Package 'CCA'

October 12, 2022

Type Package

Encoding latin1

Title Canonical Correlation Analysis

Version 1.2.1

Date 2009-03-11

Author Ignacio González, Sébastien Déjean

Maintainer Sébastien Déjean `<sebastien.dejean@math.univ-toulouse.fr>`

Description Provides a set of functions that extend the
'cancor' function with new numerical and graphical outputs. It
also include a regularized extension of the canonical
correlation analysis to deal with datasets with more variables
than observations.

Depends R (>= 2.10), fda, fields

License GPL (>= 2)

Repository CRAN

Date/Publication 2021-03-01 07:10:05 UTC

NeedsCompilation no

R topics documented:

CCA-package 2
cc 2
comput 4
estim.regul 5
img.estim.regul 6
img.matcor 6
loo 7
matcor 8
nutrimouse 9
plt.cc 10
plt.indiv 11
plt.var 12
rcc 12

Index 14

`CCA-package` *Canonical correlation analysis*

Description

The package provides a set of functions that extend the `cancor()` function with new numerical and graphical outputs. It includes a regularized extension of the canonical correlation analysis to deal with datasets with more variables than observations and enables to handle with missing values.

Details

Package:	CCA
Type:	Package
Version:	1.0
Date:	2007-02-02
License:	None

Author(s)

Ignacio González, Sébastien Déjean

Maintainer: Sébastien Déjean <sebastien.dejean@math.ups-tlse.fr>

References

www.lsp.ups-tlse.fr/CCA

`cc` *Canonical Correlation Analysis*

Description

The function performs Canonical Correlation Analysis to highlight correlations between two data matrices. It complete the `cancor()` function with supplemental numerical and graphical outputs and can handle missing values.

Usage

```
cc(X, Y)
```

Arguments

`X`	numeric matrix (n * p), containing the X coordinates.
`Y`	numeric matrix (n * q), containing the Y coordinates.

Details

The canonical correlation analysis seeks linear combinations of the 'X' variables which are the most correlated with linear combinations of the 'Y' variables.

Let PX and PY be the projector onto the respective column-space of X and Y. The eigenanalysis of PXPY provide the canonical correlations (square roots of the eigenvalues) and the coefficients of linear combinations that define the canonical variates (eigen vectors).

Value

A list containing the following components:

`cor`	canonical correlations
`names`	a list containing the names to be used for individuals and variables for graphical outputs
`xcoef`	estimated coefficients for the 'X' variables as returned by `cancor()`
`ycoef`	estimated coefficients for the 'Y' variables as returned by `cancor()`
`scores`	a list returned by the internal function `comput()` containing individuals and variables coordinates on the canonical variates basis.

Author(s)

Sébastien Déjean, Ignacio González

References

www.lsp.ups-tlse.fr/CCA

See Also

`rcc`, `plt.cc`

Examples

```
data(nutrimouse)
X=as.matrix(nutrimouse$gene[,1:10])
Y=as.matrix(nutrimouse$lipid)
res.cc=cc(X,Y)
plot(res.cc$cor,type="b")
plt.cc(res.cc)
```

`comput` *Additional computations for CCA*

Description

The `comput()` function can be viewed as an internal function. It is called by `cc()` and `rcc` to perform additional computations. The user does not have to call it by himself.

Usage

```
comput(X, Y, res)
```

Arguments

`X`	numeric matrix (n * p), containing the X coordinates.
`Y`	numeric matrix (n * q), containing the Y coordinates.
`res`	results provided by the `cc()` and `rcc()` functions.

Value

A list containing the following components:

`xscores`	X canonical variates
`yscores`	Y canonical variates
`corr.X.xscores`	Correlation bewteen X and X canonical variates
`corr.Y.xscores`	Correlation bewteen Y and X canonical variates
`corr.X.yscores`	Correlation bewteen X and Y canonical variates
`corr.Y.yscores`	Correlation bewteen Y and Y canonical variates

Author(s)

Sébastien Déjean, Ignacio González

See Also

`cc`, `rcc`

`estim.regul` *Estimate the parameters of regularization*

Description

Calulate the leave-one-out criterion on a 2D-grid to determine optimal values for the parameters of regularization.

Usage

```
estim.regul(X, Y, grid1 = NULL, grid2 = NULL, plt = TRUE)
```

Arguments

`X`	numeric matrix (n * p), containing the X coordinates.
`Y`	numeric matrix (n * p), containing the X coordinates.
`grid1`	vector defining the values of lambda1 to be tested. If NULL, the vector is defined as `seq(0.001, 1, length = 5)`
`grid2`	vector defining the values of lambda2 to be tested. If NULL, the vector is defined as `seq(0.001, 1, length = 5)`
`plt`	logical argument indicating whether an image should be plotted by calling the `img.estim.regul()` function.

Value

A 3-vector containing the 2 values of the parameters of regularization on which the leave-one-out criterion reached its maximum; and the maximal value reached on the grid.

