Package ‘concor’

February 19, 2015

Version 1.0-0.1
Date 2009-02-25
Title Concordance
Author R. Lafosse <lafosse@lsp.ups-tlse.fr>
Maintainer S. Déjean <sdejean@lsp.ups-tlse.fr>
Depends R (>= 0.99)
Description The four functions svdcp (cp for column partitioned),
svdbip or svdbip2 (bip for bi-partitioned), and svdbips (s for
a simultaneous optimization of one set of r solutions),
correspond to a "SVD by blocks" notion, by supposing each block
depending on relative subspaces, rather than on two whole
spaces as usual SVD does. The other functions, based on this
notion, are relative to two column partitioned data matrices x
and y defining two sets of subsets xi and yj of variables and
amount to estimate a link between xi and yj for the pair (xi,
yj) relatively to the links associated to all the other pairs.

Encoding latin1
License GPL
Repository CRAN
Date/Publication 2012-10-29 08:58:27
NeedsCompilation no

R topics documented:

concor ........................................................................ 2
concorcano ................................................................. 3
concoreg .................................................................... 4
concorgm ................................................................. 6
concorgmcano .......................................................... 7
concorgmreg ............................................................. 8
concors ..................................................................... 9
concorscano .............................................................. 10
Relative links of several subsets of variables $Y_j$ with another set $X$. SUCCESSIVE SOLUTIONS

Usage

```concor\(x, y, py, r\)```

Arguments

- $x, y$ are $n \times p$ and $n \times q$ matrices of $p$ and $q$ centered columns
- $py$ is a row vector which contains the numbers $q_i$, $i=1,...,ky$, of the $ky$ subsets $y_i$ of $y$: $\text{sum}(q_i)=\text{sum}(py)=q$. $py$ is the partition vector of $y$
- $r$ is the wanted number of successive solutions

Details

The first solution calculates $1+kx$ normed vectors: the vector $u[:,1]$ of $R_p$ associated to the $ky$ vectors $v_i[:,1]$ of $R_{qi}$, by maximizing $\sum_i \text{cov}(x * u[:,k], y_i * v_i[:,k])^2$, with $1+ky$ norm constraints on the axes. A component $x*u[:,k]$ is associated to $ky$ partial components $y_i*v_i[:,k]$ and to a global component $y*V[:,k]$. $\text{cov}(x * u[:,k], y_i * v_i[:,k])^2 = \sum \text{cov}(x * u[:,k], y_i * v_i[:,k])^2$. $y*V[:,k]$ is a global component of the components $y_i*v_i[:,k]$.

The second solution is obtained from the same criterion, but after replacing each $y_i$ by $y_i - y_i*v_i[:,1]*v_i[:,1]'$. And so on for the successive solutions 1,2,...,$r$. The biggest number of solutions may be $r=\inf(n,p,qi)$, when the $x'*y_i$'s are supposed with full rank; then $r_{max}=\min(c(\text{min}(py),n,p))$. For a set of $r$ solutions, the matrix $u'X'YV$ is diagonal and the matrices $u'X'Y_{ij}$ are triangular (good partition of the link by the solutions). concor.m is the svdcp.m function applied to the matrix $x'y$.

Value

- $u$ is a $p \times r$ matrix of axes in $R_p$ relative to $x$; $u'*u = \text{Identity}$
- $v$ is a $q \times r$ matrix of $ky$ row blocks $v_i (qi \times r)$ of axes in $R_{qi}$ relative to $y_i$; $v'*v_i = \text{Identity}$
- $V$ is a $q \times r$ matrix of axes in $R_q$ relative to $y$; $V'*V = \text{Identity}$
- $\text{cov2}$ is a $ky \times r$ matrix; each column $k$ contains $ky$ squared covariances $\text{cov}(x * u[:,k], y_i * v_i[:,k])^2$, the partial measures of link
References

Examples
# To make some "GPA" : so, by posing the compromise X = Y,
# "procrustes" rotations to the "compromise X" then are :
# Yj*(vj*u').

x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
c<-concor(x,y,c(3,2,4),2)
((t(x%*%c$u[,1]))%*%y[,1:3]%*%c$v[1:3,1])/10)^2;co$cov2[1,1]
t(x%*%c$u)%*%y%*%c$v

concorcano

Canonical analysis of several sets with another set

Description
Relative proximities of several subsets of variables Yj with another set X. SUCCESSIVE SOLUTIONS

Usage
concorcano(x,y,py,r)

Arguments
x is a n x p matrix of p centered variables
y is a n x q matrix of q centered variables
py is a row vector which contains the numbers qi, i=1,...,ky, of the ky subsets yi of y : \( \sum_q q_i = \text{sum}(py) = q \). py is the partition vector of y
r is the wanted number of successive solutions

Details
The first solution calculates a standardized canonical component \( cx[,1] \) of x associated to ky standardized components \( cy[,1] \) of yi by maximizing \( \sum_i \rho(cx[,1],cy[,1])^2 \).

The second solution is obtained from the same criterion, with ky orthogonality constraints for having \( \rho(cy[,1],cy[,2])=0 \) (that implies \( \rho(cx[,1],cx[,2])=0 \)). For each of the 1+ky sets, the r canonical components are 2 by 2 zero correlated.

The ky matrices \( (cx)^*cy \) are triangular.
This function uses concor function.
Value

list with following components

cx is n x r matrix of the r canonical components of x
cy is n.ky x r matrix. The ky blocks cyi of the rows n*(i-1)+1 : n*i contain the r canonical components relative to Yi
rho2 is a ky x r matrix; each column k contains ky squared canonical correlations $\rho(cx[k], cyi[k])^2$

References

Hanafi & Lafosse (2001) Generalisation de la regression lineaire simple pour analyser la dependance de K ensembles de variables avec un K+1 eme. Revue de Statistique Appliquee vol.49, n.1

Examples

