Package ‘plsr’

February 26, 2019

Type Package
Title Pleasure - Partial Least Squares Analysis with Permutation Testing
Version 0.0.1
Description Provides partial least squares analysis for the analysis of the relation between two high-dimensional data sets. Includes permutation testing and bootstrapping for resulting latent variables (following McIntosh & Lobaugh (2004) <doi:10.1016/j.neuroimage.2004.07.020>) and several visualization functions.
Depends R (>= 2.10)
License GPL-3
Encoding UTF-8
LazyData true
Imports boot, ggplot2, reshape2, shiny
RoxygenNote 6.1.1
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Jan Niklas Schneider [aut, cre], Timothy Raymond Brick [aut]
Maintainer Jan Niklas Schneider <jan_schneider@live.de>
Repository CRAN
Date/Publication 2019-02-26 13:30:03 UTC

R topics documented:

biplot.plsr ................................................................. 2
connections ............................................................. 3
explained_variance ..................................................... 3
loadings ................................................................. 4
biplot.plsr

### Description

Produces a biplot from a pls object

### Usage

```r
## S3 method for class 'pls'
biplot(x, side = "X", LVs = c(1, 2), ...)
```

### Arguments

- **x**
  - The pls object
- **side**
  - The side for which the biplot should be generated. Can be "X" (default) to generate a biplot of the loadings of X onto the latent space or "Y" for the loadings of Y.
- **LVs**
  - Vector of length two which specifies the latent variables to be plotted against each other. For example, the default LVs=c(1,2) will plot latent variable 1 against latent variable 2.
- **...**
  - optional arguments to be passed to `biplot.default`

### Examples

```r
plsr_obj <- pls(rating_data, tracking_data, 10, 10)
biplot(plsr_obj)
biplot(plsr_obj, LV=c(2,3), side="Y")
```
connections

Connection information needed to draw segments between face tracking points.

Usage
connections

Format
An object of class data.frame with 63 rows and 2 columns.

Author(s)
Jan Niklas Schneider <jan_schneider@live.de>

explained_variance

Description
Calculates explained variance per component of the original data sets X and Y.

Usage
explained_variance(plsr_obj)

Arguments
plsr_obj A plsr object

Value
A list containing the elements ExpVarX and ExpVarY, which contain the explained variances for X and Y respectively.

Examples
plsr_object = pls(rating_data,tracking_data,10,10)
explained_variance(plsr_object)
loadings  

*Print loadings of plsr object*

**Description**

This will print the loading matrices V and U that project from original data spaces X and Y to latent space.

**Usage**

```r
loadings(x, mat = NULL)
```

**Arguments**

- `x`: A plsr object.
- `mat`: Which matrix to print. Can be "V" or "U", if NULL (default) will print both.

**Examples**

```r
plsr_obj <- pls(rating_data, tracking_data, 10, 10)
loadings(plsr_obj)  # show V and U
loadings(plsr_obj, "V")  # show V only
loadings(plsr_obj, "U")  # show U only
```

---

new_plsr  

*Constructor for plsr objects*

**Description**

Constructor for plsr objects

**Usage**

```r
new_plsr(decomp = list(), perm = list(), bootstrap = list(),
         sclng = list(), org_dat = list(), cl = list())
```

**Arguments**

- `decomp`: List of singular value decomposition results
- `perm`: List of permutation testing results
- `bootstrap`: List of bootstrapping results
- `sclng`: List of scaling parameters applied to original data
- `org_dat`: List of original data
- `cl`: Call of pls function
**permutation_precision**

**Examples**

```r
# Creating an empty pls object
d = b = s = o = c = list()
pls_obj = new_plsr(decomp = d, perm = p, bootstrap = b, sclng = s, org_dat = o, cl = c)
```

**Description**

Following Ojala & Garriga (2010): "Permutation tests for studying classifier performance"

**Usage**

```r
permutation_precision(p, k)
```

**Arguments**

- `p`: The p value
- `k`: Number of permutation iterations

**Value**

The precision given `p` and `k`.

**Examples**

```r
permutation_precision(0.05, 1000)
permutation_precision(0.01, 1000)
```

---

**plot.plsr**

*Plot function for pls objects*

**Description**

Plots information about a pls object. The following plots will be generated:

- barplot of p-values of latent variables estimated via permutation testing
- Histograms of the distributions of latent variables derived via permutation testing
- A plot showing the effect of the first latent variable on the original data spaces
- Several plots to visualize bootstrapping results
Usage

```r
## S3 method for class 'plsr'
plot(x, ...)
```

Arguments

- `x` The plsr object.
- `...` Further arguments.

