Package 'ThreeWay'

October 12, 2022

Type Package

Title Three-Way Component Analysis

Version 1.1.3

Date 2015-09-07

Author Maria Antonietta Del Ferraro, Henk A.L. Kiers, Paolo Giordani

Maintainer Paolo Giordani <paolo.giordani@uniroma1.it>

Description Component analysis for three-
way data arrays by means of Candecomp/Parafac, Tucker3, Tucker2 and Tucker1 models.

Depends R(>= 2.8.1), base, stats, graphics, grDevices

License GPL (>= 2)

LazyLoad yes

Repository CRAN

NeedsCompilation no

Date/Publication 2015-09-07 10:22:29

R topics documented:

bootstrapCP *Bootstrap percentile intervals for CANDECOMP/PARAFAC*

Description

Produces percentile intervals for all output parameters. The percentile intervals indicate the instability of the sample solutions.

Usage

```
bootstrapCP(X, A, B, C, n, m, p, r, ort1, ort2, ort3, conv, centopt, normopt,
  scaleopt, maxit, laba, labb, labc)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
n	Number of A-mode entities of X
m	Number of B-mode entities of X
p	Number of C-mode entities of X
r	Number of extracted components
ort1	Type of constraints on A (see CP)
ort2	Type of constraints on B (see CP)
ort3	Type of constraints on C (see CP)
conv	Convergence criterion
centopt	Centering option (see cent3)
normopt	Normalization option (see norm3)
scaleopt	Scaling option (see renormsolCP)
maxit	Maximal number of iterations
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

A list including the following components:

Bint	Bootstrap percentile interval of every element of B
Cint	Bootstrap percentile interval of every element of C
fpint	Bootstrap percentile interval for the goodness of fit index expressed as a percentage

Note

The preprocessing must be done in same way as for sample analysis.
The resampling mode must be the A-mode.
The starting points for every bootstrap solution are two: rational (using SVD) and solution from the observed sample.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2004). Bootstrap confidence intervals for three-way methods. *Journal of Chemometrics 18:22–36*.

See Also

bootstrapT3, CP, percentile95

Examples

```
data(TV)
TVdata=TV[[1]]
labSCALE=TV[[2]]
labPROGRAM=TV[[3]]
labSTUDENT=TV[[4]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
# CP solution
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 1, 1, 1, 0, 1e-6, 10000)
## Not run:
# Bootstrap analysis on CP solution
boot <- bootstrapCP(TVdata, TVcp$A, TVcp$B, TVcp$C, 30, 16, 15, 2, 1, 1, 1,
 1e-6, 0, 0, 0, 10000, labSTUDENT, labSCALE, labPROGRAM)
# Bootstrap analysis on CP solution (when labels are not available)
boot <- bootstrapCP(TVdata, TVcp$A, TVcp$B, TVcp$C, 30, 16, 15, 2, 1, 1, 1,
 1e-6, 0, 0, 0, 10000)

## End(Not run)
```

---

bootstrapT3 *Bootstrap percentile intervals for Tucker3*

Description

Produces percentile intervals for all output parameters. The percentile intervals indicate the instability of the sample solutions.

Usage

```
bootstrapT3(X, A, B, C, G, n, m, p, r1, r2, r3, conv, centopt, normopt,
  optimalmatch, laba, labb, labc)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
G	Matricized core array (frontal slices)
n	Number of A-mode entities of X
m	Number of B-mode entities of X
p	Number of C-mode entities of X
r1	Number of extracted components for the A-mode
r2	Number of extracted components for the B-mode
r3	Number of extracted components for the C-mode
conv	Convergence criterion
centopt	Centering option (see cent3)
normopt	Normalization option (see norm3)
optimalmatch	Binary indicator (0 if the procedure uses matching via orthogonal rotation towards full solutions, 1 if the procedure uses matching via optimal transformation towards full solutions)
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

A list including the following components:

Bint	Bootstrap percentile interval of every element of B
Cint	Bootstrap percentile interval of every element of C
Gint	Bootstrap percentile interval of matricized core array (frontal slices) G
fpint	Bootstrap percentile interval for the goodness of fit index expressed as a percentage

Note

The preprocessing must be done in same way as for sample analysis.
The resampling mode must be the A-mode.
The starting points for every bootstrap solution are two: rational (using SVD) and solution from the observed sample.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2004). Bootstrap confidence intervals for three-way methods. *Journal of Chemometrics 18:22–36*.

See Also

bootstrapCP, percentile95, T3

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5],1,1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)],3,8)
# T3 solution
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 0, 1e-6)
## Not run:
# Bootstrap analysis on T3 solution using matching via optimal transformation
boot <- bootstrapT3(Bus, BusT3$A, BusT3$B, BusT3$C, BusT3$H, 7, 5, 37, 2, 2, 2,
 1e-6, 0, 0, 1, laba, labb, labc)
# Bootstrap analysis on T3 solution using matching via orthogonal rotation
# (when labels are not available)
boot <- bootstrapT3(Bus, BusT3$A, BusT3$B, BusT3$C, BusT3$H, 7, 5, 37, 2, 2, 2,
 1e-6, 0, 0, 0)

## End(Not run)
```

Bus *Bus data*

Description

Three-way data about the process of learning to read of seven first-grade children tested weekly (from week 3 to 47, but weeks 10, 19, 20, 29, 35, 36, 39, 43 were holidays and, thus, data on 37 weeks) with five different tests.

Usage

```
data(Bus)
```

Format

A matrix with 7 rows and 185 (5x37) columns.
The rows refer to the pupils.
The columns refer to the combinations of tests and weeks with the tests nested within the weeks.
The matrix contains the frontal slices next to each other of the original array.
The meanings and the ranges of the tests are as follows:
`L`: letter knowledge test (scores in 0-47);
`P`: regular orthographic short words (scores in 0-10);
`Q`: regular orthographic long words (scores in 0-10);
`S`: regular orthographic long and short words within context (scores in 0-15);
`R`: irregular orthographic long and short words (scores in 0-15).

Details

In the literature the Bus data have been analyzed by Tucker3 (see Kroonenberg, 1983; Timmerman, 2001). There is consensus on normalizing the data so to eliminate artificial differences among ranges of tests. Different centering options and numbers of extracted components have been chosen. Specifically, Kroonenberg (1983) suggests averaging over pupils and tests for each time occasions and extracting two components for every mode. Timmerman (2001) suggests to apply Tucker3 to the normalized data with two components for pupils and time occasions and one component for tests.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

P.M. Kroonenberg (1983). *Three-mode Principal Component Analysis. Theory and Applications*. DSWO Press, Leiden.
M.E. Timmerman (2001). *Component Analysis of Multisubject Multivariate Longitudinal Data*. Ph.D. Thesis, University of Groningen.

`Cc` *Columnwise centering of a matrix*

Description

Computation of a columnwise centered version of a matrix.

Usage

```
Cc(A)
```

Arguments

`A`	Matrix of any order

Value

`Ac`	Matrix columnwise centered

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

See Also

`nrm2`

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
Y <- Cc(X)
apply(Y,2,mean)
```

---

ccmat — *Columns concatenation*

Description

Concatenates the columns of two matrices next to each other.

Usage

```
ccmat(A, B)
```

Arguments

`A`	Matrix of the same order of `B`
`B`	Matrix of the same order of `A`

Value

`mat`	Matrix in which the columns of `A` and `B` are concatenated next to each other

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
Y <- matrix(rnorm(6*3),ncol=3)
Z <- ccmat(X,Y)
```

---

cent3 — *Centering of a matricized array*

Description

Centering of a matricized array across one mode (modes indicated by 1,2, or 3).

Usage

```
cent3(X, n, m, p, mode)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
mode	Centering option (1 if X is centered across A-mode, 2 if X is centered across B-mode, 3 if X is centered across C-mode)

Value

Y	Matrix of order (n x mp) containing the centered matricized array (frontal slices)

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122*.

See Also

Cc, norm3

Examples

```
X <- array(c(rnorm(120)),c(6,5,4))
# matricized array
Y <- supermat(X)
# data centered across A-mode
Z <- cent3(Y$Xa, 6, 5, 4, 1)
apply(Z,2,mean)
# data centered also across B-modes (double centering)
Z <- cent3(Z, 6, 5, 4, 2)
apply(Z,1,mean)
apply(Z,2,mean)
```

CP — *Interactive Candecomp/Parafac analysis*

Description

Detects the underlying structure of a three-way array according to the Candecomp/Parafac (CP) model.

