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` *Relative links of several subsets of variables*

Description

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

Usage

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

Arguments

`x,y`	are n x p and n x q matrices of p and q centered columns
`py`	is a row vector which contains the numbers qi, i=1,...,ky, of the ky subsets yi of y : sum(qi)=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 Rp associated to the ky vectors vi[:,1]'s of Rqi, 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 yi*vi[,k] and to a global component y*V[,k]. $\text{cov}(x * u[,k], y * V[,k])^2 = \sum \text{cov}(x * u[,k], y_i * v_i[,k])^2$. y*V[,k] is a global component of the components yi*vi[,k].

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,qi), when the x'*yi's are supposed with full rank; then rmax=min(c(min(py),n,p)). For a set of r solutions, the matrix u'X'YV is diagonal and the matrices u'X'Yjvj are triangular (good partition of the link by the solutions). concor.m is the svdcp.m function applied to the matrix x'y.

Value

list with following components

`u`	is a p x r matrix of axes in Rp relative to x; u'*u = Identity
`v`	is a q x r matrix of ky row blocks vi (qi x r) of axes in Rqi relative to yi; vi'*vi = Identity
`V`	is a q x r matrix of axes in Rq relative to y; V'*V = Identity
`cov2`	is a ky x 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

Lafosse R. & Hanafi M.(1997) Concordance d'un tableau avec K tableaux: Definition de K+1 uples synthetiques. Revue de Statistique Appliquee vol.45,n.4.

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)
co<-concor(x,y,c(3,2,4),2)
((t(x%*%co$u[,1])%*%y[,1:3]%*%co$v[1:3,1])/10)^2;co$cov2[1,1]
t(x%*%co$u)%*%y%*%co$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 : $\sum_i q_i$=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 cyi[,1] of yi by maximizing $\sum_i \rho(cx[,1], cy_i[,1])^2$.

The second solution is obtained from the same criterion, with ky orthogonality constraints for having rho(cyi[,1],cyi[,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)'*cyi 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 $\rho(cx[,k], cy_i[,k])^2$

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

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
ca<-concorcano(x,y,c(3,2,4),2)
diag(t(ca$cx)%*%ca$cy[1:10,]/10)^2
ca$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 $n \times p$ matrix of p centered explanatory variables
- `y` is a $n \times q$ 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 : $\sum_i q_i$ = 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 $varexp1 = \sum_i \rho(cx[,1], y_i * v_i[,1])^2 \text{var}(y_i * v_i[,1]))$, 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]. $\rho(cx[,k], y * V[,k])^2 \text{var}(y * V[,k]) =$ varexpk. 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 Rqi relative to yi; $v_i' * v_i = \text{Id}$
- `V` — is a $q \times r$ matrix of axes in Rq 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

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.

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

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
co<-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])
```

concorgm — *Analyzing a set of partial links between Xi and Yj*

Description

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, $\sum_i \sum_j \text{cov2}(x_i * u_i[,1], y_j * v_j[,1])$ is the optimized criterion. The second solution is calculated from the same criterion, but with $x_i - x_i * u_i[,1] * u_i[,1]'$ and $y_j - y_j * v_j[,1] * v_j[,1]'$ 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 $\text{cov2}(x_i*u_i[,k], y_j*v_j[,k])^2$, the partial links between xi and yj measured with the solution k.

References

Kissita, Cazes, Hanafi & Lafosse (2004) Deux methodes d’analyse factorielle du lien entre deux tableaux de variables partitionnées. Revue de Statistique Appliquée, Vol 52, n° 3, 73-92.

Examples

```
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 Yj with subsets Xi*

Description

Canonical analysis of subsets Yj with subsets Xi. Relative valuations by squared correlations of the proximities of subsets Xi with subsets Yj. SUCCESSIVE SOLUTIONS

Usage

```
concorgmcano(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 : $\sum_i p_i$=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, $sum_i sum_j$rho2$(cx_i[,1], cy_j[,1])$ 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 xi, and also for those defined for each yj. Each solution associates kx canonical components to ky canonical components. When kx =1 (px=p), take concorcano 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 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 $rho2(cx[n*(i-1)+1:n*i,k], cy[n*(j-1)+1:n*j,k])$, simultaneously calculated between all the yj with all the xi

References

Kissita G., Analyse canonique generalisee avec tableau de reference generalisee. Thesis, Ceremade Paris 9 Dauphine (2003).

Examples