```r
x<-matrix(rnorm(50),10,5);y<-matrix(rnorm(90),10,9)
x<-scale(x);y<-scale(y)
c<-concorcano(x,y,c(3,2,4),2)
diag(t(c$cx))%*%c$cy[1:10,]/10)*2
c$rho2[1,]
```

concoreg Redundancy of sets yj by one set x

Description

Regression of several subsets of variables Yj by another set X. SUCCESSIVE SOLUTIONS

Usage

`concoreg(x,y,py,r)`

Arguments

x is a n x p matrix of p centered explanatory variables
y is a n x q matrix of q centered variables
py is a row vector which contains the numbers $q_i$, $i = 1, ..., ky$, of the ky subsets $y_i$ of $y$: $\sum q_i = \text{sum}(py) = q$. py is the partition vector of y
r is the wanted number of successive solutions
**Details**

The first solution calculates 1+ky normed vectors: the component \( cx[,1] \) in \( R^n \) associated to the \( ky \) vectors \( vi[,1]'s \) of \( R^q_i \), by maximizing \( varexp_1 = \sum_i \rho(cx[,1], y_i * v_i[,1])^2 \text{var}(y_i * v_i[,1]) \), with 1 + \( ky \) norm constraints. A explanatory component \( cx[k] \) is associated to \( ky \) partial explained components \( yi*vi[,k] \) and also to a global explained component \( y*V[,k] \).

\( \rho(cx[,k], y*V[,k])^2 \text{var}(y*V[,k]) = varexp_k \). The total explained variance by the first solution is maximal.

The second solution is obtained from the same criterion, but after replacing each \( yi \) by \( y_i - y_i * v_i[,1]*v_i[,1]' \). And so on for the successive solutions 1,2,...,\( r \). The biggest number of solutions may be \( r = \inf(n,p,q_i) \), when the matrices \( x'*yi \) are supposed with full rank. For a set of \( r \) solutions, the matrix \( (cx)'*y*V \) is diagonal: "on average", the explanatory component of one solution is only linked with the components explained by this explanatory, and is not linked with the explained components of the other solutions. The matrices \( (cx)'*y_j*v_j \) are triangular: the explanatory component of one solution is not linked with each of the partial components explained in the following solutions. The definition of the explanatory components depends on the partition vector \( py \) from the second solution.

This function is using concor function

**Value**

list with following components

- \( cx \) the \( n \times r \) matrix of the \( r \) explanatory components
- \( v \) is a \( q \times r \) matrix of \( ky \) row blocks \( v_i (q_i \times r) \) of axes in \( R^q_i \) relative to \( yi \); \( v_i'*v_i = \text{Id} \)
- \( V \) is a \( q \times r \) matrix of axes in \( R^q \) relative to \( y \); \( V'*V = \text{Id} \)
- \( varexp \) is a \( ky \times r \) matrix; each column \( k \) contains \( ky \) explained variances \( \rho(cx[,k], y_i * v_i[,k])^2 \text{var}(y_i * v_i[,k]) \)

**References**


Chessel D. & Hanafi M. (1996) Analyses de la Co-inertie de K nuages de points. Revue de Statistique Appliquee vol.44, n.2. (this ACOM analysis of one multiset is obtained by the command : concoreg(Y,Y,py,r))

**Examples**