Usage

```
CP(data,laba,labb,labc)
```

Arguments

`data`	Array of order `n` by `m` by `p` or matrix or data.frame of order (`n x mp`) containing the matricized array (frontal slices)
`laba`	Optional vector of length `n` containing the labels of the `A`-mode entities
`labb`	Optional vector of length `m` containing the labels of the `B`-mode entities
`labc`	Optional vector of length `p` containing the labels of the `C`-mode entities

Value

A list including the following components:

`A`	Component matrix for the `A`-mode
`B`	Component matrix for the `B`-mode
`C`	Component matrix for the `C`-mode
`fit`	Fit value expressed as a percentage
`tripcos`	Matrix of the triple cosines among pairs of components (to inspect degeneracy)
`fitValues`	Fit values expressed as a percentage upon convergence for all the runs of the CP algorithm (see `CPfunc`)
`funcValues`	Function values upon convergence for all the runs of the CP algorithm (see `CPfunc`)

`cputime`	Computation times for all the runs of the CP algorithm (see `CPfunc`)
`iter`	Numbers of iterations upon convergence for all the runs of the CP algorithm (see `CPfunc`)
`fitA`	Fit contributions for the `A`-mode entities (see `CPfitpartitioning`)
`fitB`	Fit contributions for the `B`-mode entities (see `CPfitpartitioning`)
`fitC`	Fit contributions for the `C`-mode entities (see `CPfitpartitioning`)
`Bint`	Bootstrap percentile interval of every element of `B` (see `bootstrapCP`)
`Cint`	Bootstrap percentile interval of every element of `C` (see `bootstrapCP`)
`fpint`	Bootstrap percentile interval for the goodness of fit index expressed as a percentage (see `bootstrapCP`)
`Afull`	Component matrix for the `A`-mode (full data) from split-half analysis (see `splithalfCP`)
`As1`	Component matrix for the `A`-mode (split n.1) from split-half analysis (see `splithalfCP`)
`As2`	Component matrix for the `A`-mode (split n.2) from split-half analysis (see `splithalfCP`)
`Bfull`	Component matrix for the `B`-mode (full data) from split-half analysis (see `splithalfCP`)
`Bs1`	Component matrix for the `B`-mode (split n.1) from split-half analysis (see `splithalfCP`)
`Bs2`	Component matrix for the `B`-mode (split n.2) from split-half analysis (see `splithalfCP`)
`Cfull`	Component matrix for the `C`-mode (full data) from split-half analysis (see `splithalfCP`)
`Cs1`	Component matrix for the `C`-mode (split n.1) from split-half analysis (see `splithalfCP`)
`Cs2`	Component matrix for the `C`-mode (split n.2) from split-half analysis (see `splithalfCP`)
`A1`	Component matrix for the `A`-mode from Principal Component Analysis of mean values (see `pcamean`)
`B1`	Component matrix for the `B`-mode from Principal Component Analysis of mean values (see `pcamean`)
`C1`	Component matrix for the `C`-mode from Principal Component Analysis of mean values (see `pcamean`)
`A2`	Component matrix for the `A`-mode from Principal Component Analysis of mean values (see `pcamean`)
`B2`	Component matrix for the `B`-mode from Principal Component Analysis of mean values (see `pcamean`)
`C2`	Component matrix for the `C`-mode from Principal Component Analysis of mean values (see `pcamean`)
`laba`	Vector of length `n` containing the labels of the `A`-mode entities
`labb`	Vector of length `m` containing the labels of the `B`-mode entities
`labc`	Vector of length `P` containing the labels of the `C`-mode entities
`Xprep`	Matrix of order (`n x mp`) containing the matricized array (frontal slices) after preprocessing used for the analysis

Author(s)

Maria Antonietta Del Ferraro `<mariaantonietta.delferraro@yahoo.it>`
Henk A.L. Kiers `<h.a.l.kiers@rug.nl>`
Paolo Giordani `<paolo.giordani@uniroma1.it>`

References

J.D. Carroll and J.J. Chang (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. *Psychometrika 35:283–319*.
P. Giordani, H.A.L. Kiers, M.A. Del Ferraro (2014). Three-way component analysis using the R package ThreeWay. *Journal of Statistical Software 57(7):1–23*. http://www.jstatsoft.org/v57/i07/.
R.A. Harshman (1970). Foundations of the Parafac procedure: models and conditions for an 'explanatory' multi-mode factor analysis. *UCLA Working Papers in Phonetics 16:1–84*.
P.M. Kroonenberg (2008). *Applied Multiway Data Analysis*. Wiley, New Jersey.

See Also

T3, T2, T1

Examples

```
data(TV)
TVdata=TV[[1]]
labSCALE=TV[[2]]
labPROGRAM=TV[[3]]
labSTUDENT=TV[[4]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
## Not run:
# interactive CP analysis
TVcp <- CP(TVdata, labSTUDENT, labSCALE, labPROGRAM)
# interactive CP analysis (when labels are not available)
TVcp <- CP(TVdata)

## End(Not run)
```

CPdimensionalityplot *Plot fit of Candecomp/Parafac*

Description

Plots fits against numbers of dimensions, with S as labels and fits against number of effective paramaters.

Usage

```
CPdimensionalityplot(A, n, m, p)
```

Arguments

A	A matrix with columns: number of components, goodness of fit (%)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities

Note

A is usually the first and fourth columns of the output of DimSelector.
The number of effective parameters in a Candecomp/Parafac analysis is discussed in Weesie and Van Houwelingen (1983).

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

E. Ceulemans \& H.A.L. Kiers (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology 59:133–150.*
J. Weesie \& H. Van Houwelingen (1983). *GEPCAM users' manual (first draft).* Utrecht, The Netherlands: Institute of Mathematical Statistics, State University of Utrecht.

See Also

CP, DimSelector

Examples

```
data(TV)
TVdata=TV[[1]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
# Fit values of CP with different numbers of components (from 1 to 5)
FitCP <- CPrunsFit(TVdata, 30, 16, 15, 5)
OutCP <- FitCP[,c(1,4)]
CPdimensionalityplot(OutCP, 30, 16, 15)
```

`CPfitpartitioning` *Fit of each entity per mode*

Description

Computation of fit contributions.

Usage

```
CPfitpartitioning(Xprep, n, m, p, A, B, C, laba, labb, labc)
```

Arguments

Xprep	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

A list including the following components:

fitA	Fit contribution for the A-mode entities
fitB	Fit contribution for the B-mode entities
fitC	Fit contribution for the C-mode entities

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

See Also

CP

Examples

```
data(TV)
TVdata=TV[[1]]
labSCALE=TV[[2]]
labPROGRAM=TV[[3]]
labSTUDENT=TV[[4]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
# CP solution
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 1, 1, 1, 0, 1e-6, 10000)
# Fitpartitioning of the CP solution
FitCP <-  CPfitpartitioning(TVdata, 30, 16, 15, TVcp$A, TVcp$B, TVcp$C,
 labSTUDENT, labSCALE, labPROGRAM)
# Fitpartitioning of the CP solution (when labels are not available)
FitCP <-  CPfitpartitioning(TVdata, 30, 16, 15, TVcp$A, TVcp$B, TVcp$C)
```

CPfunc *Algorithm for the Candecomp/Parafac (CP) model*

Description

Alternating Least Squares algorithm for the minimization of the Candecomp/Parafac loss function.

Usage

```
CPfunc(X, n, m, p, r, ort1, ort2, ort3, start, conv, maxit, A, B, C)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n *x* mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
r	Number of extracted components
ort1	Type of constraints on A (1 for no constraints, 2 for orthogonality constraints, 3 for zero correlations constraints)
ort2	Type of constraints on B (1 for no constraints, 2 for orthogonality constraints, 3 for zero correlations constraints)
ort3	Type of constraints on C (1 for no constraints, 2 for orthogonality constraints, 3 for zero correlations constraints)
start	Starting point (0 for starting point of the algorithm from SVD's, 1 for random starting point (orthonormalized component matrices), 2 for user specified components

conv	Convergence criterion
maxit	Maximal number of iterations
A	Optional (necessary if start=2) starting value for A
B	Optional (necessary if start=2) starting value for B
C	Optional (necessary if start=2) starting value for C

Value

A list including the following components:

A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
f	Loss function value
fp	Fit value expressed as a percentage
iter	Number of iterations
tripcos	Minimal triple cosine between two components across three component matrices (to inspect degeneracy)
mintripcos	Minimal triple cosine during the iterative algorithm observed at every 10 iterations (to inspect degeneracy)
ftiter	Matrix containing in each row the function value and the minimal triple cosine at every 10 iterations
cputime	Computation time

Note

The loss function to be minimized is $sum(k)||X(k) - AD(k)B'||^2$, where $D(k)$ is a diagonal matrix holding the k-th row of C.
CPfunc is the same as CPfuncrep except that all printings are available.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

R.A. Harshman (1970). Foundations of the Parafac procedure: models and conditions for an 'explanatory' multi-mode factor analysis. *UCLA Working Papers in Phonetics 16:1–84*.

See Also

CP, CPfuncrep

Examples

```
data(TV)
TVdata=TV[[1]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
# unconstrained CP solution using two components
# (rational starting point by SVD [start=0])
TVcp <- CPfunc(TVdata, 30, 16, 15, 2, 1, 1, 1, 0, 1e-6, 10000)
# constrained CP solution using two components with orthogonal A-mode
# component matrix (rational starting point by SVD [start=0])
TVcp <- CPfunc(TVdata, 30, 16, 15, 2, 2, 1, 1, 0, 1e-6, 10000)
# constrained CP solution using two components with orthogonal A-mode
# component matrix and zero correlated C-mode component matrix
# (rational starting point by SVD [start=0])
TVcp <- CPfunc(TVdata, 30, 16, 15, 2, 2, 1, 3, 0, 1e-6, 10000)
# unconstrained CP solution using two components
# (random orthonormalized starting point [start=1])
TVcp <- CPfunc(TVdata, 30, 16, 15, 2, 1, 1, 1, 1, 1e-6, 10000)
# unconstrained CP solution using two components (user starting point [start=2])
TVcp <- CPfunc(TVdata, 30, 16, 15, 2, 1, 1, 1, 2, 1e-6, 10000,
 matrix(rnorm(30*2),nrow=30), matrix(rnorm(16*2),nrow=16),
 matrix(rnorm(15*2),nrow=15))
```

CPfuncrep — *Algorithm for the Candecomp/Parafac (CP) model*

Description

Alternating Least Squares algorithm for the minimization of the Candecomp/Parafac loss function.

Usage