```
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$cx[1:10,])%*%cc$cy[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

```
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 : $\sum p_i$=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. sum(py)=q
`r`	is the wanted number of successive solutions rmax <= min(min(px),min(py),n)

Details

For the first solution, $\sum_i \sum_j$ rho2$(cx_i[,1], y_j * v_j[,1])var(y_j * v_j[,1])$ 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 concoreg 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 rho2$(cx[n*(i-1)+1:n*i,k],y_j*v_j[,k])var(y_j*v_j[,k])$

References

Hanafi & Lafosse (2004) Regression of a multi-set by another based on an extension of the SVD. COMPSTAT'2004 Symposium

Examples

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

---

concors *"simultaneous concorgm"*

Description

concorgm with the set of r solutions simultaneously optimized

Usage

```
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 : $\sum_i p_i$=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 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 cov$(x_i * u_i[, k], y_j * v_j[, k])^2$, the partial links between xi and yj measured with the solution k

References

See svdbips

Examples

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
cs<-concors(x,c(2,3),y,c(3,2,4),2)
diag(t(x[,1:2]%*%cs$u[1:2,])%*%y[,1:3]%*%cs$v[1:3,]/10)^2
cs$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 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 : $\sum_i p_i$=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 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 $\rho(cx[n*(i-1)+1:n*i,k], cy[n*(j-1)+1:n*j,k])^2$, simultaneously calculated between all the yj with all the xi

References

See svdbips

Examples

```
x<-matrix(runif(50),10,5);y<-matrix(runif(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$(cx[n*(i-1)+1:n*i,k], y_j*v_j[,k])var(y_j*v_j[,k])$

References

See svdbips

Examples

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
crs<-concorsreg(x,c(2,3),y,c(3,2,4),2)
diag(t(crs$cx[1:10,])%*%y[,1:3]%*%crs$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

```
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

Details

The first solution calculates kx+ky normed vectors: kx vectors uk[:,1] of R^{p_k} associated to ky vectors vh[:,1]'s of R^{q_h}, 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 R^{p_k} and R^{q_h} associated to the block xkh.

The second solution is obtained from the same criterion, but after replacing each xhk by xkh-xkh*vh*vh'-uk*uk'xkh+uk*uk'xkh*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 xkh's are supposed with full rank; then rmax=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 xkh.

References

Kissita G., Cazes P., Hanafi M. & Lafosse (2004) Deux methodes d'analyse factorielle du lien entre deux tableaux de variables partitiones. Revue de Statistique Appliquee.

Examples

```
x<-matrix(runif(200),10,20)
s<-svdbip(x,c(3,4,3),c(5,15),3)
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$u[,1]%*%s$u[1:3,2:3]
czu$u[,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 : $\sum q_h = q$
`r`	is the wanted number of successive solutions

Details

The first solution calculates kx+ky normed vectors: kx vectors uk[:,1] of Rpk 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 R^{p_k} and R^{q_h} associated to the block xkh.

The second solution is obtained from the same criterion, but after replacing each xhk by xkh-xkh*vh*vh'-uk*uk'xkh+uk*uk'xkh*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 xkh's are supposed with full rank; then rmax=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}

References

Kissita G., Analyse canonique generalisee avec tableau de reference generalisee. Thesis, Ceremade Paris 9 Dauphine (2003)

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 : $\sum_k p_k = p$
`H`	is a row vector which contains the numbers qh, h=1,...,ky, of the partition of x with ky column blocks : $\sum_h q_h = q$
`r`	is the wanted number of solutions

Details

One set of r solutions is calculated by maximizing $\sum_i \sum_k \sum_h (u_k[,i]' * x_{kh} * v_h[,i])^2$, 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). rmax=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}.

References

Lafosse R. & Ten Berge J. A simultaneous CONCOR method for the analysis of two partitioned matrices. submitted.

Examples

```
x<-matrix(runif(200),10,20)
s1<-svdbip(x,c(3,4,3),c(5,5,10),2);sum(sum(sum(s1$s2)))
ss<-svdbips(x,c(3,4,3),c(5,5,10),2);sum(sum(sum(ss$s2)))
```

---

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 qi, i=1,...,kx, of the partition of x with kx column blocks xi : $\sum q_i = q$.
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 vi[,1]'s of R^{q_i}. 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_i} associated to xi. 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 xi by xi-xi*vi[,1]*vi[,1]'. And so on for the successive solutions 1,2,...,r . The biggest number of solutions may be r=inf(p,qi), when the xi's are supposed with full rank; then rmax=min([min(H),p]).

Value

list with following components

u	is a p x r matrix; u'*u = Identity
v	is a q x r matrix of kx row blocks vi (qi x r); vi'*vi = 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 xi

References

Lafosse R. & Hanafi M.(1997) Concordance d'un tableau avec K tableaux: Definition de K+1 uples synthetiques. Revue de Statistique Appliquee vol.45,n.4.

Examples

```
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