Package 'prinsimp'

February 20, 2015

Type Package

Title Finding and plotting simple basis vectors for multivariate data

Version 0.8-8

Date 2013-11-01

Author Davor Cubranic, Jonathan Zhang, Nancy Heckman, Travis Gaydos, and J.S. Marron

Maintainer Davor Cubranic <cubranic@stat.ubc.ca>

Description Provides capabilities beyond principal components analysis to focus on finding structure in low variability subspaces. Constructs and plots simple basis vectors for pre-defined and user-defined measures of simplicity.

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-02 01:21:46

R topics documented:

`prinsimp-package` *plyr: the split-apply-combine paradigm for R.*

Description

The plyr package is a set of clean and consistent tools that implement the split-apply-combine pattern in R. This is an extremely common pattern in data analysis: you solve a complex problem by breaking it down into small pieces, doing something to each piece and then combining the results back together again.

Details

Principal Component Analysis (PCA) is simply an eigenanalysis of a covariance matrix, G. Its eigenvalues λ_j can be interpreted as the variance of G in the direction of the eigenvector v_j, and $\lambda_j / \sum \lambda_k$ as the proportion of variance explained by v_j. Often, G is well-approximated using the first M eigenvectors and eigenvalues, which we call the *model space*.

The orthogonal complement of the model space (that is, the space spanned by the remaining eigenvectors), we call the *nearly null* space. The nearly null space is interesting as a space of low variability, which may be particularly important in, for instance, evolutionary biology. This package provides functions for analyzing the nearly null space and finding interesting structures of low variability, as defined by a quadratic simplicity measure. It is an expanded reimplementation in R of the method described by Gaydos et al (2013).

References

T.L. Gaydos, N.E. Heckman, M. Kirkpatrick, J.R. Stinchcombe, J. Schmitt, J. Kingsolver, J.S. Marron. (2013). Visualizing genetic constraints. *Annals of Applied Statistics* 7: 860-882.

`basisplot` *Basisplot method for Simple Partition*

Description

Produces the plots for the specified basis vectors of the class `simpart`

Usage

```
basisplot(x, display = list(model=TRUE, simple=TRUE), ...)
```

Arguments

`x`	an object of class `"simpart"`, typically result of `simpart`.
`display`	a list. Specifies which model and nearly null space simplicity basis vectors to print.
`...`	other parameters passed through to plotting functions

See Also

varsimp, varperc, simpart

Examples

```
library(prinsimp)
require(graphics)

## Caterpillar estimated covariance from Kingsolver et al (2004)
## Measurements are at times 11, 17, 23, 29, 35, 40
data(caterpillar)

cat.sim <- simpart(caterpillar, simpledim = 2,
                   x = c(11, 17, 23, 29, 35, 40), cov = TRUE)

## Plots the 4 model basis vectors and 2 simplicity basis vectors consecutively
basisplot(cat.sim)

## Display the 4 model basis vectors on one page
par(mfrow = c(2,2))
basisplot(cat.sim, display = list(model=1:4))
```

---

caterpillar *Kingsolver* et al. *caterpillar estimated covariance*

Description

Estimated broad-sense genetic variance-covariance matrix for short-term growth rate at different temperatures in fourth-instar caterpillars from Kingsolver et al (2004).

Measurements are at temperatures 11, 17, 23, 29, 35, 40.

Usage

```
data(caterpillar)
```

Format

A 6-by-6 estimated genetic covariance matrix. The matrix is obtained from the printed version of the paper and, due to rounding, it is not positive definite – one of its eigenvalues is negative. The functions in this package, such as `simpart` handle this by setting all negative eigenvalues to zero and reconstructing the covariance matrix before proceeding.

Source

Kingsolver, J. G., Ragland, G. J. and Shlichta, J. G. (2004). Quantitative genetics of continuous reaction norms: thermal sensitivity of caterpillar growth rates. *Evolution*, 58:1521–1529.