```
CPfuncrep(X, n, m, p, r, ort1, ort2, ort3, start, conv, maxit, A, B, C)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
r	Number of extracted components
ort1	Type of constraints on A (1 for no constraints, 2 for orthogonality constraints, 3 for zero correlations constraints)
ort2	Type of constraints on B (1 for no constraints, 2 for orthogonality constraints, 3 for zero correlations constraints)

ort3	Type of constraints on C (1 for no constraints, 2 for orthogonality constraints, 3 for zero correlations constraints)
start	Starting point (0 for starting point of the algorithm from SVD's, 1 for random starting point (orthonormalized component matrices), 2 for user specified components
conv	Convergence criterion
maxit	Maximal number of iterations
A	Optional (necessary if start=2) starting value for A
B	Optional (necessary if start=2) starting value for B
C	Optional (necessary if start=2) starting value for C

Value

A list including the following components:

A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
f	Loss function value
fp	Fit value expressed as a percentage
iter	Number of iterations
tripcos	Minimal triple cosine between two components across three component matrices (to inspect degeneracy)
mintripcos	Minimal triple cosine during the iterative algorithm observed at every 10 iterations (to inspect degeneracy)
ftiter	Matrix containing in each row the function value and the minimal triple cosine at every 10 iterations
cputime	Computation time

Note

The loss function to be minimized is $sum(k)||X(k) - AD(k)B'||^2$, where $D(k)$ is a diagonal matrix holding the k-th row of C.
CPfuncrep is the same as CPfunc except that all printings are suppressed. Thus, CPfuncrep can be helpful for simulation experiments.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

R.A. Harshman (1970). Foundations of the Parafac procedure: models and conditions for an 'explanatory' multi-mode factor analysis. *UCLA Working Papers in Phonetics 16:1–84*.

See Also

CP, CPfunc

Examples

```
data(TV)
TVdata=TV[[1]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
# unconstrained CP solution using two components
# (rational starting point by SVD [start=0])
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 1, 1, 1, 0, 1e-6, 10000)
# constrained CP solution using two components with orthogonal A-mode
# component matrix (rational starting point by SVD [start=0])
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 2, 1, 1, 0, 1e-6, 10000)
# constrained CP solution using two components with orthogonal A-mode
# component matrix and zero correlated C-mode component matrix
# (rational starting point by SVD [start=0])
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 2, 1, 3, 0, 1e-6, 10000)
# unconstrained CP solution using two components
# (random orthonormalized starting point [start=1])
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 1, 1, 1, 1, 1e-6, 10000)
# unconstrained CP solution using two components (user starting point [start=2])
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 1, 1, 1, 2, 1e-6, 10000,
 matrix(rnorm(30*2),nrow=30), matrix(rnorm(16*2),nrow=16),
 matrix(rnorm(15*2),nrow=15))
```

CPrunsFit *Candecomp/Parafac solutions*

Description

Computes all the Candecomp/Parafac solutions (CP) with r (from 1 to maxC) components.

Usage

```
CPrunsFit(X, n, m, p, maxC)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
maxC	Maximum dimensionality for the A-mode

Value

`out` Matrix with columns: number of components for the A-mode, number of components for the B-mode, number of components for the C-mode, goodness of fit (%), total number of components

Note

The structure of out is consistent with Tucker models. In CP, the first and forth columns are sufficient for choosing the optimal number of components.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56:449–470.*

See Also

DimSelector, LineCon, CP

Examples

```
data(TV)
TVdata=TV[[1]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
# Fit values of CP with different numbers of components (from 1 to 5)
FitCP <- CPrunsFit(TVdata, 30, 16, 15, 5)
```

DimSelector *Convex Hull procedure*

Description

Selects among three-mode principal component models of different complexities.

Usage

```
DimSelector(out, n, m, p, model)
```

Arguments

out	Matrix with columns: number of components for the A-mode, number of components for the B-mode, number of components for the C-mode, goodness of fit (%), total number of components
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
model	Kind of model (1 for Candecomp/Parafac, 2 for Tucke3, 3 for Tucker2, 4 for Tucker1

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

E. Ceulemans and H.A.L. Kiers (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology 59:133–150*.
J. Weesie and H. Van Houwelingen (1983). *GEPCAM users' manual (first draft)*. Utrecht, The Netherlands: Institute of Mathematical Statistics, State University of Utrecht.

See Also

LineCon, T3runsApproxFit T2runsApproxFit T1runsFit CPrunsFit

Examples

```
data(Bus)
# Analysis on T3 with different numbers of components (from 1 to 4 for the A-mode,
# from 1 to 3 for the B-mode, from 1 to 5 for the C-mode)
FitT3 <- T3runsApproxFit(Bus,7,5,37,4,4,4)
T3opt <- DimSelector(FitT3,7,5,37,2)
```

---

jointplotgen *Jointplots*

Description

Program for producing jointplots in general.

Usage

```
jointplotgen(K, A, B, C, fixmode, fixunit, laba, labb, labc)
```

Arguments

K	Matricized core array (frontal slices)
A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
fixmode	Mode for which one unit is to be chosen (1 for A-mode, 2 for B-mode, 3 for C-mode)
fixunit	Number of component for which joint plot is desired
laba	Vector of length n containing the labels of the A-mode entities
labb	Vector of length m containing the labels of the B-mode entities
labc	Vector of length p containing the labels of the C-mode entities

Value

fit	Percentage of info for component at hand, explained by two-dimensional plot

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

P.M. Kroonenberg (2008). *Applied Multiway Data Analysis.* Wiley, New Jersey.

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
# <- T3 solution
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 0, 1e-6)
# Joint plot for mode C and component 2
jointplotgen(BusT3$H, BusT3$A, BusT3$B, BusT3$C, 3, 2, laba, labb, labc)
```

Kinship *Kinship terms data*

Description

Three-way proximity data about 15 kinship terms produced by 6 groups of subjects.

Usage

```
data(Kinship)
```

Format

An array of order 15 x 15 x 6.
The A-mode and B-mode entities are the kinship terms (Aunt, Brother, Cousin, Daughter, Father, Granddaughter, Grandfather, Grandmother, Grandson, Mother, Nephew, Niece, Sister, Son, Uncle).
The C-mode entities are groups of subjects (First female, Second female, First male, Second male, Single female, Single male).

Details

The original data have been introduced by Rosenborg \& Kim (1975). The data were collected by asking to 6 groups of subjects to produce a partition of 15 kinship terms. Two groups (Single female and Single male) were composed by 85 male and 85 female college students, respectively, and provided a single partition. Two additional groups of, respectively, 80 male and 80 female students produced two partitions each (First female, Second female, First male, Second male). In fact, they were informed in advance that, after making the first partition, they should give a new partition of the kinship terms using a different basis of meaning. The array contains similarities. For every group of subjects, the numbers of times in which the kinship terms were grouped together are given.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

S. Rosenborg \& M.P. Kim (1975). The method of sorting as a data-gathering procedure in multivariate research. *Multivariate Behavioral Research 10:489–502*.

Examples

```
data(Kinship)
## The labels are in the data array
laba <- dimnames(Kinship)[[1]]
labb <- dimnames(Kinship)[[2]]
```

```
labc <- dimnames(Kinship)[[3]]
## Candecomp/Parafac analysis
## Not run:
CP(Kinship,laba,labb,labc)

## End(Not run)
```

`LineCon` *Middle point location*

Description

Checks whether the middle point is located below or on the line connecting its neighbors.

Usage

```
LineCon(f1, f2, f3, fp1, fp2, fp3)
```

Arguments

`f1`	Goodness-of-fit value for the first point
`f2`	Goodness-of-fit value for the second point
`f3`	Goodness-of-fit value for the third point
`fp1`	Number of effective parameters for the first point
`fp2`	Number of effective parameters for the second point
`fp3`	Number of effective parameters for the third point

Value

`ret`	Value that indicates if the middle point is located below or on the line connecting its neighbors (0 if the middle point is not located below the line connecting its neighbors, 1 if the middle point is not located on the line connecting its neighbors)

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

E. Ceulemans and H.A.L. Kiers (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology 59:133–150*.

J. Weesie and H. Van Houwelingen (1983). *GEPCAM users' manual (first draft)*. Utrecht, The Netherlands: Institute of Mathematical Statistics, State University of Utrecht.

See Also

DimSelector

Examples

```
data(Bus)
# T2-AB with 1 component for the A- and B-mode
FitBusT2AB11 <- T2funcrep(Bus, 7, 5, 37, 1, 1, 37, 0, 1e-6,1)$fp
# T2-AB with 2 components for the A-mode and 1 component for the B-mode
FitBusT2AB21 <- T2funcrep(Bus, 7, 5, 37, 2, 1, 37, 0, 1e-6, 1)$fp
# T2-AB with 1 component for the A-mode and 2 components for the B-mode
# T2-AB with 1 component for the A-mode and 2 components for the B-mode
# FitBusT2AB21>FitBusT2AB12
# T2-AB with 2 components for the A- and B-mode
FitBusT2AB22 <- T2funcrep(Bus, 7, 5, 37, 2, 2, 37, 0, 1e-6,1)$fp
# number of effective parameters n x r1 + m x r2 + r1 x r2 x p - r1^2 - r2^2
nepT2AB11 <- 47
nepT2AB21 <- 88
nepT2AB22 <- 164
ret <- LineCon(FitBusT2AB11, FitBusT2AB21, FitBusT2AB22, nepT2AB11, nepT2AB21, nepT2AB22)
```

---

meaudret *Meaudret data*

Description

Three-way data about six sampling sites along a small French stream (the Meaudret) on which ten biological and chemical variables are collected four times.

