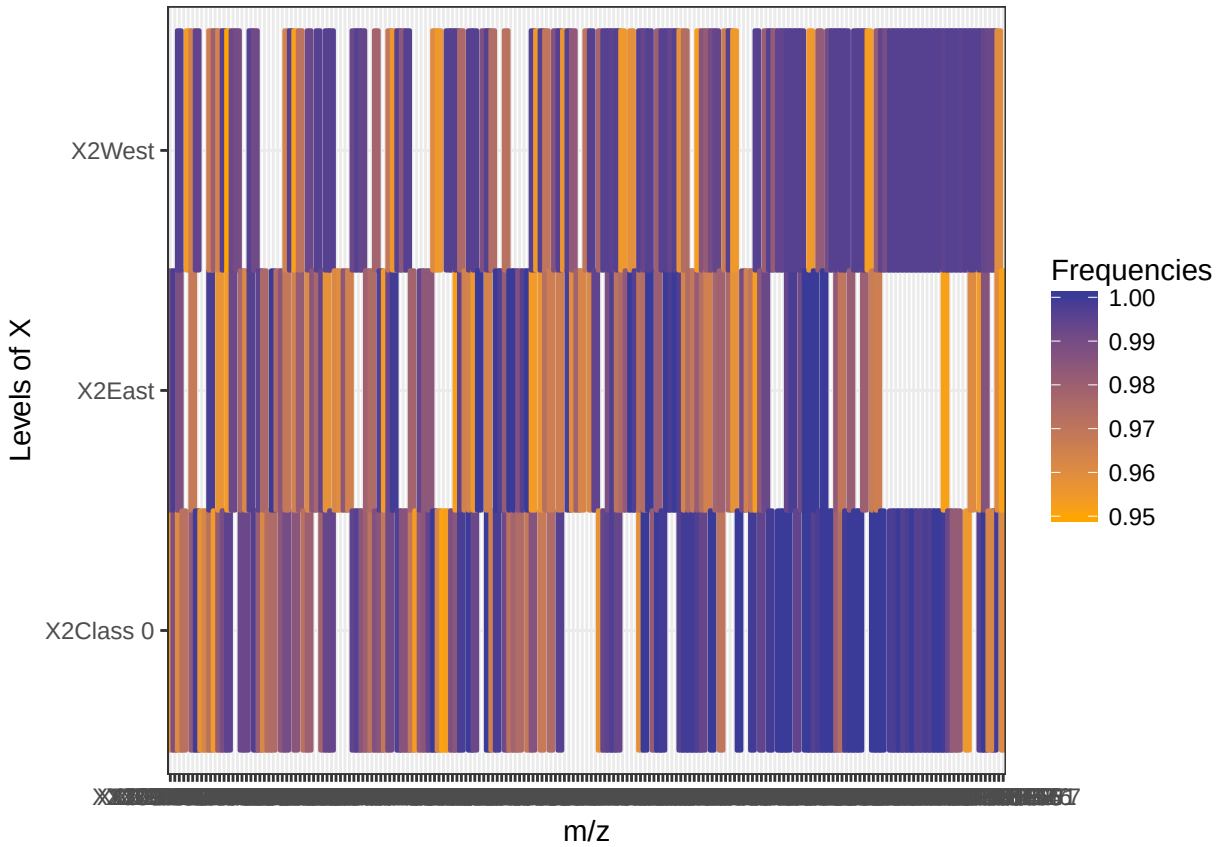
```
result=whitening_choice(residuals, c("AR1", "nonparam", "ARMA"), pAR = 1, qMA = 1)
result
```

```
##           Pvalue       Decision
## AR1            0 NO WHITE NOISE
## nonparam   0.992    WHITE NOISE
## ARMA 1 1   0.653    WHITE NOISE
```

The $AR(1)$ modelisation does not manage to remoove the dependance among the data but the two others are. We select the $ARMA(1,1)$ which is simplier than the non parametric.

```
Frequencies <- variable_selection(Y = Y, X = X, nb_repli = 100, typeDep = 'ARMA', pAR = 1, qMA = 1)
```

```
p <- ggplot(data = Frequencies[Frequencies$Frequencies >= 0.95, ],
            aes(x = Names_of_Y, y = Names_of_X, color = Frequencies, fill = Frequencies)) +
            geom_tile(size = 0.75) + scale_color_gradient2(midpoint = 0.95, mid = 'orange')  +
            theme_bw() + ylab('Levels of X') + xlab('m/z')
p
```



Hereafter, we also provide some information about the R session

```
sessionInfo()
```

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
```

```
## Running under: Ubuntu 16.04.4 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=fr_FR.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=fr_FR.UTF-8        LC_COLLATE=fr_FR.UTF-8
##  [5] LC_MONETARY=fr_FR.UTF-8    LC_MESSAGES=fr_FR.UTF-8
##  [7] LC_PAPER=fr_FR.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=fr_FR.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] glmnet_2.0-13     foreach_1.4.3     Matrix_1.2-11
##  [4] dplyr_0.7.4       purrr_0.2.3       readr_1.1.1
##  [7] tidyr_0.7.1       tibble_1.3.4      ggplot2_2.2.1
## [10] tidyverse_1.1.1   MultiVarSel_0.1.2
##
## loaded via a namespace (and not attached):
##  [1] reshape2_1.4.2   haven_1.1.0      lattice_0.20-35  colorspace_1.3-2
##  [5] htmltools_0.3.6  yaml_2.1.14      rlang_0.1.2      foreign_0.8-69
##  [9] glue_1.1.1       modelr_0.1.1     readxl_1.0.0     bindrcpp_0.2
## [13] bindr_0.1        plyr_1.8.4       stringr_1.2.0    munsell_0.4.3
## [17] gtable_0.2.0     cellranger_1.1.0 rvest_0.3.2      codetools_0.2-15
## [21] psych_1.7.8      evaluate_0.10.1  labeling_0.3     knitr_1.17
## [25] forcats_0.2.0    broom_0.4.2      Rcpp_0.12.13     scales_0.5.0
## [29] backports_1.1.1  jsonlite_1.5     mnormt_1.5-5     hms_0.3
## [33] digest_0.6.12    stringi_1.1.5    grid_3.4.4       rprojroot_1.2
## [37] tools_3.4.4      magrittr_1.5     lazyeval_0.2.0   pkgconfig_2.0.1
## [41] xml2_1.1.9000    lubridate_1.6.0  assertthat_0.2.0 rmarkdown_1.6
## [45] httr_1.3.1       iterators_1.0.8  R6_2.2.2         nlme_3.1-131.1
## [49] compiler_3.4.4
```