Examples

```
library(prinsimp)

data(caterpillar)
cat.sim <- simpart(caterpillar, simpledim = 2,
                   x = c(11, 17, 23, 29, 35, 40), cov = TRUE)
plot(cat.sim)
```

`plot.simpart` *Plot method for Simple Partition*

Description

Produces plots for the model basis vectors and the vectors forming the simplicity basis for the nearly null space, along with the variance-simplicity view and the percent of variance explained panel. Only six basis vectors can be displayed at a time.

Usage

```
## S3 method for class 'simpart'
plot(x, display = list(model=TRUE, simple=TRUE), layout, ...)
```

Arguments

`x`	an object of class `"simpart"`, typically result of `simpart`.
`display`	a list. Specifies which model and simplicity basis vectors to plot. Must be consistent with `x`.
`layout`	a matrix of locations for the component plots, following the format used in the `layout` function. The default is to draw basis plots in the left half of the figure area, arranged in a U-curve with model basis starting at the top of the left arm and simple basis at the top right. The variance-simplicity view and percent of variance explained panel take up the right half of the figure. The subplots are drawn in the following order: first model then simple basis, followed by the variance-simplicity plot, and variance-explained last.
`...`	other parameters passed through to plotting functions

Details

The variance-simplicity view and the percent of variance explained panel are always produced everytime `plot.simpart` is called. The variance-simplicity view plots the percent of variance explained and the simplicity values of the basis vectors specified in `display`. If the number of basis vectors is greater than 6 and the user specifies more than 6 in `display`, an error message is generated. In `display`, non-existent basis numbers are ignored.

See Also

`basisplot`, `varsimp`, `varperc`, `simpart`

Examples

```
library(prinsimp)
require(graphics)

## Caterpillar estimated covariance matrix from Kingsolver et al (2004)
## Measurements are at temperatures 11, 17, 23, 29, 35, 40
data(caterpillar)

## Default plot method for a 2-dimensional nearly null space
cat.sim <- simpart(caterpillar, simpledim = 2,
                   x = c(11, 17, 23, 29, 35, 40), cov = TRUE)
plot(cat.sim)

## Plot only the two simplicity basis vectors in the 2-dimensional
## nearly null space
plot(cat.sim, display = list(simple=1:2))

## Try, by mistake to plot first four simple vectors when simpledim=2.
## Will plot only 2.
plot(cat.sim, display = list(simple=1:4))

## Only plot the first two model basis vectors, the variance simplicity
## view, and the percent variance explained panel using the 'layout'
## argument
plot(cat.sim, display = list(model=1:2), layout = matrix(1:4, nrow=2, ncol=2))
```

---

print.simpart — *Print method for Simple Partition*

Description

The `print` method for class `"simpart"`. Prints the simplicity values for both the model basis vectors and the vectors in the simplicity basis of the nearly null space, the percent of variance explained by each basis vector, and the cummulative percent of variance explained up to each basis vector, relative to the total variance explained by the subspace (model or nearly null).

Usage

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

Arguments

`x`	an object of class `"simpart"`, typically result of `simpart()`.
`...`	arguments to be passed to or from other methods.

See Also

`simpart`

Examples

```
library(prinsimp)

## Caterpillar data: estimated covariance from Kingsolver et al (2004)
## Measurements are at temperatures 11, 17, 23, 29, 35, 40
data(caterpillar)

cat.sim <- simpart(caterpillar, simpledim = 2,
                   x = c(11, 17, 23, 29, 35, 40), cov = TRUE)
print(cat.sim)
```

`simpart` *Simple Partition*

Description

`simpart` partitions a d-dimensional sample space into two orthonormal subspaces: a `simpledim`-dimensional nearly null space and a $(d - \texttt{simpledim})$-dimensional model space. It provides an orthonormal basis for each subspace. The nearly null space basis is defined in terms of a simplicity measure and is ordered from most simple to least simple. The model space basis is made up of leading eigenvectors of the covariance matrix and is ordered by proportion of variance explained.

Returns the result as an object of class `simpart`.

