Package 'RGCCA'

May 11, 2017

Type Package

Title Regularized and Sparse Generalized Canonical Correlation Analysis for Multiblock Data

Version 2.1.2

Date 2017-04-26

Author Arthur Tenenhaus and Vincent Guillemot

Maintainer Arthur Tenenhaus <arthur.tenenhaus@centralesupelec.fr>

Description Multiblock data analysis concerns the analysis of several sets of variables (blocks) observed on the same group of individuals. The main aims of the RGCCA package are: (i) to study the relationships between blocks and (ii) to identify subsets of variables of each block which are active in their relationships with the other blocks.

License GPL (>= 2)

Suggests knitr, rmarkdown, ggplot2

VignetteBuilder knitr

Imports MASS, Deriv

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-05-11 06:06:45 UTC

R topics documented:

`cov2` *Variance and Covariance (Matrices)*

Description

cov2() is similar to cov() but has an additional argument. The denominator n (bias = TRUE) can be used (instead of $n-1$) to give a biased estimator of the (co)variance.

Usage

```
cov2(x, y = NULL, bias = TRUE)
```

Arguments

`x`	A numeric vector, matrix or data.frame.
`y`	A numeric vector, matrix or data.frame.
`bias`	A logical value. If bias = TRUE, n is used to give a biased estimator of the (co)variance. If bias = FALSE, $n-1$ is used (default: TRUE).

Value

`C`	Estimation of the variance (resp. covariance) of x (resp. x and y).

`defl.select` *deflation function*

Description

The function defl.select() computes residual matrices $X_{1,h+1}, \ldots, X_{J,h+1}$. These residual matrices are determined according to the following formula: $X_{j,h+1} = X_{jh} - y_{jh} p_{jh}^t$.

Usage

```
defl.select(yy, rr, nncomp, nn, nbloc)
```

Arguments

`yy`	A matrix that contains the SGCCA block components of each block: $y_{1h}, \ldots, y_{Jh}$
`rr`	A list that contains the residual matrices $X_{1h}, \ldots, X_{Jh}$
`nncomp`	A $1 \times J$ vector that contains the number of components to compute for each block.
`nn`	A $1 \times J$ vector that contains the numbers of already computed components for each block
`nbloc`	Number of blocks.

Value

resdefl	A list of J elements that contains $X_{1,h+1}, \ldots, X_{J,h+1}$.
pdefl	A list of J elements that contains $p_{1h}, \ldots, p_{Jh}$.

---

miscrossprod *Cross product function for inputs with missing data.*

Description

Given vectors x and y as arguments, the function miscrossprod() returns the cross-product $x^t y$. miscrossprod() handles missing data.

Usage

```
miscrossprod(x, y)
```

Arguments

x	A numeric vector.
y	A numeric vector.

Value

d.p	The dot product between x and y: $x^t y$

---

rgcca *Regularized Generalized Canonical Correlation Analysis (RGCCA)*

Description

Regularized Generalized Canonical Correlation Analysis (RGCCA) is a generalization of regularized canonical correlation analysis to three or more sets of variables. Given J matrices $X_1, X_2, ..., X_J$ that represent J sets of variables observed on the same set of n individuals. The matrices $X_1, X_2, ..., X_J$ must have the same number of rows, but may (and usually will) have different numbers of columns. The aim of RGCCA is to study the relationships between these J blocks of variables. It constitutes a general framework for many multi-block data analysis methods. It combines the power of multi-block data analysis methods (maximization of well identified criteria) and the flexibility of PLS path modeling (the researcher decides which blocks are connected and which are not). Hence, the use of RGCCA requires the construction (user specified) of a design matrix C, that characterize the connections between blocks. Elements of the symmetric design matrix $C = (c_{jk})$ is equal to 1 if block j and block k are connected, and 0 otherwise. The function rgcca() implements a monotonically convergent algorithm (i.e. the bounded criteria to be maximized increases at each step of the iterative procedure) that is very similar to the PLS algorithm proposed by Herman Wold and finds at convergence a stationnary point of the RGCCA optimization problem. . Moreover, depending

on the dimensionality of each block X_j, $j = 1, ..., J$, the primal (when $n > p_j$) algorithm or the dual (when $n < p_j$) algorithm is used (see Tenenhaus et al. 2015). Moreover, by deflation strategy, rgcca() allow to compute several RGCCA block components (specified by ncomp) for each block. Within each block, block components are guaranteed to be orthogonal using the deflation procedure. The so-called symmetric deflation is considered in this implementation, i.e. each block is deflated with respect to its own component(s). It should be noted that the numbers of components per block can differ from one block to another.