Examples

```r
plsr_obj <- pls(rating_data, tracking_data)
plot(plsr_obj) # will open several plots and requires user input inbetween
```

---

**plot_default**

Default plot function for plsr shiny app

Description

Default plot function for plsr shiny app

Usage

```r
plot_default(x, time_steps = 10, t)
```

Arguments

- `x` A vector of predicted data to plot
- `time_steps` Number of time steps
- `t` Which time step to plot
plot_explained_variance

Plot explained variance of plsr object

Description
Calculates and plots the variance explained in the original data X and Y by each additional latent variable.

Usage
plot_explained_variance(plsr_obj)

Arguments
plsr_obj The plsr object.

Examples
plsr_obj = pls(rating_data, tracking_data, 10, 10)
plot_explained_variance(plsr_obj)

plot_latent_variables Plots latent variables

Description
This function will plot the effects of increasing and decreasing one or several latent variables by the specified standard deviation.

Usage
plot_latent_variables(plsr_obj, lv_num = 1, sd = 3, FUN = c(barplot, barplot), args1 = NULL, args2 = NULL)

Arguments
plsr_obj A plsr object
lv_num An integer or list of integer specifying which latent variables to plot.
sd Range in standard deviations from +[sd] to -[sd].
FUN A vector containing two functions, which will be used for plotting the results of changes in the latent variable(s) in X and Y. Default is c(barplot, barplot).
args1 Arguments for the plotting function in FUN[1]
args2 Arguments for the plotting function in FUN[2]
Examples

```r
plsr_obj = pls(rating_data, tracking_data, 10, 10)

# plot latent variable effect with barplots (default) for X and Y side
plot_latent_variables(plsr_obj)

# plot latent variables with barplots for the X side and
# a custom plot function tailored to face tracking data for the Y side
plot_latent_variables(plsr_obj, lv=1:2, sd=2, FUN=c(barplot, pls::plot_frame))

# same as above but with additional arguments passed to the plotting functions
plot_latent_variables(plsr_obj, FUN = c(barplot, pls::plot_frame),
  args1=list(col="red"), args2 = list(single_frame=5))
```

---

plot_perm_distr

Plots null distributions constructed via permutation testing

Description

Plots histograms of the null distribution for values of singular values of latent variables constructed via permutation testing.

Usage

```r
plot_perm_distr(plsr_obj, ..., lwd = 2, bar_col = "grey",
  line_col = "red")
```

Arguments

- `plsr_obj`: A pls object.
- `...`: Further parameters to be passed to `hist`.
- `lwd`: Line width of vertical line indicating the estimated value of the singular value.
- `bar_col`: Color of the bars in the histograms.
- `line_col`: Color of the vertical line indicating the estimated value of the singular value.

Examples

```r
plsr_obj = pls(rating_data, tracking_data, 10, 10)
plot_perm_distr(plsr_obj)

plot_perm_distr(plsr_obj, breaks=5, lwd=5, bar_col = "white", line_col = "green")
```
plot_perm_results

Plot permutation results for plsr object

Description

Plots the p-values for the latent variables estimated through permutation testing.

Usage

plot_perm_results(plsr_obj, ..., alpha = NULL,
                   main = "Permutation Testing Results", lwd = 2, col = "red")

Arguments

- **plsr_obj**: A plsr_obj.
- **...**: Additional arguments passed to barplot.
- **alpha**: The significance threshold used. Will be indicated in the plot by a horizontal line. If NULL (default), the alpha value of the plsr object will be used.
- **main**: The title of the plot.
- **lwd**: The line width of the line indicating alpha.
- **col**: The color of the line indicating alpha.

Examples

plsr_obj = pls(rating_data, tracking_data, 10, 10)
plot_perm_results(plsr_obj)

#plot with 0.10 as the significance threshold instead of the one specified by the plsr object
#and a thicker blue-colored line to indicate it
plot_perm_results(plsr_obj, lwd=5, col="blue", alpha=0.10)

pls

Run partial least squares analysis

Description

This is the main function of the plsr package. It will calculate a partial least squares solution for the provided data and perform permutation testing and bootstrapping on the resulting latent variables. Results will be saved as a plsr object.