```r
x<-matrix(runif(50),10,5); y<-matrix(runif(90),10,9)
x<-scale(x); y<-scale(y)
cx<-concoreg(x,y,c(3,2,4),2)
(t(co$cx[,1])**%$y[,1:3]**%co$v[1:3,1])/10)^2; co$varexp[1,1]
(t(co$cx)**%$co$cx /10
diag(t(co$cx)**%$y%$co$v/10)^2
sum(co$varexp[,1]);sum(co$varexp[,2])
```
Analyzing a set of partial links between Xi and Yj, SUCCESSIVE SOLUTIONS

Usage

concorgm(x, px, y, py, r)

Arguments

- x is a n x p matrix of p centered variables
- y is a n x q matrix of q centered variables
- px is a row vector which contains the numbers pi, i=1,...,kx, of the kx subsets xi of x : sum(pi)=sum(px)=p. px is the partition vector of x
- py is the partition vector of y with ky subsets yj, j=1,...,ky
- r is the wanted number of successive solutions rmax <= min(min(px),min(py),n)

Details

For the first solution, ∑i ∑j cov2(xi * ui[1,1], yj * vj[1,1]) is the optimized criterion. The second solution is calculated from the same criterion, but with xi - xi * ui[1,1] * ui[1,1]' and yj - yj * vj[1,1] * vj[1,1]' instead of the kx+ky matrices xi and yj. And so on for the other solutions. When kx=1 (px=p), take concor.m

This function uses the svdbip function.

Value

list with following components

- u is a p x r matrix of kx row blocks ui (pi x r), the orthonormed partial axes of xi; associated partial components: xi*ui
- v is a q x r matrix of ky row blocks vj (qj x r), the orthonormed partial axes of yj; associated partial components: yj*vj
- cov2 is a kx x ky x r array; for r fixed to k, the matrix contains kxky squared covariances cov2(xi * ui[1,k], yj * vj[1,k])2, the partial links between xi and yj measured with the solution k.

References

Examples