Usage

```
rgcca(A, C = 1 - diag(length(A)), tau = rep(1, length(A)), ncomp = rep(1,
  length(A)), scheme = "centroid", scale = TRUE, init = "svd",
  bias = TRUE, tol = 1e-08, verbose = TRUE)
```

Arguments

A	A list that contains the J blocks of variables $X_1, X_2, ..., X_J$.
C	A design matrix that describes the relationships between blocks (default: complete design).
tau	tau is either a $1*J$ vector or a $max(ncomp)*J$ matrix, and contains the values of the shrinkage parameters (default: tau = 1, for each block and each dimension). If tau = "optimal" the shrinkage paramaters are estimated for each block and each dimension using the Schafer and Strimmer (2005) analytical formula . If tau is a $1 * J$ numeric vector, tau[j] is identical across the dimensions of block X_j. If tau is a matrix, tau[k, j] is associated with X_{jk} (kth residual matrix for block j)
ncomp	A $1 * J$ vector that contains the numbers of components for each block (default: rep(1, length(A)), which gives one component per block.)
scheme	The value is "horst", "factorial", "centroid" or any diffentiable convex scheme function g designed by the user (default: "centroid").
scale	If scale = TRUE, each block is standardized to zero means and unit variances and then divided by the square root of its number of variables (default: TRUE).
init	The mode of initialization to use in RGCCA algorithm. The alternatives are either by Singular Value Decompostion ("svd") or random ("random") (Default: "svd").
bias	A logical value for biaised or unbiaised estimator of the var/cov (default: bias = TRUE).
tol	The stopping value for convergence.
verbose	If verbose = TRUE, the progress will be report while computing (default: TRUE).

Value

Y	A list of J elements. Each element of Y is a matrix that contains the RGCCA components for the corresponding block.
a	A list of J elements. Each element of a is a matrix that contains the outer weight vectors for each block.

astar	A list of J elements. Each element of astar is a matrix defined as Y[[j]][, h] = A[[j]]%*%astar[[j]][, h].
C	A design matrix that describes the relation between blocks (user specified).
tau	A vector or matrix that contains the values of the shrinkage parameters applied to each block and each dimension (user specified).
scheme	The scheme chosen by the user (user specified).
ncomp	A $1 * J$ vector that contains the numbers of components for each block (user specified).
crit	A vector that contains the values of the criteria across iterations.
primal_dual	A $1 * J$ vector that contains the formulation ("primal" or "dual") applied to each of the J blocks within the RGCCA alogrithm
AVE	indicators of model quality based on the Average Variance Explained (AVE): AVE(for one block), AVE(outer model), AVE(inner model).

References

Tenenhaus M., Tenenhaus A. and Groenen PJF (2017), Regularized generalized canonical correlation analysis: A framework for sequential multiblock component methods, Psychometrika, in press

Tenenhaus A., Philippe C., & Frouin V. (2015). Kernel Generalized Canonical Correlation Analysis. Computational Statistics and Data Analysis, 90, 114-131.

Tenenhaus A. and Tenenhaus M., (2011), Regularized Generalized Canonical Correlation Analysis, Psychometrika, Vol. 76, Nr 2, pp 257-284.

Schafer J. and Strimmer K., (2005), A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statist. Appl. Genet. Mol. Biol. 4:32.