Usage

```
data(meaudret)
```

Format

An array of order 6 x 10 x 4.
The A-mode entities are sampling sites (Site1, ..., Site6).
The B-mode entities are biological and chemical variables (Temp, Debi, PH, Cond, Oxyg, Biod, Chem, NH4, NO3, PO4).
The C-mode entities are months (June, August, November, February).

Details

The ranges of the variables are very different and, therefore, normalization of the raw data is recommended. The data have been used by Kiers (1991) in order to show the existing relations among three-way methods.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56:449–470*.

Examples

```
data(meaudret)
## The labels are in the data array
laba <- dimnames(meaudret)[[1]]
labb <- dimnames(meaudret)[[2]]
labc <- dimnames(meaudret)[[3]]
## Candecomp/Parafac analysis
## Not run:
CP(meaudret,laba,labb,labc)
## Tucker3 analysis
T3(meaudret,laba,labb,labc)
## Tucker2 analysis
T2(meaudret,laba,labb,labc)
## Tucker1 analysis
T1(meaudret,laba,labb,labc)

## End(Not run)
```

---

norm3 — *Normalization of a matricized array*

Description

Normalization of a matricized array within one mode (modes indicated by 1,2, or 3) to sum of squares equal to product of size of other modes.

Usage

```
norm3(X, n, m, p, mode)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities

p
: Number of C-mode entities

mode
: Normalization option (1 if X is normalized within A-mode, 2 if X is normalized within B-mode, 3 if X is normalized within C-mode)

Value

Y
: Matrix of order (n x mp) containing the normalized matricized array (frontal slices)

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122*.

See Also

cent3, nrm2

Examples

```
X <- array(c(rnorm(120)), c(6,5,4))
# matricized array
Y <- supermat(X)
# data normalized within A-mode
Z <- norm3(Y$Xa, 6, 5, 4, 1)
apply(Z^2,1,sum)
# data normalized within C-mode
Z <- norm3(Y$Xa, 6, 5, 4, 3)
Z <- permnew(Z, 6, 5, 4)
Z <- permnew(Z, 5, 4, 6)
apply(Z^2, 1, sum)
```

---

normvari *Normalized varimax rotation*

Description

Produces normalized varimax rotated version of A and rotation matrix T.

Usage

```
normvari(A)
```

Arguments

A	Matrix to be to be rotated

Value

A list including the following components:

B	Rotated version of A (B=AT)
T	Rotation matrix
f	Varimax function value

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H. Kaiser (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika 23:187–200*.

See Also

varim

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
Y <- normvari(X)
# normalized varimax rotated version of X
Y$B
# rotation matrix
Y$T
```

---

nrm2 — *Columnwise normalization of a matrix*

Description

Computation of a columnwise normalized version of a matrix.

Usage

```
nrm2(A)
```

Arguments

A	Matrix of any order

Value

N	Matrix columnwise normalized

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

See Also

Cc

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
Y <- nrm2(X)
apply(Y^2, 2, sum)
```

---

ord *Order*

Description

In case of vectors, an ordering of its elements in ascending order is produced; in case of matrices, the ordering in ascending order refers to every column.

Usage

```
ord(X)
```

Arguments

X	Vector or matrix to be ordered

Value

A	Vector or matrix with the elements sorted in ascending order
a	Vector or matrix with the ordering indices

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

Examples

```
# vector
x <- rnorm(6)
y <- ord(x)
# matrix
X <- matrix(rnorm(6*3),ncol=3)
Y <- ord(X)
```

---

orth — *Orthonormalization of a matrix*

Description

Returns an orthonormal basis for the range of `A`.

Usage

```
orth(A)
```

Arguments

`A`	Matrix to be orthogonalized

Value

`Q`	Orthonormal basis for the range of `A`

Note

The columns of `Q` span the same space as the columns of `A` with `t(Q)Q=I`.
The number of columns of `Q` is the rank of `A`.

Author(s)

Maria Antonietta Del Ferraro `<mariaantonietta.delferraro@yahoo.it>`
Henk A.L. Kiers `<h.a.l.kiers@rug.nl>`
Paolo Giordani `<paolo.giordani@uniroma1.it>`

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
Y <- orth(X)
```

`orthmax2` *Orthomax Rotation*

Description

Produces a simultaneous orthomax rotation of two matrices (using one rotation matrix).

Usage

```
orthmax2(A1, A2, gam1, gam2, conv)
```

Arguments

`A1`	First matrix to be rotated with the same number of columns of A2
`A2`	Second matrix to be rotated with the same number of columns of A1
`gam1`	orthmax parameter for `A1`
`gam2`	orthmax parameter for `A2`
`conv`	Optional convergence value (default 1e-6)

Value

A list including the following components:

`B1`	Rotated version of `A1`
`B2`	Rotated version of `A2`
`T`	Rotation matrix
`f`	Orthomax function value

Note

The function to be maximized is $f = sum((A1^2) - 1/m1 * gam1 * sum((sum(A1^2))^2))^2 + sum((A2^2) - 1/m2 * gam2 * sum((sum(A2^2))^2))^2$.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

R. Jennrich (1970). Orthogonal rotation algorithms. *Psychometrika 35:229–235*.

See Also

`varim`

Examples

```
X <- matrix(rnorm(8*3),ncol=3)
Y <- matrix(rnorm(6*3),ncol=3)
orthXY <- orthmax2(X,Y,1,2)
# rotated version of X
orthXY$B1
# rotated version of Y
orthXY$B2
# rotation matrix
orthXY$T
```

---

pcamean *PCA of the mean matrix*

Description

Performs Principal Component Analysis (PCA) of the mean matrix aggregated over mode number indicated by `aggregmode`.

Usage

```
pcamean(X, n, m, p, laba, labb, labc)
```

Arguments

`X`	Matrix (or data.frame coerced to a matrix) of order (`n` x `mp`) containing the matricized array (frontal slices)
`n`	Number of `A`-mode entities
`m`	Number of `B`-mode entities
`p`	Number of `C`-mode entities
`laba`	Optional vector of length `n` containing the labels of the `A`-mode entities
`labb`	Optional vector of length `m` containing the labels of the `B`-mode entities
`labc`	Optional vector of length `p` containing the labels of the `C`-mode entities

Value

A list including the following components:

`Y`	An object of class `matrix` containing the mean matrix
`ev`	A vector containing the eigenvalues of `Y`
`A1`	Component matrix for the `A` mode based on varimax rotation of loadings
`B1`	Component matrix for the `B` mode based on varimax rotation of loadings
`C1`	Component matrix for the `C` mode based on varimax rotation of loadings
`A2`	Component matrix for the `A` mode based on oblique 'HKIC' (Harris-Kaiser Independent Cluster) orthomax rotation of loadings

B2	Component matrix for the B mode based on oblique 'HKIC' (Harris-Kaiser Independent Cluster) orthomax rotation of loadings
C2	Component matrix for the C mode based on oblique 'HKIC' (Harris-Kaiser Independent Cluster) orthomax rotation of loadings

Note

aggregmode denotes the mode over which means are computed (1 for A-mode, 2 for B-mode, 3 for C-mode).
aggregmode is provided interactively.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H. Kaiser (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika 23:187–200*.
C. Harris \& H. Kaiser (1964). Some mathematical notes on three-mode factor analysis. *Psychometrika 29:347–362*.

Examples

```
data(TV)
TVdata=TV[[1]]
labSCALE=TV[[2]]
labPROGRAM=TV[[3]]
labSTUDENT=TV[[4]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
## Not run:
# PCA on the mean matrix
TVpcamean <- pcamean(TVdata, 30, 16, 15, labSTUDENT, labSCALE, labPROGRAM)
# PCA on the mean matrix (when labels are not available)
TVpcamean <- pcamean(TVdata, 30, 16, 15)

## End(Not run)
```

---

pcasup1 *PCASup Analysis*

Description

Computes PCASup analysis for the direction concerning the reduced mode.