```r
x <- matrix(runif(50), 10, 5); y <- matrix(runif(90), 10, 9)
x <- scale(x); y <- scale(y)
cg <- concorgm(x, c(2, 3), y, c(3, 2, 4), 2)
diag(t(x[, 1:2]) %*% cg$u[1:2, ]) %*% y[, 1:3] %*% cg$v[1:3, ]/10)^2
cg$cov2[1, 1, ]
```

---

**concorgmcano**

**Canonical analysis of subsets \( Y_j \) with subsets \( X_i \)**

**Description**

Canonical analysis of subsets \( Y_j \) with subsets \( X_i \). Relative valuations by squared correlations of the proximities of subsets \( X_i \) with subsets \( Y_j \). SUCCESSIVE SOLUTIONS

**Usage**

`concorgmcano(x, px, y, py, r)`

**Arguments**

- **x**: is a \( n \times p \) matrix of \( p \) centered variables
- **y**: is a \( n \times q \) matrix of \( q \) centered variables
- **px**: is a row vector which contains the numbers \( p_i \), \( i=1,...,k_x \), of the \( k_x \) subsets \( x_i \) of \( x \): \( \sum_i p_i = \text{sum}(px) = p \). \( px \) is the partition vector of \( x \)
- **py**: is the partition vector of \( y \) with \( k_y \) subsets \( y_j \), \( j=1,...,k_y \)
- **r**: is the wanted number of successive solutions \( r_{\text{max}} \leq \min(\min(px), \min(py), n) \)

**Details**

For the first solution, \( \sum_{i} \sum_{j} \rho^2(\text{cxi}[i, 1], \text{cyj}[j, 1]) \) is the optimized criterion. The other solutions are calculated from the same criterion, but with orthogonalities for having two by two zero correlated the canonical components defined for each \( x_i \), and also for those defined for each \( y_j \). Each solution associates \( k_x \) canonical components to \( k_y \) canonical components. When \( k_x = 1 \) (\( px = p \)), take `concorcano` function.

This function uses the `concorgm` function.

**Value**

List with following components

- **cx**: is a \( n.k_x \times r \) matrix of \( k_x \) row blocks \( cxi \) (\( n \times r \)). Each row block contains \( r \) partial canonical components
- **cy**: is a \( n.k_y \times r \) matrix of \( k_y \) row blocks \( cyj \) (\( n \times r \)). Each row block contains \( r \) partial canonical components
- **rho2**: is a \( k_x \times k_y \times r \) array; for a fixed solution \( k \), \( \text{rho2}[., k] \) contains \( k_x k_y \) squared correlations \( \rho^2(cx[n*(i-1) + 1 : n*i, k], cy[n*(j-1) + 1 : n*j, k]) \), simultaneously calculated between all the \( y_j \) with all the \( x_i \)
References


Examples

```r
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
cc<-concorgmcano(x,c(2,3),y,c(3,2,4),2)
diag(t(cc$x[,1:10,])%*%cc$y[,1:10,])/10^2
cc$rho2[1,1]
```

---

### concorgmreg

**Regression of subsets Yj by subsets Xi**

#### Description

Regression of subsets Yj by subsets Xi for comparing all the explanatory-explained pairs (Xi,Yj).

**SUCCESSIVE SOLUTIONS**

#### Usage

```r
concorgmreg(x,px,y,py,r)
```

#### Arguments

- **x** is a n x p matrix of p centered variables
- **y** is a n x q matrix of q centered variables
- **px** is a row vector which contains the numbers pi, i=1,...,kx, of the kx subsets xi of x : \( \sum p_i = \text{sum}(px) = p \). px is the partition vector of the columns of x.
- **py** is the partition vector of y with ky subsets yj, j=1,...,ky. \( \text{sum}(py) = q \)
- **r** is the wanted partition vector of \( \text{successive solutions} \) \( r_{max} \leq \min(\min(px),\min(py),n) \)

#### Details

For the first solution, \( \sum_j \sum_i \rho2(\text{cx}[,1], y_j * v_{j[,1]}), \text{var}(y_j * v_{j[,1]}) \) is the optimized criterion. The second solution is calculated from the same criterion, but with \( y_j - y_j * v_{j[,1]} * v_{j[,1]}' \) instead of the matrices yj and with orthogonalities for having two by two zero correlated the explanatory components defined for each matrix xi. And so on for the other solutions. One solution k associates kx explanatory components (in cx[,k]) to ky explained components. When kx =1 (px=p), take concorc function

This function uses the concorgm function
Value

list with following components

cx is a n.kx x r matrix of kx row blocks cxi (n x r). Each row block contains r partial explanatory components

v is a q x r matrix of ky row blocks vj (qj x r), the orthonormed partial axes of yj; The components yj*vj are the explained components

varexp is a kx x ky x r array; for a fixed solution k, the matrix varexp[,k] contains kxky explained variances obtained by a simultaneous regression of all the yj by all the xi, so the values \( \rho^2 \left( c_x [n \ast (i - 1) + 1 : n \ast i, k], y_j \ast v_j[, k] \right) \)

References


Examples