Examples

```
#############
# Example 1 #
#############
data(Russett)
X_agric =as.matrix(Russett[,c("gini","farm","rent")])
X_ind = as.matrix(Russett[,c("gnpr","labo")])
X_polit = as.matrix(Russett[ , c("demostab", "dictator")])
A = list(X_agric, X_ind, X_polit)
#Define the design matrix (output = C)
C = matrix(c(0, 0, 1, 0, 0, 1, 1, 1, 0), 3, 3)
result.rgcca = rgcca(A, C, tau = c(1, 1, 1), scheme = "factorial", scale = TRUE)
lab = as.vector(apply(Russett[, 9:11], 1, which.max))
plot(result.rgcca$Y[[1]], result.rgcca$Y[[2]], col = "white",
     xlab = "Y1 (Agric. inequality)", ylab = "Y2 (Industrial Development)")
text(result.rgcca$Y[[1]], result.rgcca$Y[[2]], rownames(Russett), col = lab, cex = .7)

#############
# Example 2 #
#############
data(Russett)
X_agric =as.matrix(Russett[,c("gini","farm","rent")])
```

```
X_ind = as.matrix(Russett[,c("gnpr","labo")])
X_polit = as.matrix(Russett[ , c("inst", "ecks", "death",
                                 "demostab", "dictator")])
A = list(X_agric, X_ind, X_polit, cbind(X_agric, X_ind, X_polit))

#Define the design matrix (output = C)
C = matrix(c(0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0), 4, 4)
result.rgcca = rgcca(A, C, tau = c(1, 1, 1, 0), ncomp = rep(2, 4),
                     scheme = function(x) x^4, scale = TRUE) # HPCA
lab = as.vector(apply(Russett[, 9:11], 1, which.max))
plot(result.rgcca$Y[[4]][, 1], result.rgcca$Y[[4]][, 2], col = "white",
     xlab = "Global Component 1", ylab = "Global Component 2")
text(result.rgcca$Y[[4]][, 1], result.rgcca$Y[[4]][, 2], rownames(Russett),
     col = lab, cex = .7)

## Not run:
#####################################
# example 3: RGCCA and leave one out #
#####################################
Ytest = matrix(0, 47, 3)
X_agric =as.matrix(Russett[,c("gini","farm","rent")])
X_ind = as.matrix(Russett[,c("gnpr","labo")])
X_polit = as.matrix(Russett[ , c("demostab", "dictator")])
A = list(X_agric, X_ind, X_polit)
#Define the design matrix (output = C)
C = matrix(c(0, 0, 1, 0, 0, 1, 1, 1, 0), 3, 3)
result.rgcca = rgcca(A, C, tau = rep(1, 3), ncomp = rep(1, 3),
                     scheme = "factorial", verbose = TRUE)

for (i in 1:nrow(Russett)){
 B = lapply(A, function(x) x[-i, ])
 B = lapply(B, scale2)
 resB = rgcca(B, C, tau = rep(1, 3), scheme = "factorial", scale = FALSE, verbose = FALSE)
 #  look for potential conflicting sign among components within the loo loop.
 for (k in 1:length(B)){
   if (cor(result.rgcca$a[[k]], resB$a[[k]]) >= 0)
     resB$a[[k]] = resB$a[[k]] else resB$a[[k]] = -resB$a[[k]]
 }
 Btest =lapply(A, function(x) x[i, ])
 Btest[[1]]=(Btest[[1]]-attr(B[[1]],"scaled:center")) /
                 (attr(B[[1]],"scaled:scale"))/sqrt(NCOL(B[[1]]))
 Btest[[2]]=(Btest[[2]]-attr(B[[2]],"scaled:center")) /
                 (attr(B[[2]],"scaled:scale"))/sqrt(NCOL(B[[2]]))
 Btest[[3]]=(Btest[[3]]-attr(B[[3]],"scaled:center")) /
                 (attr(B[[3]],"scaled:scale"))/sqrt(NCOL(B[[3]]))
 Ytest[i, 1] = Btest[[1]]%*%resB$a[[1]]
 Ytest[i, 2] = Btest[[2]]%*%resB$a[[2]]
 Ytest[i, 3] = Btest[[3]]%*%resB$a[[3]]
}
lab = apply(Russett[, 9:11], 1, which.max)
plot(result.rgcca$Y[[1]], result.rgcca$Y[[2]], col = "white",
     xlab = "Y1 (Agric. inequality)", ylab = "Y2 (Ind. Development)")
text(result.rgcca$Y[[1]], result.rgcca$Y[[2]], rownames(Russett),
```

```
     col = lab, cex = .7)
text(Ytest[, 1], Ytest[, 2], substr(rownames(Russett), 1, 1),
     col = lab, cex = .7)

## End(Not run)
```

---

rgccak *Internal function for computing the RGCCA parameters (RGCCA block components, outer weight vectors, etc.).*