Usage

```
pcasup1(X, n, m, p, model)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
model	Tucker1 model choice (1 for T1-A, 2 for T1-B, 3 for T2-C)

Value

A list including the following components:

A	Matrix of the eingenvectors of the supermatrix containing the frontal slices of the array (A-mode)
B	Matrix of the eingenvectors of the supermatrix containing the horizontal slices of the array (B-mode)
C	Matrix of the eingenvectors of the supermatrix containing the lateral slices of the array (C-mode)
la	Vector of the eigenvalues of the supermatrix containing the frontal slices of the array (A-mode)
lb	Vector of the eigenvalues of the supermatrix containing the horizontal slices of the array (B-mode)
lc	Vector of the eigenvalues of the supermatrix containing the lateral slices of the array (C-mode)

Note

pcasup1 computes the Tucker1 solution.
Cumulative sum of eigenvalues and fits from PCAsup applied to the reduced mode are automatically printed.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56: 449–470*.
H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122*.

L.R Tucker (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31: 279–311*.

See Also

T1

Examples

```
data(Bus)
# PCA-sup for T1-B
pcasupBus <- pcasup1(Bus, 7, 5, 37, 2)
```

---

pcasup2 *PCASup Analysis*

Description

Computes PCASup analysis for the directions concerning the reduced modes.

Usage

```
pcasup2(X, n, m, p, model)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
model	Tucker2 model choice (1 for T2-AB, 2 for T2-AC, 3 for T2-BC)

Value

A list including the following components:

A	Matrix of the eingenvectors of the supermatrix containing the frontal slices of the array (A-mode)
B	Matrix of the eingenvectors of the supermatrix containing the horizontal slices of the array (B-mode)
C	Matrix of the eingenvectors of the supermatrix containing the lateral slices of the array (C-mode)
la	Vector of the eigenvalues of the supermatrix containing the frontal slices of the array (A-mode)
lb	Vector of the eigenvalues of the supermatrix containing the horizontal slices of the array (B-mode)
lc	Vector of the eigenvalues of the supermatrix containing the lateral slices of the array (C-mode)

Note

Cumulative sum of eigenvalues and fits from PCAsup applied to the reduced modes are automatically printed.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56: 449–470*.
H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122*.
L.R Tucker (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31: 279–311*.

See Also

T2

Examples

```
data(Bus)
# PCA-sup for T2-AB
pcasupBus <- pcasup2(Bus, 7, 5, 37, 1)
```

---

pcasup3 *PCASup Analysis*

Description

Computes PCASup analysis in all the three directions.

Usage

```
pcasup3(X, n, m, p)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities

Value

A list including the following components:

- `A` Matrix of the eingenvectors of the supermatrix containing the frontal slices of the array (A-mode)
- `B` Matrix of the eingenvectors of the supermatrix containing the horizontal slices of the array (B-mode)
- `C` Matrix of the eingenvectors of the supermatrix containing the lateral slices of the array (C-mode)
- `la` Vector of the eigenvalues of the supermatrix containing the frontal slices of the array (A-mode)
- `lb` Vector of the eigenvalues of the supermatrix containing the horizontal slices of the array (B-mode)
- `lc` Vector of the eigenvalues of the supermatrix containing the lateral slices of the array (C-mode)

Note

pcasup3 computes the Tucker3 solution according to Tucker (1966).
Cumulative sum of eigenvalues and fits from PCAsup applied to the A-, B- and C-modes are automatically printed.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56: 449–470*.
H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122*.
L.R Tucker (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31: 279–311*.

See Also

T3

Examples

```
data(Bus)
## Not run:
# PCA-sup
pcasupBus <- pcasup3(Bus, 7, 5, 37)

## End(Not run)
```

percentile95 *95% percentile intervals*

Description

Computes 2.5% and 97.5% percentiles for all columns of X.

Usage

```
percentile95(X)
```

Arguments

X	Matrix

Value

A list including the following components:

lo	Vector of the 2.5% percentiles of the values in the columns of X
up	Vector of the 97.5% percentiles of the values in the columns of X

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

See Also

bootstrapCP,bootstrapT3

Examples

```
X <- matrix(rnorm(50*3),ncol=3)
perc95X <- percentile95(X)
```

permnew *Permutation of a matricized array*

Description

Permutes the matricized (n x m x p) array X to the matricized array Y of order (m x p x n).

Usage

```
permnew(X,n,m,p)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) containing the matricized array
n	Number of A-mode entities of the array X
m	Number of B-mode entities of the array X
p	Number of C-mode entities of the array X

Value

Y	Matrix containing the permuted matricized array

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122.*

Examples

```
X <- array(c(rnorm(120)),c(6,5,4))
dim(X)
# matricized array
Xa <- supermat(X)$Xa
# matricized X with the A-mode entities in its rows
dim(Xa)
# matricized X with the B-mode entities in its rows
Xb <- permnew(Xa, 6, 5, 4)
dim(Xb)
# matricized X with the C-mode entities in its rows
Xc <- permnew(Xb, 5, 4, 6)
dim(Xc)
```

perms *Permutation*

Description

Gives all the permutations of the first integer numbers.

Usage

```
perms(n)
```

Arguments

n	Integer

Value

z	Matrix containing in its rows all the permutation of the first n integer numbers

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2004). Bootstrap confidence intervals for three-way methods. *Journal of Chemometrics 18:22–36*.

Examples

```
P <- perms(4)
```

phi *Phi coefficient*

Description

Computes the phi coefficients among columns of two matrices.

Usage

```
phi(a,b)
```

Arguments

a	Vector or matrix of the same order of b
b	Vector or matrix of the same order of a

Value

p	Matrix containing the phi coefficients

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

L.R Tucker (1951). A method for synthesis of factor analysis studies. *Personnel Research Section Report No. 984*. Department of the Army, Washington, DC.

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
Y <- matrix(rnorm(6*3),ncol=3)
P <- phi(X,Y)
```

---

rarray — *Array reconstruction*

Description

Produces an array starting from its matricization with all the frontal slices of the array next to each other.

Usage

```
rarray(Xa, n, m, p)
```

Arguments

Xa	Matrix (or data.frame coerced to a matrix) containing the elements of the frontal slices of an array
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities

Value

X	Array leading to Xa

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122*.

Examples

```
# matricized array (frontal slice)
Xa <- matrix(1:8,nrow=2)
X <- rarray(Xa, 2, 2, 2)
# original array
X
```

---

renormsolCP *Scaling of the Candecomp/Parafac solution*

Description

Scales the Candecomp/Parafac solution producing two component matrices normalized to unit sum of squares (and compensating this scaling in the remaining component matrix).

Usage

```
renormsolCP(A, B, C, mode)
```

Arguments

A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
mode	Scaling option (1 if scaling for B- and C-modes, 2 if scaling for A- and C-modes, 3 if scaling for A- and B-modes)

Value

A list including the following components:

A	Component matrix for the A-mode after normalization
B	Component matrix for the B-mode after normalization
C	Component matrix for the C-mode after normalization

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

Examples

```
data(TV)
TVdata=TV[[1]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
# CP solution
TVcp <- CPfuncrep(TVdata, 30, 16, 15, 2, 1, 1, 1, 0, 1e-6, 10000)
# sums of squares of A, B and C
sum(TVcp$A^2)
sum(TVcp$B^2)
sum(TVcp$C^2)
# Renormalization by scaling B- and C-modes
TVcpScalBC <- renormsolCP(TVcp$A, TVcp$B, TVcp$C, 1)
# sums of squares of A, B and C after renormalization
sum(TVcpScalBC$A^2)
sum(TVcpScalBC$B^2)
sum(TVcpScalBC$C^2)
```

---

renormsolT3 *Renormalization of the Tucker3 (and Tucker2) solution*

Description

Renormalizes the Tucker3 solution producing a core normalized to unit sum of squares (and compensating the core normalization in the component matrices).

Usage

```
renormsolT3(A, B, C, G, mode)
```

Arguments

A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
G	Matricized core array (frontal slices)
mode	Renormalization option (1 if renormalization with respect to A-mode, 2 if renormalization with respect to B-mode, 3 if renormalization with respect to C-mode)

Value

A list including the following components:

A	Component matrix for the A-mode after normalization of the core
B	Component matrix for the B-mode after normalization of the core
C	Component matrix for the C-mode after normalization of the core
H	Normalized matricized core array (frontal slices)

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
# T3 solution
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 0, 1e-6)
# sums of squares of A and core
sum(BusT3$A^2)
sum(BusT3$H^2)
# Renormalization with respect to the A-mode
BusT3rA <- renormsolT3(BusT3$A, BusT3$B, BusT3$C, BusT3$H,1)
# sums of squares of A and core after renormalization
sum(BusT3rA$A^2)
sum(BusT3rA$H^2)
```

---

splithalfCP *Split-Half Analysis*

Description

Performs split-half analysis for Candecomp/Parafac.

Usage

```
splithalfCP(X, n, m, p, r, centopt, normopt, scaleopt, addanal, conv,
 maxit, ort1, ort2, ort3, laba, labb, labc)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
r	Number of extracted components
centopt	Centering option (see cent3)
normopt	Normalization option (see norm3)
scaleopt	Scaling option (see renormsolCP)
addanal	Number of additional runs
conv	Convergence criterion
maxit	Maximal number of iterations
ort1	Type of constraints on A (see CP)
ort2	Type of constraints on B (see CP)
ort3	Type of constraints on C (see CP)
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

Afull	Component matrix for the A-mode (full data)
As1	Component matrix for the A-mode (split n.1)
As2	Component matrix for the A-mode (split n.2)
Bfull	Component matrix for the B-mode (full data)
Bs1	Component matrix for the B-mode (split n.1)
Bs2	Component matrix for the B-mode (split n.2)
Cfull	Component matrix for the C-mode (full data)
Cs1	Component matrix for the C-mode (split n.1)
Cs2	Component matrix for the C-mode (split n.2)

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

P.M. Kroonenberg (2008). *Applied Multiway Data Analysis.* Wiley, New Jersey.

See Also

CP

Examples

```
data(TV)
TVdata=TV[[1]]
labSCALE=TV[[2]]
labPROGRAM=TV[[3]]
labSTUDENT=TV[[4]]
# permutation of the modes so that the A-mode refers to students
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
## Not run:
# Split-half analysis on CP solution
splitCP <- splithalfCP(TVdata, 30, 16, 15, 2, 0, 0, 0, 5, 1e-6, 10000, 1, 1, 1,
 labSTUDENT, labSCALE, labPROGRAM)
# Split-half analysis on CP solution (when labels are not available)
splitCP <- splithalfCP(TVdata, 30, 16, 15, 2, 0, 0, 0, 5, 1e-6, 10000, 1, 1, 1)

