Package ‘armada’

April 4, 2019

Type Package

Title A Statistical Methodology to Select Covariates in
High-Dimensional Data under Dependence

Version 0.1.0

Description Two steps variable selection procedure in a context of high-dimensional dependent data
but few observations. First step is dedicated to eliminate dependence between variables (clustering
of variables, followed by factor analysis inside each cluster).
Second step is a variable selection using by aggregation of adapted methods.
Bastien B., Chakir H., Gegout-Petit A., Muller-Gueudin A., Shi Y.
A statistical methodology to select covariates in high-dimensional data under dependence.
Application to the classification of genetic profiles associated with outcome of a non-small-cell

License GPL-3

LazyData true

RoxygenNote 6.1.1

Imports stats, mvtnorm, ClustOfVar, FAMT, graphics, VSURF, glmnet,
anapuce, qvalue, parallel, doParallel, impute, ComplexHeatmap,
circlize

NeedsCompilation no

Author Aurelie Gueudin [aut, cre],
Anne Gegout-Petit [aut]

Maintainer Aurelie Gueudin <aurelie.gueudin@univ-lorraine.fr>

Repository CRAN

Date/Publication 2019-04-04 16:00:06 UTC

R topics documented:

ARMADA ................................................................. 2
ARMADA.heatmap .................................................. 3
ARMADA.select ...................................................... 4
ARMADA.summary .................................................. 6
clustering ............................................................. 6
Scores of all the covariates present in X, given the vector Y of the response.

Usage

ARMADA(X, Y, nclust = NULL, clusterType = c("PSOCK", "FORK"),
        parallel = FALSE)

Arguments

X the matrix (or data.frame) of covariates, dimension n*p (n is the sample size, p
the number of covariates). X must have rownames, which are the names of the
n subjects (i.e. the user ID of the n subjects). X must have colnames, which are
the names of the p covariates.

Y the vector of the response, length n.

nclust the number of clusters in the covariates dataset X.

clusterType to precise the type of cluster of the machine. Possible choices: "PSOCK", or
"FORK" (for UNIX or MAC systems, but not for WINDOWS).

parallel = TRUE if the calculus are made in parallel (default choice is FALSE).

Value

a 3-list with: "tree" which is the dendrogram of the data X, "nclust" which is a proposition of the
number of clusters in the data X, "result" which is a data.frame with p rows and 2 columns, the first
column gives the names of the covariates, the second column is the scores of the covariates.

Examples

library(ClustOfVar)
library(impute)
library(FAMT)
library(VSURF)
library(glmnet)
library(anapuce)
library(qvalue)
set.seed(1)
p <- 40
n <- 30
indexRow <- paste0("patient",1:n)
indexCol <- paste0("G",1:p)
X <- matrix(rnorm(p*n),ncol=p)
colnames(X) <- indexCol
rownames(X) <- indexRow
Y <- c(rep(-1,n/2), rep(1,n/2))
Y<-as.factor(Y)
resultat <- ARMADA(X,Y, nclust=1)
## Not run:
X<-toys.data$x
Y<-toys.data$y
result<-ARMADA(X,Y, nclust=2)
## End(Not run)

ARMADA.heatmap

Heatmap of the selected covariates.

Description

Heatmap of the selected covariates.

Usage

ARMADA.heatmap(X, Y, res.ARMADA.summary, threshold = 5)

Arguments

X
the matrix (or data.frame) of covariates, dimension n*p (n is the sample size, p
the number of covariates). X must have rownames, which are the names of the
n subjects (i.e. the user ID of the n subjects). X must have colnames, which are
the names of the p covariates.

Y
the vector of the response, length n.

res.ARMADA.summary
the result of the function ARMADA, or output of the function ARMADA.summary.

threshold
an integer between 0 and 8: the selected covariates are those which have a score
greater or equal to "threshold."

Details

This function plots the heatmap of the covariates which have a score higher than some threshold
chosen by the user, with respect to the values of Y.
Value

the plot of the heatmap, and a data.frame of the selected covariables.