Usage

```
simpart(y, simpledim, ...)

## S3 method for class 'formula'
simpart(formula, simpledim, data = NULL, ...)

## Default S3 method:
simpart(y, simpledim, measure = c('first', 'second', 'periodic'),
        x = seq(d), cov=FALSE, reverse=rep(FALSE, d), na.action, ...)
```

Arguments

`formula`	a formula with no response variable, referring only to numeric variables.
`y`	a matrix or data frame that specifies the data, or a covariance matrix. Data matrix has d columns, covariance matrix is $d \times d$.
`simpledim`	the dimension of the nearly null space of the covariance matrix. It is equal to d minus the dimension of the model space.
`measure`	a function that calculates a simplicity measure of a vector, based on a non-negative definite symmetric matrix Lambda. There are three built in simplicity measures, specified by 'first', 'second', or 'periodic' that correspond to first divided difference, second divided difference and periodic simplicity respectively. The argument `measure` can take a user specified function.

`data`	an optional data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
`x`	a vector of independent variable values (for functional data), length equal to d, the number of columns of `y`. If not supplied, a sequence from 1 to d is used.
`cov`	a logical value. If true, then `y` is assumed to be a $d \times d$ covariance matrix. If false, y is assumed to be an $n \times d$ data matrix which `simpart` uses to calculate a $d \times d$ covariance matrix.
`reverse`	a logical vector of length d. If the i-th element is true, the i-th basis vector is "reversed" by multiplication by -1. Basis vectors are arranged with model basis first, then simplicity basis. If length of `reverse` is less than d, then the remaining entries of `reverse` are assumed to be false, and the corresponding basis vectors remain unchanged.
`na.action`	specify how missing data should be treated.
`...`	arguments passed to or from other methods. If `x` is a formula one might specify `cov` or `reverse`. If `"periodic"` is chosen as the `measure`, period is specified as a numeric. If `measure` is user specified, its arguments are passed here.

Details

`simpart` is a generic function with `"formula"` and `"default"` methods.

`simpart` implements a method described in Gaydos et al (2013).

When `cov=FALSE`, the covariance matrix is calculated using the data matrix `y`. The calculation uses divisor n, the number of rows of `y`.

Value

`simpart` returns a list with class `"simpart"` containing the following components:

`model`	a $d \times (d - \texttt{simpledim})$ matrix with columns containing the basis of the model space, that is, containing the first $(d - \texttt{simpledim})$ eigenvectors of the covariance matrix. Basis vectors are arranged in descending order of eigenvalue, that is, in descending order of the proportion of variance explained.
`simple`	$d \times \texttt{simpledim}$ matrix with columns containing the simplicity basis of the nearly null space. Basis vectors are arranged in descending order of simplicity.
`variance`	list of three components: **model** variances associated with the vectors in the model basis. **simple** variances associated with the vectors in the simplicity basis of the nearly null space. **full** variances associated with eigenvectors of the covariance matrix, that is, its eigenvalues.
`simplicity`	list of three components: **model** simplicity values of the vectors in the model basis. **simple** eigenvalues of the vectors in the simplicity basis of the nearly null space. **full** simplicity values of the simplicity basis when `simpledim=d`.

`call`	the matched call
`measure`	the simplicity measure used: `"first"`, `"second"`, `"periodic"` or an user specified measure function
`varperc`	the percent of variance explained by the corresponding basis vector, as a list of two components: **model** percent of variance explained by the vectors in the model basis. **simple** percent of variance explained by the vectors in the simplicity basis of the nearly null space.
`scores`	if `y` is the data matrix, the scores on the basis vector loadings.

Note

The simplicity values of the simplicity basis when `simpledim=d` are equal to the eigenvalues of the non-negative definite matrix, Lambda, that defines the simplicity measure.

References

T.L. Gaydos, N.E. Heckman, M. Kirkpatrick, J.R. Stinchcombe, J. Schmitt, J. Kingsolver, J.S. Marron. (2013). Visualizing genetic constraints. *Annals of Applied Statistics* 7: 860-882.