Author(s)

Sébastien Déjean, Ignacio González

See Also

`loo`

Examples

```
#data(nutrimouse)
#X=as.matrix(nutrimouse$gene)
#Y=as.matrix(nutrimouse$lipid)
#res.regul = estim.regul(X,Y,c(0.01,0.5),c(0.1,0.2,0.3))
```

`img.estim.regul` *Plot the cross-validation criterion*

Description

This function provide a visualization of the values of the cross-validation criterion obtained on a grid defined in the function `estim.regul()`.

Usage

```
img.estim.regul(estim)
```

Arguments

`estim`	Object returned by `estim.regul()`.

Author(s)

Sébastien Déjean, Ignacio González

See Also

`estim.regul`

`img.matcor` *Image of correlation matrices*

Description

Display images of the correlation matrices within and between two data matrices.

Usage

```
img.matcor(correl, type = 1)
```

Arguments

`correl`	Correlation matrices as returned by the `matcor()` function
`type`	character determining the kind of plots to be produced: either one ((p+q) * (p+q)) matrix or three matrices (p * p), (q * q) and (p * q)

Details

Matrices are pre-processed before calling the `image()` function in order to get, as in the numerical representation, the diagonal from upper-left corner to bottom-right one.

Author(s)

Sébastien Déjean, Ignacio González

See Also

`matcor`

Examples

```
data(nutrimouse)
X=as.matrix(nutrimouse$gene)
Y=as.matrix(nutrimouse$lipid)
correl=matcor(X,Y)
img.matcor(correl)
img.matcor(correl,type=2)
```

`loo` *Leave-one-out criterion*

Description

The `loo()` function can be viewed as an internal function. It is called by `estim.regul()` to obtain optimal values for the two parameters of regularization.

Usage

```
loo(X, Y, lambda1, lambda2)
```

Arguments

`X`	numeric matrix (n * p), containing the X coordinates.
`Y`	numeric matrix (n * q), containing the Y coordinates.
`lambda1`	parameter of regularization for X variables
`lambda2`	parameter of regularization for Y variables

Author(s)

Sébastien Déjean, Ignacio González

See Also

`estim.regul`

`matcor` *Correlations matrices*

Description

The function computes the correlation matrices within and between two datasets.

Usage

```
matcor(X, Y)
```

Arguments

`X`	numeric matrix (n * p), containing the X coordinates.
`Y`	numeric matrix (n * q), containing the Y coordinates.

Value

A list containing the following components:

`Xcor`	Correlation matrix (p * p) for the X variables
`Ycor`	Correlation matrix (q * q) for the Y variables
`XYcor`	Correlation matrix ((p+q) * (p+q)) between X and Y variables

Author(s)

Sébastien Déjean, Ignacio González

See Also

`img.matcor`

Examples

```
data(nutrimouse)
X=as.matrix(nutrimouse$gene)
Y=as.matrix(nutrimouse$lipid)
correl=matcor(X,Y)
img.matcor(correl)
img.matcor(correl,type=2)
```

`nutrimouse` *Nutrimouse dataset*

Description

The `nutrimouse` dataset comes from a nutrition study in the mouse. It was provided by Pascal Martin from the Toxicology and Pharmacology Laboratory (French National Institute for Agronomic Research).

Usage

```
data(nutrimouse)
```

Format

A list containing the following components:

- `gene`: data frame (40 * 120) with numerical variables
- `lipid`: data frame (40 * 21) with numerical variables
- `diet`: factor vector (40)
- `genotype`: factor vector (40)

Details

Two sets of variables were measured on 40 mice:

- expressions of 120 genes potentially involved in nutritional problems.
- concentrations of 21 hepatic fatty acids.
  The 40 mice were distributed in a 2-factors experimental design (4 replicates):
- Genotype (2-levels factor): wild-type and PPARalpha -/-
- Diet (5-levels factor): Oils used for experimental diets preparation were corn and colza oils (50/50) for a reference diet (REF), hydrogenated coconut oil for a saturated fatty acid diet (COC), sunflower oil for an Omega6 fatty acid-rich diet (SUN), linseed oil for an Omega3-rich diet (LIN) and corn/colza/enriched fish oils for the FISH diet (43/43/14).

Source

P. Martin, H. Guillou, F. Lasserre, S. Déjean, A. Lan, J-M. Pascussi, M. San Cristobal, P. Legrand, P. Besse, T. Pineau - Novel aspects of PPARalpha-mediated regulation of lipid and xenobiotic metabolism revealed through a nutrigenomic study. Hepatology, in press, 2007.

References

www.inra.fr/internet/Centres/toulouse/pharmacologie/pharmaco-moleculaire/acceuil.html

Examples

```
data(nutrimouse)
boxplot(nutrimouse$lipid)
```

`plt.cc` *Graphical outputs for canonical correlation analysis*

Description

This function calls either `plt.var()` or `plt.indiv()` or both functions to provide individual and/or variable representation on the canonical variates.