Examples

library(ClustOfVar)
library(impute)
library(FAMT)
library(VSURF)
library(glmnet)
library(anapuce)
library(qvalue)
library(ComplexHeatmap)
library(circlize)
set.seed(1)
p <- 40
n <- 30
indexRow <- paste0("patient",1:n)
indexCol <- paste0("G",1:p)
X <- matrix(rnorm(p*n),ncol=p)
colnames(X) <- indexCol
rownames(X) <- indexRow
Y <- c(rep(-1,n/2), rep(1,n/2))
Y<-as.factor(Y)
resultat <- ARMADA(X,Y, nclust=1)
tracer <- ARMADA.heatmap(X, Y, resultat[[3]], threshold=5)
## Not run:
X<-toys.data$x
Y<-toys.data$y
result<-ARMADA(X,Y, nclust=2)
select<-ARMADA.heatmap(X, Y, result[[3]], threshold=5)

## End(Not run)
Arguments

**X**
the matrix (or data.frame) of covariates, dimension n*p (n is the sample size, p the number of covariates). X must have rownames and colnames.

**X.decorrele**
the matrix of decorrelated covariates, dimension n*p (n is the sample size, p the number of covariates). X.decorrele has been obtained by the function X_decor.

**Y**
the vector of the response, length n.

**test**
the type of test to apply ("wilox.test" or "t.test" if Y is a binary variable; "kruskal.test" or "anova" if Y is a factor with more than 2 levels; "cor.test" if Y is a continuous variable).

**type.cor.test**
if test="cor.test", precise the type of test (possible choices: "pearson","kendall", "spearman"). Default value is NULL, which corresponds to "pearson".

**type.measure_glmnet**
argument for the lasso regression. The lasso regression is done with the function cv.glmnet (package glmnet), and you can precise the type of data in cv.glmnet. Possible choices for type.measure.glmnet: "deviance" (for gaussian models, logistic, regression and Cox), "class" (for binomial or multinomial regression).

**family_glmnet**
argument for the lasso regression. The lasso regression is done with the function glmnet. Possible choices for family_glmnet: "gaussian" (if Y is quantitative), "binomial" (if Y is a factor with two levels), "multinomial" (if Y is a factor with more than two levels).

**clusterType**
to precise the type of cluster of the machine. Possible choices: "PSOCK" or "FORK" (for UNIX or MAC systems, but not for WINDOWS).

**parallel**
TRUE if the calculus are made in parallel.

Details

The function ARMADA.select applies 8 selection methods on the decorrelated covariates (named X.decorrele), given the variable of interest Y. It returns a list of 8 vectors of the selected covariates, each vector correspond to one selection method. The methods are (in the order): Random forest (threshold step), Random forest (interpretation step), Lasso, multiple testing with Bonferroni, multiple testing with Benjamini-Hochberg, multiple testing with qvalues, multiple testing with localfdr, FAMT.

Value

a list with 8 vectors, called: genes_rf_thres, genes_rf_interp, genes_lasso, genes_bonferroni, genes_BH, genes_qvalues, genes_localfdr, genes_FAMT. The 8 vectors are the selected covariates by the corresponding selection methods.
**ARMADA.summary**

*Scores of the covariates X*

**Description**

Scores of the covariates X

**Usage**

ARMADA.summary(X, resultat.ARMADA.select)

**Arguments**

- **X**
  - the matrix (or data.frame) of covariates, dimension n*p (n is the sample size, p the number of covariates). X must have colnames.

- **resultat.ARMADA.select**
  - the output of the ARMADA.select function: a list with 8 vectors, called: genes_rf_thres, genes_rf_interp, genes_lasso, genes_bonferroni, genes_BH, genes_qvalues, genes_localfdr, genes_FAMT. The 8 vectors are the selected covariates by the corresponding selection methods.

**Details**

The function ARMADA.summary gives the scores of all the covariates. The score of a variable is an integer between 0 and 8, and represents the number of selections of this variable by the 8 selection methods.

**Value**

gene_list: data.frame with p rows and 2 columns, the first column gives the names of the covariates, the second column is the scores of the covariates.