## End(Not run)
```

---

splithalfT3 *Split-Half Analysis*

Description

Performs split-half analysis for Tucker3.

Usage

```
splithalfT3(X, n, m, p, r1, r2, r3, centopt, normopt, renormmode,
 wa_rel, wb_rel, wc_rel, addanal, conv, laba, labb, labc)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
r1	Number of extracted components for the A-mode
r2	Number of extracted components for the B-mode
r3	Number of extracted components for the C-mode
centopt	Centering option (see cent3)
normopt	Normalization option (see norm3)

renormmode	Renormalization option (see renormsolT3)
wa_rel	Relative weight for simplicity of A-mode
wb_rel	Relative weight for simplicity of B-mode
wc_rel	Relative weight for simplicity of C-mode
addanal	Number of additional runs
conv	Convergence criterion
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

Afull	Component matrix for the A-mode (full data)
As1	Component matrix for the A-mode (split n.1)
As2	Component matrix for the A-mode (split n.2)
Bfull	Component matrix for the B-mode (full data)
Bs1	Component matrix for the B-mode (split n.1)
Bs2	Component matrix for the B-mode (split n.2)
Cfull	Component matrix for the C-mode (full data)
Cs1	Component matrix for the C-mode (split n.1)
Cs2	Component matrix for the C-mode (split n.2)
Kfull	Matricized core array (frontal slices) (full data)
Ks1	Matricized core array (frontal slices) (split n.1)
Ks2	Matricized core array (frontal slices) (split n.2)
Kss1	Matricized core array (frontal slices) (using full data solutions for A,B and C for split n.1)
Kss2	Matricized core array (frontal slices) (using full data solutions for A,B and C for split n.2)

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

P.M. Kroonenberg (2008). *Applied Multiway Data Analysis*. Wiley, New Jersey.

See Also

T3

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5],1,1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)],3,8)
## Not run:
# Split-half analysis on T3 solution
splitT3 <- splithalfT3(Bus, 7, 5, 37, 2, 2, 2, 0, 0, 0, 3, 3, 0, 5, 1e-6,
 laba, labb, labc)
# Split-half analysis on T3 solution (when labels are not available)
splitT3 <- splithalfT3(Bus, 7, 5, 37, 2, 2, 2, 0, 0, 0, 3, 3, 0, 5, 1e-6)

## End(Not run)
```

SUM *Summary*

Description

Summary of the elements of a matrix.

Usage

```
SUM(A)
```

Arguments

A	Matrix or data.frame (coerced to a matrix)

Value

A list including the following components:

row	Vector containing the sum of squares of every row
col	Vector containing the sum of squares of every column
mr	Vector containing the mean of every row
mc	Vector containing the mean of every column
minc	Vector containing the minimum of every column
maxc	Vector containing the maximum of every for column
valueMinr	Vector containing the columns corresponding to the minimum values of every row
valueMinc	Vector containing the rows corresponding to the minimum values of every column
valueMaxr	Vector containing the columns corresponding to the maximum values of every row

valueMaxc	Vector containing the rows corresponding to the maximum values of every column
ssq	Sum of squares of the matrix
cumsumr	Matrix containing the cumulative sums of every row
cumsumc	Matrix containing the cumulative sums of every column

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
summary <- SUM(X)
```

---

supermat *Matrix unfolding*

Description

Produces matricizations of a three-way array into matrices denoted as super-matrices.

Usage

```
supermat(X)
```

Arguments

X	Array to be unfolded

Value

A list including the following components:

Xa	Super-matrix with B-mode entities nested within C-mode entities (all the frontal slices of the array next to each other)
Xb	Super-matrix with C-mode entities nested within A-mode entities (all the horizontal slices of the array next to each other)
Xc	Super-matrix with A-mode entities nested within B-mode entities (all the lateral slices of the array next to each other)

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics 14:105–122*.

Examples

```
# array (2x2x2) with integers from 1 to 8
X <- array(1:8,c(2,2,2))
Y <- supermat(X)
# matricized arrays
Y$Xa
Y$Xb
Y$Xc
```

T1 *Interactive Tucker1 analysis*

Description

Detects the underlying structure of a three-way array according to the Tucker1 (T1) model.

Usage

```
T1(dati, laba, labb, labc)
```

Arguments

`dati`	Array of order `n` by `m` by `p` or matrix or data.frame of order (`n x mp`) containing the matricized array (frontal slices)
`laba`	Optional vector of length `n` containing the labels of the `A`-mode entities
`labb`	Optional vector of length `m` containing the labels of the `B`-mode entities
`labc`	Optional vector of length `p` containing the labels of the `C`-mode entities

Value

A list including the following components:

`A`	Component matrix for the `A`-mode
`B`	Component matrix for the `B`-mode
`C`	Component matrix for the `C`-mode
`core`	Matricized core array (frontal slices)
`fit`	Fit value expressed as a percentage
`fitA`	Fit contributions for the `A`-mode entities (see `T3fitpartitioning`)
`fitB`	Fit contributions for the `B`-mode entities (see `T3fitpartitioning`)

fitC	Fit contributions for the C-mode entities (see T3fitpartitioning)
laba	Vector of length n containing the labels of the A-mode entities
labb	Vector of length m containing the labels of the B-mode entities
labc	Vector of length P containing the labels of the C-mode entities
Xprep	Matrix of order (n x mp) containing the matricized array (frontal slices) after preprocessing used for the analysis

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

P. Giordani, H.A.L. Kiers, M.A. Del Ferraro (2014). Three-way component analysis using the R package ThreeWay. *Journal of Statistical Software 57(7):1–23*. http://www.jstatsoft.org/v57/i07/.
P.M. Kroonenberg (2008). *Applied Multiway Data Analysis*. Wiley, New Jersey.
L.R Tucker (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31:279–311*.

See Also

CP,T3,T2

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5],1,1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)],3,8)
## Not run:
# interactive T1 analysis
BusT1 <- T1(Bus, laba, labb, labc)
# interactive T1 analysis (when labels are not available)
BusT1 <- T1(Bus)

## End(Not run)
```

T1runsFit *Tucker1 solutions*

Description

Computes all the Tucker1 solutions using PCASup results with r1 (from 1 to maxa, if A-mode reduced), r2 (from 1 to maxb, if B-mode reduced) and r3 (from 1 to maxc, if C-mode reduced) components.

Usage

```
T1runsFit(X, n, m, p, maxa, maxb, maxc, model)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
maxa	Maximum dimensionality for the A-mode
maxb	Maximum dimensionality for the B-mode
maxc	Maximum dimensionality for the C-mode
model	Tucker1 model choice (1 for T1-A, 2 for T1-B, 3 for T2-C)

Value

out	Matrix with columns: number of components for the A-mode, number of components for the B-mode, number of components for the C-mode, goodness of fit (%), total number of components

Note

Cumulative sum of eigenvalues and fits from PCAsup applied to the reduced mode are automatically printed.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56:449–470*.

See Also

DimSelector, LineCon, pcasup1, T1

Examples

```
data(Bus)
# Fit values of T1-A with different numbers of components (from 1 to 5)
FitT1 <- T1runsFit(Bus, 7, 5, 37, 5, 5, 37, 1)
```

T2 — *Interactive Tucker2 analysis*

Description

Detects the underlying structure of a three-way array according to the Tucker2 (T2) model.

Usage

```
T2(dati, laba, labb, labc)
```

Arguments

dati	Array of order n x m x p or matrix or data.frame of order (n x mp) containing the matricized array (frontal slices)
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

A list including the following components:

A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
core	Matricized core array (frontal slices)
fit	Fit value expressed as a percentage
fitValues	Fit values expressed as a percentage upon convergence for all the runs of the CP algorithm (see T2func)
funcValues	Function values upon convergence for all the runs of the CP algorithm (see T2func)
cputime	Computation times for all the runs of the CP algorithm (see T2func)
iter	Numbers of iterations upon convergence for all the runs of the CP algorithm (see T2func)

fitA	Fit contributions for the A-mode entities (see T3fitpartitioning)
fitB	Fit contributions for the B-mode entities (see T3fitpartitioning)
fitC	Fit contributions for the C-mode entities (see T3fitpartitioning)
fitAB	Fit contributions for the A-and mode B component combinations (see T3fitpartitioning)
fitAC	Fit contributions for the A-and mode C component combinations (see T3fitpartitioning)
fitBC	Fit contributions for the B-and mode C component combinations (see T3fitpartitioning)
laba	Vector of length n containing the labels of the A-mode entities
labb	Vector of length m containing the labels of the B-mode entities
labc	Vector of length P containing the labels of the C-mode entities
Xprep	Matrix of order (n x mp) containing the matricized array (frontal slices) after preprocessing used for the analysis

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

P. Giordani, H.A.L. Kiers, M.A. Del Ferraro (2014). Three-way component analysis using the R package ThreeWay. *Journal of Statistical Software 57(7):1–23*. http://www.jstatsoft.org/v57/i07/.
P.M. Kroonenberg (2008). *Applied Multiway Data Analysis*. Wiley, New Jersey.
L.R Tucker (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31:279–311*.