Description

The function rgccak() is called by rgcca() and does not have to be used by the user. The function rgccak() computes the RGCCA block components, outer weight vectors, etc., for each block and each dimension. Depending on the dimensionality of each block $X_j, j = 1, ..., J$, the primal (when $n > p_j$) or the dual (when $n < p_j$) algorithm is used (see Tenenhaus et al. 2015)

Usage

```
rgccak(A, C, tau = "optimal", scheme = "centroid", scale = FALSE,
  verbose = FALSE, init = "svd", bias = TRUE, tol = 1e-08)
```

Arguments

A	A list that contains the J blocks of variables. Either the blocks $(X_1, X_2, ..., X_J)$ or the residual matrices $(X_{h1}, X_{h2}, ..., X_{hJ})$.
C	A design matrix that describes the relationships between blocks. (Default: complete design).
tau	A $1 * J$ vector that contains the values of the shrinkage parameters τ_j, $j = 1, ..., J$. (Default: $\tau_j = 1$, $j = 1, ..., J$). If tau = "optimal" the shrinkage intensity paramaters are estimated using the Schafer and Strimmer (2005) analytical formula.
scheme	The value is "horst", "factorial", "centroid" or any diffentiable convex scheme function g designed by the user (default: "centroid").
scale	if scale = TRUE, each block is standardized to zero means and unit variances (default: TRUE).
verbose	Will report progress while computing if verbose = TRUE (default: TRUE).
init	The mode of initialization to use in the RGCCA algorithm. The alternatives are either by Singular Value Decompostion or random (default : "svd").
bias	A logical value for either a biaised or unbiaised estimator of the var/cov.
tol	Stopping value for convergence.

Value

Y	A $n * J$ matrix of RGCCA outer components
Z	A $n * J$ matrix of RGCCA inner components
a	A list of outer weight vectors
crit	The values of the objective function to be optimized in each iteration of the iterative procedure.
AVE	Indicators of model quality based on the Average Variance Explained (AVE): AVE(for one block), AVE(outer model), AVE(inner model).
C	A design matrix that describes the relationships between blocks (user specified).
tau	$1 * J$ vector containing the value for the tau penalties applied to each of the J blocks of data (user specified)
scheme	The scheme chosen by the user (user specified).

References

Tenenhaus M., Tenenhaus A. and Groenen PJF (2017), Regularized generalized canonical correlation analysis: A framework for sequential multiblock component methods, Psychometrika, in press

Tenenhaus A., Philippe C., & Frouin V. (2015). Kernel Generalized Canonical Correlation Analysis. Computational Statistics and Data Analysis, 90, 114-131.

Tenenhaus A. and Tenenhaus M., (2011), Regularized Generalized Canonical Correlation Analysis, Psychometrika, Vol. 76, Nr 2, pp 257-284.

Schafer J. and Strimmer K., (2005), A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statist. Appl. Genet. Mol. Biol. 4:32.

Russett *Russett data*

Description

The Russett data set (Russett, 1964) are studied in Gifi (1990). Three blocks of variables have been defined for 47 countries. The first block X1 = [GINI, FARM, RENT] is related to "Agricultural Inequality". The second block X2 = [GNPR, LABO] describes "Industrial Development". The third one X3 = [INST, ECKS, DEAT] measures "Political Instability". An additional variable DEMO describes the political regime: stable democracy, unstable democracy or dictatorship. Russett collected this data to study relationships between Agricultural Inequality, Industrial Development and Political Instability. Russett's hypotheses can be formulated as follows: It is difficult for a country to escape dictatorship when its agricultural inequality is above-average and its industrial development below-average.

Usage

```
data(Russett)
```

Format

A data frame with 47 observations on the following 11 numeric variables.

`gini` Inequality of land distribution

`farm` % farmers that own half of the land

`rent` % farmers that rent all their land

`gnpr` Gross national product per capita ($1955)

`labo` % of labor force employed in agriculture

`inst` Instability of executive (45-61)

`ecks` Number of violent internal war incidents (46-61)

`death` Number of people killed as a result of civic group violence (50-62)

`demostab` binary variable equal to 1 for stable democraty and 0 otherwise

`demoinst` binary variable equal to 1 for unstable democraty and 0 otherwise

`dictator` binary variable equal to 1 for dictatorship and 0 otherwise

References

Russett B.M. (1964), Inequality and Instability: The Relation of Land Tenure to Politics, World Politics 16:3, 442-454.