---

**clustering**

*To obtain the dendrogram of the covariates contained in the data.frame X, and a proposition for the number of clusters of covariates in X.*

**Description**

To obtain the dendrogram of the covariates contained in the data.frame X, and a proposition for the number of clusters of covariates in X.

**Usage**

clustering(X, plot = TRUE)
covariables

Arguments

  X       the matrix (or data.frame) of covariates, dimension n*p (n is the sample size, p the number of covariates).
  plot    if plot = TRUE" (default value): it gives the dendrogram and the plot of the height versus the number of clusters, for the 30 first clusters.

Value

a 2-list composed by: "tree" (the dendrogram of X), and "nclust" which is a proposition of the number of clusters. The proposed number of clusters is calculated as following: in the graph of the decreasing height versus the number of clusters, we define variation_height = (height[1:29]-height[2:30])/height[2:30], and our proposition is nclust=min(which(variation_height<0.05)). It is preferable that the user chooses its own number of clusters. Warning: nclust must be not too high. Indeed, if nclust is too high, the clusters contain a small number of covariates, and it is then possible that all the covariates of one or several cluster(s) are included in H0. In that case, the FAMT procedure will have a dysfunction.

Examples

toysNdata
x<toysNdata$X
covariables(x)

---

covariables  concatenation of the rownames of X and of the response vector Y.

Description

covariables  concatenation of the rownames of X and of the response vector Y.

Usage

covariables(X, Y)

Arguments

  X       the matrix (or data.frame) of covariates, dimension n*p (n is the sample size, p the number of covariates). X must have rownames, which are the names of the n subjects (i.e. the user ID of the n subjects).
  Y       the vector of the response, length n.

Details

internal function. Concatenation of the rownames of X (X is the matrix n*p of the covariates), and of the response vector Y. X must have rownames, which are the names of the n subjects (i.e. the user ID of the n subjects).
Value

a data.frame with dimension n*2: the first column gives the names of the subjects, and the second column is Y.

Examples

```r
X <- matrix(rnorm(50), nrow=10)
rownames(X) <- letters[1:10]
covariables(X, 1:10)
```

toys.data  

Description

toys.data is a simple simulated dataset of a binary classification problem, introduced by Weston et al..

Usage

toys.data

Format

An object of class list of length 2.

Details

- `$Y$`: output variable: a factor with 2 levels "-1" and "1";
- `$x$`: a data-frame containing input variables: with 30 obs. of 50 variables.

The data-frame $x$ is composed by 2 independent clusters, each cluster contains 25 correlated variables. It is an equiprobable two class problem, $Y$ belongs to -1,1, with 12 true variables (6 true variables in each cluster), the others being noise. The simulation model is defined through the conditional distribution of the $X^j$ for $Y=y$. In the first cluster, the $X^j$ are simulated in the following way:

- with probability 0.7, $X^j \sim N(y,2)$ for $j=1,2,3$, and $X^j \sim N(0,2)$ for $j=4,5,6$;
- with probability 0.3, $X^j \sim N(0,2)$ for $j=1,2,3$, and $X^j \sim N(y(j-3),2)$ for $j=4,5,6$;
- the other variables are noise, $X^j \sim N(0,1)$ for $j=7, \ldots, 25$.

The second cluster of 25 variables is simulated in a similar way.

Source

toys.data.multi

Examples

library(ClustOfVar)
library(impute)
library(FAMT)
library(VSURF)
library(glmnet)
library(anapuce)
library(qvalue)
X<-toys.data$x
Y<-toys.data$y
scoreX<-data.frame(c(rep(8,6),rep(0,19),rep(8,6),rep(0,19)))
rownames(scoreX)<-colnames(X)
select<-ARMADA.heatmap(X, Y, scoreX, threshold=1)
  ## Not run:
  result<-ARMADA(X,Y, nclust=2)
  select<-ARMADA.heatmap(X, Y, result[[3]], threshold=5)
  ## End(Not run)

Description

toys.data.multi is a simple simulated dataset of a multinomial classification problem.

Usage

toys.data.multi

Format

An object of class list of length 2.