```r
x<-.matrix(runif(50),10,5);y<-.matrix(runif(90),10,9)
x<-.scale(x);y<-.scale(y)
cc<-.concorgmreg(x[,2,3],y[,3,2:4],2)
diag(t(cc$cx[,1:10])%*%y[,1:3]%*%cc$v[,1:3]/10)^2
cc$varexp[1,1,]
```

concors is a simultaneous concorgm

Description

concorgm with the set of r solutions simultaneously optimized

Usage

```r
concors(x,px,y,py,r)
```

Arguments

- `x` is a n x p matrix of p centered variables
- `y` is a n x q matrix of q centered variables
- `px` is a row vector which contains the numbers pi, i=1,...,kx, of the kx subsets xi of x : \( \sum_i p_i = \text{sum}(px) = p. \) px is the partition vector of x
- `py` is the partition vector of y with ky subsets yj, j=1,...,ky
- `r` is the wanted number of successive solutions rmax <= min(min(px),min(py),n)

Details

This function uses the svdbips function
Value

list with following components

\( u \) is a \( p \times r \) matrix of \( k_x \) row blocks \( u_i \) (\( p_i \times r \)), the orthonormed partial axes of \( x_i \); associated partial components: \( x_i^*u_i \)

\( v \) is a \( q \times r \) matrix of \( k_y \) row blocks \( v_j \) (\( q_j \times r \)), the orthonormed partial axes of \( y_j \); associated partial components: \( y_j^*v_j \)

\( \text{cov2} \) is a \( k_x \times k_y \times r \) array; for \( r \) fixed to \( k \), the matrix contains \( k_xk_y \) squared covariances \( \text{cov}(x_i^*u_i[k],y_j^*v_j[k])^2 \), the partial links between \( x_i \) and \( y_j \) measured with the solution \( k \)

References

See svdbips

Examples

