Package ‘whitening’

March 6, 2018

Version 1.0.0

Date 2018-03-05

Title Whitening and High-Dimensional Canonical Correlation Analysis

Author Korbinian Strimmer, Takoua Jendoubi, Agnan Kessy, Alex Lewin

Maintainer Korbinian Strimmer <strimmerlab@gmail.com>

Depends R (>= 3.0.2), corpcor (>= 1.6.9)

Imports stats

Suggests

Description Implements the whitening methods (ZCA, PCA, Cholesky, ZCA-cor, and PCA-cor) discussed in Kessy, Lewin, and Strimmer (2018) ``Optimal whitening and decorrelation'', The American Statistician, <doi:10.1080/00031305.2016.1277159>, as well as the whitening approach to Canonical Correlation Analysis allowing negative canonical correlations described in Jendoubi and Strimmer (2018) ``Probabilistic canonical correlation analysis: a whitening approach'', <arXiv:1802.03490>.

License GPL (>= 3)

URL http://strimmerlab.org/software/whitening/

NeedsCompilation no

Repository CRAN

Date/Publication 2018-03-06 18:51:10 UTC

R topics documented:

whitening-package — *The whitening Package*

Description

The "whitening" package implements the whitening methods (ZCA, PCA, Cholesky, ZCA-cor, and PCA-cor) discussed in Kessy, Lewin, and Strimmer (2018) as well as the whitening approach to Canonical Correlation Analysis allowing negative canonical correlations described in Jendoubi and Strimmer (2018).

Author(s)

Korbinian Strimmer (`http://strimmerlab.org/`) with Takoua Jendoubi, Agnan Kessy, and Alex Lewin.

References

Kessy, A., A. Lewin, and K. Strimmer. 2018. Optimal whitening and decorrelation. The American Statistician. `https://doi.org/10.1080/00031305.2016.1277159`

Jendoubi, T., and K. Strimmer 2018. Probabilistic canonical correlation analysis: a whitening approach. `https://arxiv.org/abs/1802.03490`

Website: `http://strimmerlab.org/software/whitening/`

See Also

`whiteningMatrix`. `whiten`, `cca`, and `scca`.

scca — *Estimate Canonical Correlations and Directions (Shrinkage and Empirical Estimates)*

Description

scca computes canonical correlations and directions using a shrinkage estimate of the joint correlation matrix of X and Y.

cca computes canonical correlations and directions based on empirical correlations.

Usage

```
scca(X, Y, lambda.cor, scale=TRUE, verbose=TRUE)
cca(X, Y, scale=TRUE)
```

Arguments

`X`	First data matrix, with samples in rows and variables in columns.
`Y`	Second data matrix, with samples in rows and variables in columns.
`lambda.cor`	Shrinkage intensity for estimating the joint correlation matrix - see `cor.shrink`. If not specified this will be estimated from the data.
`scale`	Determines whether canonical directions are computed for standardized or raw data. Note that if data are not standardized the canonical directions contain the scale of the variables.
`verbose`	Report shrinkage intensities-

Details

The canonical directions in this function are scaled in such a way that they correspond to whitening matrices - see Jendoubi and Strimmer (2018) for details. Note that the sign convention for the canonical directions employed here allows purposely for both positive and negative canonical correlations.

The function `scca` uses some clever matrix algebra to avoid computation of full correlation matrices, and hence can be applied to high-dimensional data sets - see Jendoubi and Strimmer (2018) for details.

`cca` it is a shortcut for running `scca` with `lambda.cor=0` and `verbose=FALSE`.

If `scale=FALSE` the standard deviations needed for the canonical directions are estimated by `apply(X, 2, sd)` and `apply(X, 2, sd)`.

If X or Y contains only a single variable the correlation-adjusted cross-correlations K reduce to the CAR score (see `carscore`) described in Strimmer and Zuber (2011).

Value

`scca` and `cca` return a list with the following components:

`K` - the correlation-adjusted cross-correlations.

`lambda` - the canonical correlations.

`WX` - the whitening matrix for X, with canonical directions in the rows.

`WY` - the whitening matrix for Y, with canonical directions in the rows.

`scale` - whether data was standardized (if codescale=FALSE then canonical directions include scale of the data).

`scca` additionally returns the shrinkage intensity in the variable `lambda.cor` where `lambda.cor.estimated` indicates whether it was specified or estimated.

Author(s)

Korbinian Strimmer (`http://strimmerlab.org`) with Takoua Jendoubi.