See Also

`summary.simpart`, `plot.simpart`

Examples

```
library(prinsimp)
require(graphics)

## Caterpillar data: estimated covariance from Kingsolver et al (2004)
## Measurements are at temperatures 11, 17, 23, 29, 35, 40
data(caterpillar)

## Analyze 5 dimensional model space, 1 dimensional nearly null space
## First divided difference simplicity measure
simpart(caterpillar, simpledim=1, cov=TRUE)  # Need to specify x

simpart(caterpillar, simpledim=1,
        x=c(11, 17, 23, 29, 35, 40), cov=TRUE)

## Second divided difference simplicity measure and 3-dimensional model space
simpart(caterpillar, simpledim=3, measure="second",
        x=c(11, 17, 23, 29, 35, 40), cov=TRUE)
```

summary.simpart *Summary method for Simple Partition*

Description

The summary method for class "simpart". Prints the dimensions of the nearly null space in the simpart object, the percent of total variance explained by each basis vector, the cumulative percent of total variance explained (with accumulation restarting in the nearly null space), and the simplicity values of each basis vector. The model basis vectors are ordered by the percent of variance explained in descending order. The vectors in the simplicity basis of the nearly null space are ordered by their simplicity measures in descending order.

Usage

```
## S3 method for class 'simpart'
summary(object, loadings = FALSE, ...)

## S3 method for class 'summary.simpart'
print(x, digits = 3, loadings = x$print.loadings,
       ...)
```

Arguments

object	an object of class "simpart", as from simpart().
loadings	logical. If true, all basis vectors are printed.
x	an object of class "summary.simpart".
digits	the number of significant digits to be used in listing loadings.
...	arguments to be passed to or from other methods.

Value

object with additional component print.loadings.

See Also

simpart

Examples

```
library(prinsimp)

## Caterpillar data: estimated covariance from Kingsolver et al (2004)
## Measurements are at temperatures 11, 17, 23, 29, 35, 40
data(caterpillar)

cat.sim <- simpart(caterpillar, simpledim = 2,
                   x = c(11, 17, 23, 29, 35, 40), cov = TRUE)
```

```
summary(cat.sim)

print(summary(cat.sim, loadings = TRUE), digits = 2)
```

varperc *Varperc method for Simple Partition*

Description

Produces the percent of variance explained plot for an object of the class `simpart`

Usage

```
varperc(x, ...)
```

Arguments

`x`	an object of class `"simpart"`, typically result of `simpart`.
`...`	other parameters passed through to plotting functions

See Also

`basisplot`, `varsimp`, `simpart`

Examples

```
library(prinsimp)
require(graphics)

## Caterpillar estimated covariance from Kingsolver et al (2004)
## Measurements are at times 11, 17, 23, 29, 35, 40
data(caterpillar)

cat.sim <- simpart(caterpillar, simpledim = 2,
                   x = c(11, 17, 23, 29, 35, 40), cov = TRUE)

varperc(cat.sim)
```

varsimp — *Varsimp method for Simple Partition*

Description

Produces the variance-simplicity view for the specified basis vectors of the d-dimensional sample space.

Usage

```
varsimp(x, display = list(model=TRUE, simple=TRUE), full.simple = TRUE, ...)
```

Arguments

`x`	an object of class `"simpart"`, typically result of `simpart`.
`display`	a list specifying which model basis vectors and which nearly null space simplicity basis vectors to print.
`full.simple`	logical indicating whether to draw the background lines at simplicity values of the full space.
`...`	other parameters passed through to plotting functions

See Also

`basisplot`, `varperc`, `simpart`

Examples

```
library(prinsimp)
require(graphics)

## Caterpillar estimated covariance from Kingsolver et al (2004)
## Measurements are at times 11, 17, 23, 29, 35, 40
data(caterpillar)

cat.sim <- simpart(caterpillar, simpledim = 2,
                   x = c(11, 17, 23, 29, 35, 40), cov = TRUE)

## Display all 6 basis vectors in the variance-simplicity view
varsimp(cat.sim)

## Display only the 4 model basis vectors in the variance-simplicity view
varsimp(cat.sim, display = list(model=1:4))
```

Index