Details

- $Y$: output variable: a factor with 3 levels "-1", "0", and "2";
- $x$ A data-frame containing input variables: with 60 obs. of 50 variables.

The data-frame $x$ is composed by 2 independent clusters, each cluster contains 25 correlated variables. It is an equiprobable three class problem, $Y$ belongs to -1,0,1. There is only 6 true variables, that are in the first cluster, the others being noise. The simulation model is defined through the conditional distribution of the $X^j$ for $Y=y$. In the first cluster, the $X^j$ are simulated in the following way:

- $X^j \sim N(2*y,2)$ for $j=1,2,3,4,5,6$;
- the other variables are noise, $X^j \sim N(0,1)$ for $j=7, \ldots, 25$.

The second cluster of 25 variables contains only noise variables.
Examples

```r
library(ClustOfVar)
library(impute)
library(FAMT)
library(VSURF)
library(glmnet)
library(anapuce)
library(qvalue)

X <- toys.data.multi$x
Y <- toys.data.multi$Y
scoreX <- data.frame(c(rep(8, 6), rep(0, 44)))
rownames(scoreX) <- colnames(X)

select <- ARMADA.heatmap(X, Y, scoreX, threshold = 1)

## Not run:
result <- ARMADA(X, Y, nclust = 2)
select <- ARMADA.heatmap(X, Y, result[[3]], threshold = 5)

## End(Not run)
```

Description

`toys.data.reg` is a simple simulated dataset of a regression problem.

Usage

`toys.data.reg`

Format

An object of class `list` of length 2.

Details

- `$Y$`: output variable;
- `$x$`: A data-frame containing input variables: with 30 obs. of 50 variables.

The data-frame `x` is composed by 2 independent clusters, each cluster contains 25 correlated variables. There is only 5 true variables, that are in the first cluster: \( Y = 50^*(x[.1] + x[.2] + x[.3] + x[.4] + x[.5]) \). The other variables are noise.
library(glmnet)
library(anapuce)
library(qvalue)
X<-toys.data.reg$x
Y<-toys.data.reg$Y
scoreX<-data.frame(c(rep(8,5),rep(8,45)))
rownames(scoreX)<-colnames(X)
select<-ARMADA.heatmap(X, Y, scoreX, threshold=1)
## Not run:
result<-ARMADA(X,Y, nclust=2)
select<-ARMADA.heatmap(X, Y, result[[3]], threshold=5)
## End(Not run)

---

### X_decor

Decorrelation of a matrix X, given a response variable Y.

#### Description

Decorrelation of a matrix X, given a response variable Y.

#### Usage

```r
X_decor(X, Y, tree = NULL, nclust = 1, maxnbfactors = 10)
```

#### Arguments

- **X**: the matrix (or data.frame) of covariates, dimension n*p (n is the sample size, p the number of covariates). X must have colnames and rownames.
- **Y**: the vector of the response, length n.
- **tree**: the dendrogram of the covariates (object obtained before by the function clustering). By default, tree=NULL.
- **nclust**: integer, the number of clusters in the covariates (1 by default).
- **maxnbfactors**: integer, the maximum number of factors in the clusters. By default: maxnbfactors=10.

#### Details

The function X_decor applies the factor analysis method FAMT in the different clusters of variables. The clusters must have been defined before (with the function "clustering").

#### Value

A matrix X.decorrele, with the same dimension, same rownames and same colnames than X.
Examples

```r
toys.data
X<-toys.data$x
Y<-toys.data$Y
Tree <- clustering(X,plot=FALSE)
nclust <- Tree[[2]]
tree <- Tree[[1]]
library(ClustOfVar)
library(FAMT)
X.deco<- X_decor(X, Y, tree, nclust, maxnbfactors=10)
```
Index

*Topic **datasets**
  toys.data, 8
  toys.data.multi, 9
  toys.data.reg, 10

ARMADA, 2
ARMADA.heatmap, 3
ARMADA.select, 4
ARMADA.summary, 6

clustering, 6
covariables, 7

toys.data, 8
toys.data.multi, 9
toys.data.reg, 10

X_decor, 11