References

Jendoubi, T., and K. Strimmer 2018. Probabilistic canonical correlation analysis: a whitening approach. https://arxiv.org/abs/1802.03490

Zuber, V., and K. Strimmer. 2011. High-dimensional regression and variable selection using CAR scores. Statist. Appl. Genet. Mol. Biol. 10: 34. https://doi.org/10.2202/1544-6115.1730

See Also

cancor and carscore.

Examples

```
# load whitening library
library("whitening")


# example data set
data(LifeCycleSavings)
X = as.matrix( LifeCycleSavings[, 2:3] )
Y = as.matrix( LifeCycleSavings[, -(2:3)] )
n = nrow(X)
colnames(X) # "pop15" "pop75"
colnames(Y) # "sr"   "dpi"   "ddpi"

# CCA

cca.out = cca(X, Y, scale=TRUE)
cca.out$lambda  # canonical correlations
cca.out$WX      # whitening matrix / canonical directions X
cca.out$WY      # whitening matrix / canonical directions Y
cca.out$K       # correlation-adjusted cross-correlations

# CCA whitened data
CCAX = scale(X)
CCAY = scale(Y)
zapsmall(cov(CCAX))
zapsmall(cov(CCAY))
zapsmall(cov(CCAX,CCAY)) # canonical correlations


# compare with built-in function cancor
# note different signs in correlations and directions!
cancor.out = cancor(scale(X), scale(Y))
cancor.out$cor                  # canonical correlations
t(cancor.out$xcoef)*sqrt(n-1)   # canonical directions X
t(cancor.out$ycoef)*sqrt(n-1)   # canonical directions Y


## see "User guides, package vignettes and other documentation"
## for examples with high-dimensional data using the scca function
```

whiteningMatrix — *Compute Whitening Matrix and Whiten Data*

Description

whiteningMatrix computes the whitening matrix W corresponding to the five natural whitening procedures discussed in Kessy, Lewin, and Strimmer (2018).

whiten whitens data X using the empirical covariance matrix $cov(X)$ as basis for computing the whitening transformation.

Usage

```
whiteningMatrix(Sigma, method=c("ZCA", "PCA", "Cholesky",
                                 "ZCA-cor", "PCA-cor"))
whiten(X, method=c("ZCA", "PCA", "Cholesky", "ZCA-cor", "PCA-cor"))
```

Arguments

Sigma	Covariance matrix.
method	Determines the type of whitening.
X	Data matrix, with samples in rows and variables in columns.

Details

ZCA whitening, or Mahalanobis whitening ensures that the average covariance between whitened and orginal variables is maximal. Likewise, ZCA-cor whitening leads to whitened variables that are maximally correlated (on average) with the original variables.

In contrast, PCA and PCA-cor whitening lead to maximally compressed whitened variables, as measured by squared covariance and correlation, respectively.

Cholesky whitening is the unique whitening procedure that results from lower-triangular positive diagonal cross-covariance and cross-correlations matrices.

In PCA and PCA-cor eigenvector matrices with positive diagonal are used, in order to resolve the sign-ambiguity and also to make cross-correlations and cross-correlations positive diagonal.

For details see Kessy, Lewin, and Strimmer (2018).

ZCA-cor whitening is implicitely employed in computing CAT and CAR scores (cf. catscore and carscore).

Canonical correlation analysis (CCA) can also be understood as a special form form of whitening.

Value

whiteningMatrix returns a square whitening matrix W.

whiten returns the whitened data matrix $Z = XW'$.

Author(s)

Korbinian Strimmer (http://strimmerlab.org) with Agnan Kessy and Alex Lewin.

References

Kessy, A., A. Lewin, and K. Strimmer. 2018. Optimal whitening and decorrelation. The American Statistician. https://doi.org/10.1080/00031305.2016.1277159

See Also

catscore and carscore.

Examples

```
# load whitening library
library("whitening")

######

# example data set
# E. Anderson. 1935.  The irises of the Gaspe Peninsula.
# Bull. Am. Iris Soc. 59: 2--5
data("iris")
X = as.matrix(iris[,1:4])
d = ncol(X) # 4
n = nrow(X) # 150
colnames(X) # "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"

# estimate covariance
S = cov(X)

# ZCA-cor whitening matrix
W.ZCAcor = whiteningMatrix(S, method="ZCA-cor")

# whitened data
Z.ZCAcor.1 = tcrossprod(X, W.ZCAcor)
zapsmall( cov(Z.ZCAcor.1) )

# directly compute whitened data from X
Z.ZCAcor.2 = whiten(X, method="ZCA-cor")
zapsmall( cov(Z.ZCAcor.2) )
```

Index