See Also

CP,T3,T1

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
## Not run:
# interactive T2 analysis
BusT2 <- T2(Bus, laba, labb, labc)
# interactive T2 analysis (when labels are not available)
BusT2 <- T2(Bus)

## End(Not run)
```

T2func — *Algorithm for the Tucker2 model*

Description

Alternating Least Squares algorithm for the minimization of the Tucker2 loss function.

Usage

```
T2func(X, n, m, p, r1, r2, r3, start, conv, model, A, B, C, H)
```

Arguments

`X`	Matrix (or data.frame coerced to a matrix) of order (`n` x `mp`) containing the matricized array (frontal slices)
`n`	Number of `A`-mode entities
`m`	Number of `B`-mode entities
`p`	Number of `C`-mode entities
`r1`	Number of extracted components for the `A`-mode
`r2`	Number of extracted components for the `B`-mode
`r3`	Number of extracted components for the `C`-mode
`start`	Starting point: 0 starting point of the algorithm from generalized eigenvalue decomposition, 1 random starting point (orthonormalized component matrices), 2 if users specified component matrices
`conv`	Convergence criterion
`model`	Tucker2 model choice (1 for T2-AB, 2 for T2-AC, 3 for T2-BC)
`A`	Optional (necessary if start=2) starting value for `A`
`B`	Optional (necessary if start=2) starting value for `B`
`C`	Optional (necessary if start=2) starting value for `C`
`H`	Optional (necessary if start=2) starting value for the matricized core array (frontal slices)

Value

A list including the following components:

`A`	Orthonormal component matrix for the `A`-mode
`B`	Orthonormal component matrix for the `B`-mode
`C`	Orthonormal component matrix for the `C`-mode
`H`	Matricized core array (frontal slices)
`f`	Loss function value

fp	Fit percentage
iter	Number of iterations
cputime	Computation time
La	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for A-mode
Lb	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for B-mode
Lc	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for C-mode

Note

The loss function to be minimized is $||X_A - AG_A kron(C', B')||^2$ where X_A and G_A denote the matricized (frontal slices) data array and core array, respectively, and kron stands for the Kronecker product.
T2func is the same as T2funcrep except that all printings are available.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers, P.M. Kroonenberg \& J.M.F. ten Berge (1992). An efficient algorithm for TUCKALS3 on data with large numbers of observation units. *Psychometrika 57:415–422*.
P.M. Kroonenberg and J. de Leeuw (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika 45:69–97*.

See Also

T2, T2funcrep

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
# T2-AB solution using two components for the A- and B-modes
# (rational starting point by SVD [start=0])
BusT2 <- T2func(Bus, 7, 5, 37, 2, 2, 37, 0, 1e-6, 1)
# T2-AC solution using two components for for the A- and C-modes
# (random orthonormalized starting point [start=1])
BusT2 <- T2func(Bus, 7, 5, 37, 2, 5, 2, 1, 1e-6, 2)
# T2-BC solution using two components for the B- and C- modes
# (user starting point [start=2])
```

```
BusT2 <- T2func(Bus, 7, 5, 37, 7, 2, 2, 1, 1e-6, 3, diag(7),
 matrix(rnorm(5*2),nrow=5), matrix(rnorm(37*2),nrow=37),
 matrix(rnorm(7*4),nrow=7))
```

T2funcrep *Algorithm for the Tucker2 model*

Description

Alternating Least Squares algorithm for the minimization of the Tucker2 loss function.

Usage

```
T2funcrep(X, n, m, p, r1, r2, r3, start, conv, model, A, B, C, H)
```

Arguments

`X`	Matrix (or data.frame coerced to a matrix) of order (`n` x `mp`) containing the matricized array (frontal slices)
`n`	Number of `A`-mode entities
`m`	Number of `B`-mode entities
`p`	Number of `C`-mode entities
`r1`	Number of extracted components for the `A`-mode
`r2`	Number of extracted components for the `B`-mode
`r3`	Number of extracted components for the `C`-mode
`start`	Starting point: 0 starting point of the algorithm from generalized eigenvalue decomposition, 1 random starting point (orthonormalized component matrices), 2 if users specified component matrices
`conv`	Convergence criterion
`model`	Tucker2 model choice (1 for T2-AB, 2 for T2-AC, 3 for T2-BC)
`A`	Optional (necessary if start=2) starting value for `A`
`B`	Optional (necessary if start=2) starting value for `B`
`C`	Optional (necessary if start=2) starting value for `C`
`H`	Optional (necessary if start=2) starting value for the matricized core array (frontal slices)

Value

A list including the following components:

A	Orthonormal component matrix for the A-mode
B	Orthonormal component matrix for the B-mode
C	Orthonormal component matrix for the C-mode
H	Matricized core array (frontal slices)
f	Loss function value
fp	Fit percentage
iter	Number of iterations
cputime	Computation time
La	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for A-mode
Lb	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for B-mode
Lc	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for C-mode

Note

The loss function to be minimized is $||X_A - AG_A kron(C', B')||^2$ where X_A and G_A denote the matricized (frontal slices) data array and core array, respectively, and kron stands for the Kronecker product.
T2funcrep is the same as T2func except that all printings are suppressed. Thus, T2funcrep can be helpful for simulation experiments.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers, P.M. Kroonenberg \& J.M.F. ten Berge (1992). An efficient algorithm for TUCKALS3 on data with large numbers of observation units. *Psychometrika 57:415–422*.
P.M. Kroonenberg and J. de Leeuw (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika 45:69–97*.

See Also

T2, T2func

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
# T2-AB solution using two components for the A- and B-modes
# (rational starting point by SVD [start=0])
BusT2 <- T2funcrep(Bus, 7, 5, 37, 2, 2, 37, 0, 1e-6,1)
# T2-AC solution using two components for for the A- and C-modes
# (random orthonormalized starting point [start=1])
BusT2 <- T2funcrep(Bus, 7, 5, 37, 2, 5, 2, 1, 1e-6, 2)
# T2-BC solution using two components for the B- and C- modes
# (user starting point [start=2])
BusT2 <- T2funcrep(Bus, 7, 5, 37, 7, 2, 2, 1, 1e-6, 3, diag(7),
 matrix(rnorm(5*2),nrow=5), matrix(rnorm(37*2),nrow=37),
 matrix(rnorm(7*4),nrow=7))
```

T2runsApproxFit *Approximated Tucker2 solutions*

Description

Computes all the approximated Tucker2 solutions using PCASup results with r1 (from 1 to maxa, if A-mode reduced), r2 (from 1 to maxb, if B-mode reduced) and r3 (from 1 to maxc, if C-mode reduced) components.

Usage

```
T2runsApproxFit(X, n, m, p, maxa, maxb, maxc, model)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
maxa	Maximum dimensionality for the A-mode
maxb	Maximum dimensionality for the B-mode
maxc	Maximum dimensionality for the C-mode
model	Tucker2 model choice (1 for T2-AB, 2 for T2-AC, 3 for T2-BC)

Value

out	Matrix with columns: number of components for the A-mode, number of components for the B-mode, number of components for the C-mode, goodness of fit (%), total number of components

Note

Cumulative sum of eigenvalues and fits from PCAsup applied to the reduced modes are automatically printed.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56:449–470.*

See Also

DimSelector, LineCon, pcasup2, T2

Examples

```
data(Bus)
# Fit values of T2-AB with different numbers of components
# (from 1 to 3 for the B-mode, from 1 to 5 for the C-mode)
FitT2 <- T2runsApproxFit(Bus, 7, 5, 37, 7, 3, 5, 3)
```

T3 *Interactive Tucker3 analysis*

Description

Detects the underlying structure of a three-way array according to the Tucker3 (T3) model.

Usage

```
T3(data, laba, labb, labc)
```

Arguments

data	Array of order n x m x p or matrix or data.frame of order (n x mp) containing the matricized array (frontal slices)
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

A list including the following components:

A	Component matrix for the A-mode
B	Component matrix for the B-mode
C	Component matrix for the C-mode
core	Matricized core array (frontal slices)
fit	Fit value expressed as a percentage
fitValues	Fit values expressed as a percentage upon convergence for all the runs of the CP algorithm (see T3func)
funcValues	Function values upon convergence for all the runs of the CP algorithm (see T3func)
cputime	Computation times for all the runs of the CP algorithm (see T3func)
iter	Numbers of iterations upon convergence for all the runs of the CP algorithm (see T3func)
fitA	Fit contributions for the A-mode entities (see T3fitpartitioning)
fitB	Fit contributions for the B-mode entities (see T3fitpartitioning)
fitC	Fit contributions for the C-mode entities (see T3fitpartitioning)
fitAB	Fit contributions for the A-and mode B component combinations (see T3fitpartitioning)
fitAC	Fit contributions for the A-and mode C component combinations (see T3fitpartitioning)
fitBC	Fit contributions for the B-and mode C component combinations (see T3fitpartitioning)
Bint	Bootstrap percentile interval of every element of B (see bootstrapT3)
Cint	Bootstrap percentile interval of every element of C (see bootstrapT3)
Kint	Bootstrap percentile interval of every element of core (see bootstrapT3)
fpint	Bootstrap percentile interval for the goodness of fit index expressed as a percentage (see bootstrapT3)
Afull	Component matrix for the A-mode (full data) from split-half analysis (see splithalfT3)
As1	Component matrix for the A-mode (split n.1) from split-half analysis (see splithalfT3)
As2	Component matrix for the A-mode (split n.2) from split-half analysis (see splithalfT3)
Bfull	Component matrix for the B-mode (full data) from split-half analysis (see splithalfT3)
Bs1	Component matrix for the B-mode (split n.1) from split-half analysis (see splithalfT3)
Bs2	Component matrix for the B-mode (split n.2) from split-half analysis (see splithalfT3)
Cfull	Component matrix for the C-mode (full data) from split-half analysis (see splithalfT3)
Cs1	Component matrix for the C-mode (split n.1) from split-half analysis (see splithalfT3)
Cs2	Component matrix for the C-mode (split n.2) from split-half analysis (see splithalfT3)
Kfull	Matricized core array (frontal slices) (full data) from split-half analysis (see splithalfT3)
Ks1	Matricized core array (frontal slices) (split n.1) from split-half analysis (see splithalfT3)