```r
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
xs<-concorscano(x,px,y,py,r)
diag(t(x[,1:2]%*%xs$u[1:2,])%*%y[,1:3]%*%xs$v[1:3,])/10^2
xs$cov2[1,1,]
```

concorscano

"simultaneous concorgmcano"

Description

concorgmcano with the set of \( r \) solutions simultaneously optimized

Usage

concorscano(x,px,y,py,r)

Arguments

- \( x \) is a \( n \times p \) matrix of \( p \) centered variables
- \( y \) is a \( n \times q \) matrix of \( q \) centered variables
- \( p_x \) is a row vector which contains the numbers \( p_i, i=1,...,k_x \), of the \( k_x \) subsets \( x_i \) of \( x: \sum_i p_i=\text{sum}(px)=p. p_x \) is the partition vector of \( x \)
- \( p_y \) is the partition vector of \( y \) with \( k_y \) subsets \( y_j, j=1,...,k_y \)
- \( r \) is the wanted number of successive solutions \( r_{\text{max}} <= \min(\min(px),\min(py),n) \)

Details

This function uses the concors function
Value

list with following components

cx  is a n.kx x r matrix of kx row blocks cxi (n x r). Each row block contains r partial canonical components

cy  is a n.ky x r matrix of ky row blocks cyj (n x r). Each row block contains r partial canonical components

rho2  is a kx x ky x r array; for a fixed solution k, rho2["k"] contains kxky squared correlations $\rho(cx[n \ast (i - 1) + 1 : n \ast i, k], cy[n \ast (j - 1) + 1 : n \ast j, k])^2$, simultaneously calculated between all the yj with all the xi

References

See svdbips

Examples

x<-matrix(rnorm(50),10,5);y<-matrix(rnorm(90),10,9)
x<-scale(x);y<-scale(y)
cca<-concorscano(x,c(2,3),y,c(3,2,4),2)
diag(t(cca$cx[1:10,]%*%cca$cy[1:10,]/10)^2)
cca$rho2[1,1,

concorsreg  "simultaneous concorgmreg"

Description

concorgmreg with the set of r solutions simultaneously optimized

Usage

concorsreg(x,px,y,py,r)

Arguments

x  is a n x p matrix of p centered variables
y  is a n x q matrix of q centered variables
px  is a row vector which contains the numbers pi, i=1,....kx, of the kx subsets xi of x : sum(pi)=sum(px)=p. px is the partition vector of x
py  is the partition vector of y with ky subsets yj, j=1,....ky
r  is the wanted number of successive solutions rmax <= min(min(px),min(py),n)

Details

This function uses the concors function
Value

list with following components

- **cx** is a n.kx x r matrix of kx row blocks cxi (n x r). Each row block contains r partial explanatory components
- **v** is a q x r matrix of ky row blocks vj (qj x r), the orthonormed partial axes of yj; The components yj*vj are the explained components.
- **varexp** is a kx x ky x r array; for a fixed solution k, the matrix varexp[,k] contains kxky explained variances obtained by a simultaneous regression of all the yj by all the xi, so the values rho2(cx[n*(i-1)+1:n*i,k],yj*vj[,k])

References

See svdbip

Examples

```r
x<-matrix(rnorm(50),10,5);y<-matrix(rnorm(90),10,9)
x<-scale(x);y<-scale(y)
crs<-concorsreg(x,c(2,3),y,c(3,2,4),2)
diag(t(crs$x[1:10,])%*%y[,1:3]%*%crs$v[1:3,]/10)^2
crs$varexp[1,1]
```

svdbip

**SVD for one bipartitioned matrix x**

Description

SVD for bipartitioned matrix x. r successive Solutions

Usage

```r
svdbip(x,K,H,r)
```

Arguments

- **x** is a p x q matrix
- **K** is a row vector which contains the numbers pk, k=1,...,kx, of the partition of x with kx row blocks : sum(pk)=p
- **H** is a row vector which contains the numbers qh, h=1,...,ky, of the partition of x with ky column blocks : sum(qh)=q
- **r** is the wanted number of successive solutions
**svdbip**

### Details

The first solution calculates $k_x + k_y$ normed vectors: $k_x$ vectors $u_k[:,1]$ of $R^{p_k}$ associated to $k_y$ vectors $v_h[:,1]$'s of $R^{q_h}$, by maximizing $\sum_k \sum_h (u_k[:,1]' \times x_{kh} \times v_h[:,1])^2$, with $k_x + k_y$ norm constraints. A value $(u_k[:,1]' \times x_{kh} \times v_h[:,1])^2$ measures the relative link between $R^{p_k}$ and $R^{q_h}$ associated to the block $x_{kh}$.

The second solution is obtained from the same criterion, but after replacing each $x_{hk}$ by $x_{kh} - x_{kh} \times v_h^* u_k^* x_{kh} + u_k^* x_{kh} \times v_h^* v_h'$. And so on for the successive solutions $1, 2, ..., r$. The biggest number of solutions may be $r = \inf(pk, qh)$, when the $x_{kh}$'s are supposed with full rank; then $r_{max} = \min(\min(K), \min(H))$.

When $K=p$ (or $H=q$, with $t(x)$), svdcp function is better. When $H=q$ and $K=p$, it is the usual svd (with squared singular values).

Convergence of algorithm may be not global. So the below proposed initialisation of the algorithm may be not very suitable for some data sets. Several different random initialisations with normed vectors might be considered and the best result then chosen.

### Value

- **u** is a $p \times r$ matrix of $k_x$ row blocks $u_k$ ($p_k \times r$); $u_k' u_k = \text{Identity}$.
- **v** is a $q \times r$ matrix of $k_y$ row blocks $v_h$ ($q_h \times r$); $v_h' v_h = \text{Identity}$
- **s2** is a $k_x \times k_y \times r$ array; with $r$ fixed, each matrix contains $k_x k_y$ values $(u_h' \times x_{kh} \times v_k)^2$, the partial (squared) singular values relative to $x_{kh}$.

### References


### Examples