Usage

pls(X, Y, n_perm = 100, n_boot = 100, scale = T, verbose = F,
    alpha = 0.05)
predict.plsr

Arguments

- **X**: A matrix of m observations on n_x variables.
- **Y**: A matrix of m observations on n_y dimensions.
- **n_perm**: Number of permutation iterations. Default is 100.
- **n_boot**: Number of bootstrap iterations. Default is 100.
- **scale**: Scaling of X and Y (Boolean).
- **verbose**: Provides additional output.
- **alpha**: The significance level for permutation testing.

Value

A plsR Object.

Examples

```r
X = matrix(rnorm(300), ncol = 3)
Y = matrix(rnorm(1000), ncol = 10)
pls(X,Y)
pls(X,Y, n_perm = 10, n_boot = 10)

#running pls function on included data of the package
plsr_obj=pls(rating_data,tracking_data,1000,1000)
#inspecting results:
plot(plsr_obj)
summary(plsr_obj)
```

---

**predict.plsr**

**Predict from a plsR object**

Description

This function can be used to make predictions from one original data space to the other. Prediction
direction can be forward, meaning X to Y direction and backward, meaning Y to X prediction.

Usage

```r
## S3 method for class 'plsR'
predict(object, new_data, direction = "forward", ...)
```

Arguments

- **object**: A plsR object.
- **new_data**: The data from which you want to predict.
- **direction**: The direction of prediction. Default is "forward" meaning X to Y. Every other argument will result in backward prediction.
- **...**: Additional arguments.
**print.plsr**

### Examples

```r
plsr_obj = pls(rating_data, tracking_data, 10, 10)
prediction = predict(plsr_obj, runif(7, 1, 101), "forward")

# visualizing results with face tracking data specific function
plsr:::plot_frame(prediction)
```

---

**print.plsr**

*Print plsr object*

### Description

Prints information about a plsr object.

### Usage

```r
## S3 method for class 'plsr'
print(x, ...)
```

### Arguments

- `x` A plsr object.
- `...` Further arguments.

### Examples

```r
X = matrix(rnorm(300), ncol=3)
Y = matrix(rnorm(1000), ncol = 10)
plsr_obj = pls(X, Y)
print(plsr_obj)
```

---

**rating_data**

*Emotion ratings on the dimensions happy, sad, surprised, disgusted, angry, fearful and interested*

### Description

Emotion ratings on the dimensions happy, sad, surprised, disgusted, angry, fearful and interested

### Usage

```r
rating_data
```

### Format

An object of class data.frame with 40 rows and 7 columns.
Author(s)

Jan Niklas Schneider <jan_schneider@live.de>

rating_data_emo_means  

Mean emotion ratings on the dimensions happy, sad, surprised, disgusted, angry, fearful and interested

Description

Mean emotion ratings on the dimensions happy, sad, surprised, disgusted, angry, fearful and interested

Usage

rating_data_emo_means

Format

An object of class data.frame with 40 rows and 7 columns.

Author(s)

Jan Niklas Schneider <jan_schneider@live.de>

summary.plsr  

Summary of plsr object

Description

Summary of plsr object

Usage

## S3 method for class 'plsr'
summary(object, ...)

Arguments

object  
A plsr object.

...  
Further arguments.

Examples

plsr_obj = pls(rating_data,tracking_data,10,10)
summary(plsr_obj)
tracking_data

Face tracking data on 40 videos of emotional facial expressions.

Description

Face tracking data on 40 videos of emotional facial expressions.

Usage

tracking_data

Format

An object of class data.frame with 40 rows and 1360 columns.

Author(s)

Jan Niklas Schneider <jan_schneider@live.de>

tracking_data_emo_means

Face tracking data on 40 videos of emotional facial expressions.

Description

Face tracking data on 40 videos of emotional facial expressions.

Usage

tracking_data_emo_means

Format

An object of class data.frame with 40 rows and 1360 columns.

Author(s)

Jan Niklas Schneider <jan_schneider@live.de>
Index

*Topic datasets
  connections, 3
  rating_data, 11
  rating_data_emo_means, 12
  tracking_data, 13
  tracking_data_emo_means, 13

biplot.plsr, 2
connections, 3
explained_variance, 3
loadings, 4
new_plsr, 4
permutation_precision, 5
plot.plsr, 5
plot_default, 6
plot_explained_variance, 7
plot_latent_variables, 7
plot_perm_distr, 8
plot_perm_results, 9
pls, 9
predict.plsr, 10
print.plsr, 11

rating_data, 11
rating_data_emo_means, 12

summary.plsr, 12

tracking_data, 13
tracking_data_emo_means, 13