Gifi, A. (1990), Nonlinear multivariate analysis, Chichester: Wiley.

Examples

```
#Loading of the Russett dataset
data(Russett)
#Russett is partitioned into three blocks (X_agric, X_ind, X_polit)
X_agric =as.matrix(Russett[,c("gini","farm","rent")])
X_ind = as.matrix(Russett[,c("gnpr","labo")])
X_polit = as.matrix(Russett[ , c("inst", "ecks", "death", "demostab",
                                 "demoinst", "dictator")])
A = list(X_agric, X_ind, X_polit)
lapply(A, dim)
```

---

scale2 *Scaling and Centering of Matrix-like Objects*

Description

Standardization (to zero means and unit variances) of matrix-like objects.

Usage

```
scale2(A, center = TRUE, scale = TRUE, bias = TRUE)
```

Arguments

A	A numeric matrix.
center	A logical value. If center = TRUE, each column is translated to have zero mean (default: TRUE).
scale	A logical value. If scale = TRUE, each column is transformed to have unit variance (default = TRUE).
bias	Logical value for biaised $(1/n)$ or unbiaised $(1/(n-1))$ estimator of the var/cov (default = TRUE).

Value

A	The centered and/or scaled matrix. The centering and scaling values (if any) are returned as attributes "scaled:center" and "scaled:scale".

---

sgcca *Variable Selection For Generalized Canonical Correlation Analysis (SGCCA)*

Description

SGCCA extends RGCCA to address the issue of variable selection. Specifically, RGCCA is combined with an L1-penalty that gives rise to Sparse GCCA (SGCCA) which is implemented in the function sgcca(). Given J matrices $X_1, X_2, ..., X_J$, that represent J sets of variables observed on the same set of n individuals. The matrices $X_1, X_2, ..., X_J$ must have the same number of rows, but may (and usually will) have different numbers of columns. Blocks are not necessarily fully connected within the SGCCA framework. Hence the use of SGCCA requires the construction (user specified) of a design matrix (C) that characterizes the connections between blocks. Elements of the symmetric design matrix $C = (c_{jk})$ are equal to 1 if block j and block k are connected, and 0 otherwise. The SGCCA algorithm is very similar to the RGCCA algorithm and keeps the same monotone convergence properties (i.e. the bounded criteria to be maximized increases at each step of the iterative procedure and hits at convergence a stationary point). Moreover, using a deflation strategy, sgcca() enables the computation of several SGCCA block components (specified by ncomp) for each block. Block components for each block are guaranteed to be orthogonal when using this deflation strategy. The so-called symmetric deflation is considered in this implementation, i.e. each block is deflated with respect to its own component. Moreover, we stress that the numbers of components per block could differ from one block to another.

Usage

```
sgcca(A, C = 1 - diag(length(A)), c1 = rep(1, length(A)), ncomp = rep(1,
  length(A)), scheme = "centroid", scale = TRUE, init = "svd",
  bias = TRUE, tol = .Machine$double.eps, verbose = FALSE)
```

Arguments

A	A list that contains the J blocks of variables $X_1, X_2, ..., X_J$.
C	A design matrix that describes the relationships between blocks (default: complete design).
c1	Either a $1 * J$ vector or a $max(ncomp) * J$ matrix encoding the L1 constraints applied to the outer weight vectors. Elements of c1 vary between $1/sqrt(p_j)$ and 1 (larger values of c1 correspond to less penalization). If c1 is a vector, L1-penalties are the same for all the weights corresponding to the same block but different components: $$forallh, \|a_{j,h}\|_{L_1} \leq c_1[j]\sqrt{p_j},$$ with p_j the number of variables of X_j. If c1 is a matrix, each row h defines the constraints applied to the weights corresponding to components h: $$forallh, \|a_{j,h}\|_{L_1} \leq c_1[h,j]\sqrt{p_j}.$$
ncomp	A $1 * J$ vector that contains the numbers of components for each block (default: rep(1, length(A)), which means one component per block).
scheme	Either "horst", "factorial" or "centroid" (Default: "centroid").
scale	If scale = TRUE, each block is standardized to zero means and unit variances and then divided by the square root of its number of variables (default: TRUE).
init	Mode of initialization use in the SGCCA algorithm, either by Singular Value Decompostion ("svd") or random ("random") (default : "svd").
bias	A logical value for biaised or unbiaised estimator of the var/cov.
tol	Stopping value for convergence.
verbose	Will report progress while computing if verbose = TRUE (default: TRUE).