Usage

```
plt.cc(res, d1 = 1, d2 = 2, int = 0.5, type = "b", ind.names = NULL,
var.label = FALSE, Xnames = NULL, Ynames = NULL)
```

Arguments

`res`	Object returned by `cc()` or `rcc()`
`d1`	The dimension that will be represented on the horizontal axis
`d2`	The dimension that will be represented on the vertical axis
`int`	The radius of the inner circle
`type`	Character "v" (variables), "i" (individuals) or "b" (both) to specifying the plot to be done.
`ind.names`	vector containing the names of the individuals
`var.label`	logical indicating whether label should be plotted on the variables representation
`Xnames`	vector giving the names of X variables
`Ynames`	vector giving the names of Y variables

Author(s)

Sébastien Déjean, Ignacio González

References

www.lsp.ups-tlse.fr/Biopuces/CCA

See Also

`plt.indiv`, `plt.var`

Examples

```
data(nutrimouse)
X=as.matrix(nutrimouse$gene[,1:10])
Y=as.matrix(nutrimouse$lipid)
res.cc=cc(X,Y)
plt.cc(res.cc)
plt.cc(res.cc,d1=1,d2=3,type="v",var.label=TRUE)
```

`plt.indiv` *Individuals representation for CCA*

Description

This function provides individuals representation on the canonical variates.

Usage

```
plt.indiv(res, d1, d2, ind.names = NULL)
```

Arguments

`res`	Object returned by `cc()` or `rcc()`
`d1`	The dimension that will be represented on the horizontal axis
`d2`	The dimension that will be represented on the vertical axis
`ind.names`	vector containing the names of the individuals

Author(s)

Sébastien Déjean, Ignacio González

References

www.lsp.ups-tlse.fr/Biopuces/CCA

See Also

`plt.var`, `plt.cc`

`plt.var` *Variables representation for CCA*

Description

This function provides variables representation on the canonical variates.

Usage

```
plt.var(res, d1, d2, int = 0.5, var.label = FALSE, Xnames = NULL, Ynames = NULL)
```

Arguments

`res`	Object returned by `cc` or `rcc`
`d1`	The dimension that will be represented on the horizontal axis
`d2`	The dimension that will be represented on the vertical axis
`int`	The radius of the inner circle
`var.label`	logical indicating whether label should be plotted on the variables representation
`Xnames`	vector giving the names of X variables
`Ynames`	vector giving the names of Y variables

Author(s)

Sébastien Déjean, Ignacio González

References

www.lsp.ups-tlse.fr/Biopuces/CCA

See Also

`plt.indiv`, `plt.cc`

`rcc` *Regularized Canonical Correlation Analysis*

Description

The function performs the Regularized extension of the Canonical Correlation Analysis to seek correlations between two data matrices when the number of columns (variables) exceeds the number of rows (observations)

Usage

```
rcc(X, Y, lambda1, lambda2)
```

Arguments

`X`	numeric matrix (n * p), containing the X coordinates.
`Y`	numeric matrix (n * q), containing the Y coordinates.
`lambda1`	Regularization parameter for X
`lambda2`	Regularization parameter for Y

Details

When the number of columns is greater than the number of rows, the matrice X'X (and/or Y'Y) may be ill-conditioned. The regularization allows the inversion by adding a term on the diagonal.

Value

A list containing the following components:

`corr`	canonical correlations
`names`	a list containing the names to be used for individuals and variables for graphical outputs
`xcoef`	estimated coefficients for the 'X' variables as returned by `cancor()`
`ycoef`	estimated coefficients for the 'Y' variables as returned by `cancor()`
`scores`	a list returned by the internal function comput() containing individuals and variables coordinates on the canonical variates basis.

Author(s)

Sébastien Déjean, Ignacio González

References

Leurgans, Moyeed and Silverman, (1993). Canonical correlation analysis when the data are curves. J. Roy. Statist. Soc. Ser. B. 55, 725-740.

Vinod (1976). Canonical ridge and econometrics of joint production. J. Econometr. 6, 129-137.

See Also

`cc`, `estim.regul`, `plt.cc`

Examples

```
data(nutrimouse)
X=as.matrix(nutrimouse$gene)
Y=as.matrix(nutrimouse$lipid)
res.cc=rcc(X,Y,0.1,0.2)
plt.cc(res.cc)
```

Index

∗ **datasets**
    nutrimouse, 9
∗ **dplot**
    img.estim.regul, 6
    img.matcor, 6
    plt.cc, 10
    plt.indiv, 11
    plt.var, 12
∗ **multivariate**
    cc, 2
    CCA-package, 2
    comput, 4
    estim.regul, 5
    loo, 7
    matcor, 8
    rcc, 12

cc, 2, *4*, *13*
CCA *(*CCA-package*)*, 2
CCA-package, 2
comput, 4

estim.regul, 5, *6*, *7*, *13*

img.estim.regul, 6
img.matcor, 6, *8*

loo, *5*, 7

matcor, *7*, 8

nutrimouse, 9

plt.cc, *3*, 10, *11–13*
plt.indiv, *10*, 11, *12*
plt.var, *10*, *11*, 12

rcc, *3*, *4*, 12