```r
x<-matrix(runif(200),10,20)
s<-svdbip(x,c(3,4,3),c(5,15),2)
zu<-cbind(x[1:3,1:5]%*%s$v[1:5,1],x[1:3,6:20]%*%s$v[6:20,1])
czu<-svd(zu);
czu$s[,1]%*%s[u[1:3,2:3]
czu$s[,1] # is a compromise between the vectors xj*vj[,1],
# orthogonal to the partial vectors uk[,k] relative to the
# following solutions (k>1); (in a same way, the singular
# vectors ui and vj of an usual SVD of x verifies ui'*x*vj)=0,
# when i is not equal to j)
```
svdbip2

SVD for bipartitioned matrix x

Description

SVD for bipartitioned matrix x. r successive Solutions. As SVDBIP, but with another algorithm and another initialisation.

Usage

svdbip2(x,K,H,r)

Arguments

x is a p x q matrix
K is a row vector which contains the numbers pk, k=1,...,kx, of the partition of x with kx row blocks: \( \sum_k p_k = p \)
H is a row vector which contains the numbers qh, h=1,...,ky, of the partition of x with ky column blocks: \( \sum_q q_h = q \)
r is the wanted number of successive solutions

Details

The first solution calculates kx+ky normed vectors: kx vectors uk[,1] of Rp\(k\) associated to ky vectors vh[,1]'s of Rqh, by maximizing \( \sum_k \sum_h (u_k[,1]' * x_{kh} * v_h[,1])^2 \), with kx+ky norm constraints. A value \((u_k[,1]' * x_{kh} * v_h[,1])^2\) measures the relative link between Rp\(k\) and Rqh associated to the block x\(kh\).

The second solution is obtained from the same criterion, but after replacing each x\(hk\) by x\(kh\)-x\(kh\)*vh*vh'-uk*uk'x\(kh\)+uk'*vh*vh'. And so on for the successive solutions 1,2,...,r. The biggest number of solutions may be \( r=\inf(pk,qh) \), when the x\(kh\)'s are supposed with full rank; then \( r_{\text{max}}=\min(\min(K),\min(H)) \).

When K=p (or H=q, with t(x)), svdcp function is better. When H=q and K=p, it is the usual svd (with squared singular values).

Convergence of algorithm may be not global. So the below proposed initialisation of the algorithm may be not very suitable for some data sets. Several different random initialisations with normed vectors might be considered and the best result then choosen.

Value

list with following components

u is a p x r matrix of kx row blocks uk (pk x r); uk'*uk = Identity
v is a q x r matrix of ky row blocks vh (qh x r); vh'*vh = Identity
s2 is a kx x ky x r array; with r fixed, each matrix contains kxky values \((u_h'*x_{kh} * v_k)^2\), the partial (squared) singular values relative to x\(kh\)
svdbips

References

Examples
x<-matrix(runif(200),10,20)
s2<-svdbip2(x,c(3,4,3),c(5,5,10),3);s2$s2
s1<-svdbip(x,c(3,4,3),c(5,5,10),3);s1$s2

svdbips

SVD for bipartitioned matrix x

Description
SVD for bipartitioned matrix x. SIMULTANEOUS SOLUTIONS. ("simultaneous svdbip")

Usage
svdbips(x,K,H,r)

Arguments
x is a p x q matrix
K is a row vector which contains the numbers pk, k=1,...,kx, of the partition of x with kx row blocks : \( \sum \Sigma p_k = p \)
H is a row vector which contains the numbers qh, h=1,...,ky, of the partition of x with ky column blocks : \( \sum \Sigma q_h = q \)
r is the wanted number of solutions

Details