Value

Y	A list of J elements. Each element of Y is a matrix that contains the SGCCA components for each block.
a	A list of J elements. Each element of a is a matrix that contains the outer weight vectors for each block.
astar	A list of J elements. Each element of astar is a matrix defined as Y[[j]][, h] = A[[j]]%*%astar[[j]][, h]
C	A design matrix that describes the relationships between blocks (user specified).
scheme	The scheme chosen by the user (user specified).
c1	A vector or matrix that contains the value of c1 applied to each block $\mathbf{X}_j$, $j = 1, \ldots, J$ and each dimension (user specified).
ncomp	A $1 \times J$ vector that contains the number of components for each block (user specified).
crit	A vector that contains the values of the objective function at each iterations.
AVE	Indicators of model quality based on the Average Variance Explained (AVE): AVE(for one block), AVE(outer model), AVE(inner model).

References

Tenenhaus, A., Philippe, C., Guillemot, V., Le Cao, K. A., Grill, J., and Frouin, V. , "Variable selection for generalized canonical correlation analysis.," Biostatistics, vol. 15, no. 3, pp. 569-583, 2014.

Examples

```
#############
# Example 1 #
#############
## Not run:
# Download the dataset's package at http://biodev.cea.fr/sgcca/.
# --> gliomaData_0.4.tar.gz

require(gliomaData)
data(ge_cgh_locIGR)

A <- ge_cgh_locIGR$multiblocks
Loc <- factor(ge_cgh_locIGR$y) ; levels(Loc) <- colnames(ge_cgh_locIGR$multiblocks$y)
C <-  matrix(c(0, 0, 1, 0, 0, 1, 1, 1, 0), 3, 3)
tau = c(1, 1, 0)

# rgcca algorithm using the dual formulation for X1 and X2
# and the dual formulation for X3
A[[3]] = A[[3]][, -3]
result.rgcca = rgcca(A, C, tau, ncomp = c(2, 2, 1), scheme = "factorial", verbose = TRUE)
# sgcca algorithm
result.sgcca = sgcca(A, C, c1 = c(.071,.2, 1), ncomp = c(2, 2, 1),
                     scheme = "centroid", verbose = TRUE)

############################
# plot(y1, y2) for (RGCCA) #
############################
layout(t(1:2))
plot(result.rgcca$Y[[1]][, 1], result.rgcca$Y[[2]][, 1], col = "white", xlab = "Y1 (GE)",
     ylab = "Y2 (CGH)", main = "Factorial plan of RGCCA")
text(result.rgcca$Y[[1]][, 1], result.rgcca$Y[[2]][, 1], Loc, col = as.numeric(Loc), cex = .6)
plot(result.rgcca$Y[[1]][, 1], result.rgcca$Y[[1]][, 2], col = "white", xlab = "Y1 (GE)",
     ylab = "Y2 (GE)", main = "Factorial plan of RGCCA")
text(result.rgcca$Y[[1]][, 1], result.rgcca$Y[[1]][, 2], Loc, col = as.numeric(Loc), cex = .6)

############################
# plot(y1, y2) for (SGCCA) #
############################
layout(t(1:2))
plot(result.sgcca$Y[[1]][, 1], result.sgcca$Y[[2]][, 1], col = "white", xlab = "Y1 (GE)",
     ylab = "Y2 (CGH)", main = "Factorial plan of SGCCA")
text(result.sgcca$Y[[1]][, 1], result.sgcca$Y[[2]][, 1], Loc, col = as.numeric(Loc), cex = .6)

plot(result.sgcca$Y[[1]][, 1], result.sgcca$Y[[1]][, 2], col = "white", xlab = "Y1 (GE)",
     ylab = "Y2 (GE)", main = "Factorial plan of SGCCA")
```

```
text(result.sgcca$Y[[1]][, 1], result.sgcca$Y[[1]][, 2], Loc, col = as.numeric(Loc), cex = .6)

# sgcca algorithm with multiple components and different L1 penalties for each components
# (-> c1 is a matrix)
init = "random"
result.sgcca = sgcca(A, C, c1 = matrix(c(.071,.2, 1, 0.06, 0.15, 1), nrow = 2, byrow = TRUE),
                     ncomp = c(2, 2, 1), scheme = "factorial", scale = TRUE, bias = TRUE,
                      init = init, verbose = TRUE)
# number of non zero elements per dimension
apply(result.sgcca$a[[1]], 2, function(x) sum(x!=0))
     #(-> 145 non zero elements for a11 and 107 non zero elements for a12)
apply(result.sgcca$a[[2]], 2, function(x) sum(x!=0))
     #(-> 85 non zero elements for a21 and 52 non zero elements for a22)
init = "svd"
result.sgcca = sgcca(A, C, c1 = matrix(c(.071,.2, 1, 0.06, 0.15, 1), nrow = 2, byrow = TRUE),
                     ncomp = c(2, 2, 1), scheme = "factorial", scale = TRUE, bias = TRUE,
                      init = init, verbose = TRUE)
## End(Not run)
```

---

sgccak *Internal function for computing the SGCCA parameters (SGCCA block components, outer weight vectors etc.)*

Description

The function sgccak() is called by sgcca() and does not have to be used by the user. sgccak() enables the computation of SGCCA block components, outer weight vectors, etc., for each block and each dimension.

Usage

```
sgccak(A, C, c1 = rep(1, length(A)), scheme = "centroid", scale = FALSE,
  tol = .Machine$double.eps, init = "svd", bias = TRUE, verbose = TRUE)
```

Arguments

A	A list that contains the J blocks of variables from which block components are constructed. It could be eiher the original matrices $(X_1, X_2, ..., X_J)$ or the residual matrices $(X_{h1}, X_{h2}, ..., X_{hJ})$.
C	A design matrix that describes the relationships between blocks.
c1	A $1 * J$ vector that contains the value of c1 applied to each block. The L1 bound on a[[j]] is $$\|a_j\|_{\ell_1} \leq c_1[j]\sqrt{p_j}.$$ with p_j the number of variables of X_j and with c1[j] between 0 and 1 (larger L1 bound corresponds to less penalization).
scheme	Either "horst", "factorial" or "centroid" (default: centroid).

scale	If scale = TRUE, each block is standardized to zero means and unit variances (default: TRUE).
tol	Stopping value for convergence.
init	Mode of initialization of the SGCCA algorithm. Either by Singular Value Decompostion ("svd") or random ("random") (default: "svd").
bias	Logical value for biaised $(1/n)$ or unbiaised $(1/(n-1))$ estimator of the var/cov.
verbose	Reports progress while computing, if verbose = TRUE (default: TRUE).

Value

Y	A $n * J$ matrix of SGCCA block components.
a	A list of J elements. Each element contains the outer weight vector of each block.
crit	The values of the objective function at each iteration of the iterative procedure.
converg	Speed of convergence of the alogrithm to reach the tolerance.
AVE	Indicators of model quality based on the Average Variance Explained (AVE): AVE(for one block), AVE(outer model), AVE(inner model).
C	A design matrix that describes the relationships between blocks (user specified).
scheme	The scheme chosen by the user (user specified).

---

soft.threshold *The function soft.threshold() soft-thresholds a vector such that the L1-norm constraint is satisfied.*

Description

The function soft.threshold() soft-thresholds a vector such that the L1-norm constraint is satisfied.

Usage

```
soft.threshold(x, sumabs = 1)
```

Arguments

x	A numeric vector.
sumabs	A numeric constraint on x's L1 norm.

Value

Returns a vector resulting from the soft thresholding of x given sumabs

Examples

```
x <- rnorm(10)
soft.threshold(x,0.5)
```

`tau.estimate` *Optimal shrinkage intensity parameters.*

Description

Estimation of the optimal shrinkage parameters as described in [1,2] and implemented in a more general version within the SHIP package [2].

Usage

```
tau.estimate(x)
```

Arguments

`x`	Data set on which the covariance matrix is estimated.

Value

`tau`	Optimal shrinkage intensity parameter

References

[1] Schaefer J. and Strimmer K., 2005. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statist. Appl. Genet. Mol. Biol. 4:32.

[2] Jelizarow M., Guillemot V., Tenenhaus A., Strimmer K., Boulesteix A.-L., 2010. Over-optimism in bioinformatics: an illustration. Bioinformatics 26:1990-1998.

Index