One set of r solutions is calculated by maximizing \( \sum \Sigma \Sigma (u_k[,i]' * x_{kh} * v_h[,i])^2 \), with kx-ky orthonormality constraints (for each uk and each vh). For each fixed r value, the solution is totally new (does’nt consist to complete a previous calculus of one set of r-1 solutions). \( r_{max}=\min(\min(K),\min(H)) \). When r=1, it is svdbip (thus it is svdcp when r=1 and kx=1).

Convergence of algorithm may be not global. So the below proposed initialisation of the algorithm may be not very suitable for some data sets. Several different random initialisations with normed vectors might be considered and the best result then choosen....

Value
list with following components
u is a p x r matrix of kx row blocks uk (pk x r); uk’*uk = Identity
v is a q x r matrix of ky row blocks vh (qh x r); vh’*vh = Identity
s2 is a kx x ky x r array; for a fixed solution k, each matrix s2[,k] contains kxky values \( (u'_h * x_{kh} * v_k)^2 \), the "partial (squared) singular values" relative to \( x_{kh} \).
svdcp

References


Examples

\[
\begin{align*}
x & \leftarrow \text{matrix}(\text{runif}(200), 10, 20) \\
\text{s1} & \leftarrow \text{svdbip}(x, c(3, 4, 3), c(5, 5, 10), 2); \text{sum(sum(sum(s1$s2)))} \\
\text{ss} & \leftarrow \text{svdbips}(x, c(3, 4, 3), c(5, 5, 10), 2); \text{sum(sum(sum(ss$s2)))}
\end{align*}
\]

svdcp

SVD for a Column Partitioned matrix x

Description

SVD for a Column Partitioned matrix x. r global successive solutions

Usage

svdcp(x, H, r)

Arguments

- **x**: is a p x q matrix
- **H**: is a row vector which contains the numbers q_i, i=1,...,kx, of the partition of x with kx column blocks x_i: \( \sum q_i = q \).
- **r**: is the wanted number of successive solutions.

Details

The first solution calculates 1+kx normed vectors: the vector u[1] of \( R_p \) associated to the kx vectors v_i[1]’s of \( R_q \). by maximizing \( \sum_i (u[1]’ * x_i * v_i[1])^2 \), with 1+kx norm constraints. A value \( (u[1]’ * x_i * v_i[1])^2 \) measures the relative link between \( R_p \) and \( R_q \) associated to x_i. It corresponds to a partial squared singular value notion, since \( \sum_i (u[1]’ * x_i * v_i[1])^2 = s^2 \), where s is the usual first singular value of x.

The second solution is obtained from the same criterion, but after replacing each x_i by x_i-x_i*v_i[1]*v_i[1]’.

And so on for the successive solutions 1,2,...,r . The biggest number of solutions may be r=inf(p,qi), when the x_i’s are supposed with full rank; then rmax=min(min(H),p).

Value

- **u**: is a p x r matrix; u’*u = Identity
- **v**: is a q x r matrix of kx row blocks v_i (qi x r); v_i’*v_i = Identity
- **s2**: is a kx x r matrix; each column k contains kx values \( (u[k]’ * x_i * v_i[k])^2 \), the partial (squared) singular values relative to x_i
References


Examples

```r
x <- matrix(runif(200), 10, 20)
s <- svdcp(x, c(5, 5, 10), 1)
ss <- svd(x); ss$d[1]^2
sum(s$s2)
```
Index

concor, 2
concorcano, 3
concoreg, 4
concorgm, 6
concorgmcano, 7
concorgmreg, 8
concors, 9
concorscano, 10
concorsreg, 11

svdbip, 12
svdbip2, 14
svdbips, 15
svdcp, 16