Ks2	Matricized core array (frontal slices) (split n.2) from split-half analysis (see splithalfT3)
Kss1	Matricized core array (frontal slices) (using full data solutions for A,B and C for split n.1) from split-half analysis (see splithalfT3)
Kss2	Matricized core array (frontal slices) (using full data solutions for A,B and C for split n.2) from split-half analysis (see splithalfT3)
Aplot	Coordinates for plots of the A-mode entities
Bplot	Coordinates for plots of the B-mode entities
Cplot	Coordinates for plots of the C-mode entities
CBplot	Coordinates for plots of the C and B-mode entities using the A-mode projected in it as axes (to be added in plot, i.e. coordinates in ($CBplot,$A))
ACplot	Coordinates for plots of the A and C-mode entities using the B-mode projected in it as axes (to be added in plot, i.e. coordinates in ($ACplot,$B))
BAplot	Coordinates for plots of the B and A-mode entities using the C-mode projected in it as axes (to be added in plot, i.e. coordinates in ($BAplot,$C))
A1	Component matrix for the A-mode from Principal Component Analysis of mean values (see pcamean)
B1	Component matrix for the B-mode from Principal Component Analysis of mean values (see pcamean)
C1	Component matrix for the C-mode from Principal Component Analysis of mean values (see pcamean)
A2	Component matrix for the A-mode from Principal Component Analysis of mean values (see pcamean)
B2	Component matrix for the B-mode from Principal Component Analysis of mean values (see pcamean)
C2	Component matrix for the C-mode from Principal Component Analysis of mean values (see pcamean)
laba	Vector of length n containing the labels of the A-mode entities
labb	Vector of length m containing the labels of the B-mode entities
labc	Vector of length P containing the labels of the C-mode entities
Xprep	Matrix of order (n x mp) containing the matricized array (frontal slices) after preprocessing used for the analysis

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

P. Giordani, H.A.L. Kiers, M.A. Del Ferraro (2014). Three-way component analysis using the R package ThreeWay. *Journal of Statistical Software 57(7):1–23*. http://www.jstatsoft.org/v57/i07/.
P.M. Kroonenberg (2008). *Applied Multiway Data Analysis*. Wiley, New Jersey.
L.R Tucker (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31:279–311*.

See Also

CP,T2,T1

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5],1,1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)],3,8)
## Not run:
# interactive T3 analysis
BusT3 <- T3(Bus, laba, labb, labc)
# interactive T3 analysis (when labels are not available)
BusT3 <- T3(Bus)

## End(Not run)
```

T3dimensionalityplot *Plot fit of Tucker3*

Description

Plots fits against numbers of dimensions, with PQR as labels and fits against number of effective paramaters.

Usage

```
T3dimensionalityplot(A, n, m, p)
```

Arguments

A	Matrix with columns: number of components for the A-mode, number of components for the B-mode, number of components for the C-mode, goodness of fit (%), total number of components
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities

Note

A is usually the output of DimSelector.
The number of effective parameters in a Candecomp/Parafac analysis is discussed in Weesie and Van Houwelingen (1983).

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

E. Ceulemans \& H.A.L. Kiers (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology 59:133–150*.
J. Weesie and H. Van Houwelingen (1983). *GEPCAM users' manual (first draft)*. Utrecht, The Netherlands: Institute of Mathematical Statistics, State University of Utrecht.

See Also

T3, DimSelector

Examples

```
data(Bus)
# Fit values of T3 with different numbers of components (from 1 to 4 for the A-mode,
# from 1 to 3 for the B-mode, from 1 to 5 for the C-mode)
FitT3 <- T3runsApproxFit(Bus,7,5,37,4,3,5)
T3dimensionalityplot(FitT3,7,5,37)
```

T3fitpartitioning *Fit of each entity per mode*

Description

Computation of fit contributions by combinations of modes in case of 'renormalization'.

Usage

```
T3fitpartitioning(Xprep, n, m, p, AS, BT, CU, K, renormmode, laba, labb, labc)
```

Arguments

Xprep	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
AS	Component matrix for the A-mode
BT	Component matrix for the B-mode
CU	Component matrix for the C-mode
K	Matricized core array (frontal slices)
renormmode	Renormalization option (0 for no renormalization, 1 for fit contribution to total fit of each B- and C-mode component combination, 2 for fit contribution to total fit of each A- and C-mode component combination, 3 for fit contribution to total fit of each A- and B-mode component combination)
laba	Optional vector of length n containing the labels of the A-mode entities
labb	Optional vector of length m containing the labels of the B-mode entities
labc	Optional vector of length p containing the labels of the C-mode entities

Value

A list including the following components:

fitA	Fit contribution for the A-mode entities
fitB	Fit contribution for the B-mode entities
fitC	Fit contribution for the C-mode entities
ABcontr	Contribution to the goodness of fit contributions by combinations of A- and B-modes in case of ‘renormalization’
BCcontr	Contribution to the goodness of fit contributions by combinations of B- and C-modes in case of ‘renormalization’
ACcontr	Contribution to the goodness of fit contributions by combinations of A- and C-modes in case of ‘renormalization’

Note

The computation of the fit contributions by combinations of modes is done in case of ‘renormalization’.
In Tucker1, renormmode must be equal to 0.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

See Also

T3, T2, T1

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
# T3 solution
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 0, 1e-6)
# Fitpartitioning of the T3 solution
FitT3 <- T3fitpartitioning(Bus, 7, 5, 37, BusT3$A, BusT3$B, BusT3$C, BusT3$H, 0,
 laba, labb, labc)
# Fitpartitioning of the T3 solution (when labels are not available)
FitT3 <- T3fitpartitioning(Bus, 7, 5, 37, BusT3$A, BusT3$B, BusT3$C, BusT3$H, 0)
```

T3func *Algorithm for the Tucker3 model*

Description

Alternating Least Squares algorithm for the minimization of the Tucker3 loss function.

Usage

```
T3func(X, n, m, p, r1, r2, r3, start, conv, A, B, C, H)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
r1	Number of extracted components for the A-mode
r2	Number of extracted components for the B-mode
r3	Number of extracted components for the C-mode
start	Starting point (0 starting point of the algorithm from generalized eigenvalue decomposition, 1 random starting point (orthonormalized component matrices), 2 if users specified component matrices
conv	Convergence criterion
A	Optional (necessary if start=2) starting value for A

B Optional (necessary if start=2) starting value for B

C Optional (necessary if start=2) starting value for C

H Optional (necessary if start=2) starting value for the matricized core array (frontal slices)

Value

A list including the following components:

A Orthonormal component matrix for the A-mode

B Orthonormal component matrix for the B-mode

C Orthonormal component matrix for the C-mode

H Matricized core array (frontal slices)

f Loss function value

fp Fit percentage

iter Number of iterations

cputime Computation time

La Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for A-mode

Lb Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for B-mode

Lc Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for C-mode

Note

The loss function to be minimized is $||X_A - AG_A kron(C', B')||^2$ where X_A and G_A denote the matricized (frontal slices) data array and core array, respectively, and kron stands for the Kronecker product.
T3func is the same as T3funcrep except that all printings are available.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers, P.M. Kroonenberg \& J.M.F. ten Berge (1992). An efficient algorithm for TUCKALS3 on data with large numbers of observation units. *Psychometrika 57:415–422*.
P.M. Kroonenberg \& J. de Leeuw (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika 45:69–97*.

See Also

T3, T3funcrep

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
# T3 solution using two components for all the modes
# (rational starting point by SVD [start=0])
BusT3 <- T3func(Bus, 7, 5, 37, 2, 2, 2, 0, 1e-6)
# T3 solution using two components for all the modes
# (random orthonormalized starting point [start=1])
BusT3 <- T3func(Bus, 7, 5, 37, 2, 2, 2, 1, 1e-6)
# T3 solution using two components for all the modes
# (user starting point [start=2])
BusT3 <- T3func(Bus, 7, 5, 37, 2, 2, 2, 1, 1e-6, matrix(rnorm(7*2),nrow=7),
 matrix(rnorm(5*2),nrow=5), matrix(rnorm(37*2),nrow=37),
 matrix(rnorm(2*4),nrow=2))
```

T3funcrep *Algorithm for the Tucker3 model*

Description

Alternating Least Squares algorithm for the minimization of the Tucker3 loss function.

Usage

```
T3funcrep(X, n, m, p, r1, r2, r3, start, conv, A, B, C, H)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
r1	Number of extracted components for the A-mode
r2	Number of extracted components for the B-mode
r3	Number of extracted components for the C-mode
start	Starting point (0 starting point of the algorithm from generalized eigenvalue decomposition, 1 random starting point (orthonormalized component matrices), 2 if users specified component matrices
conv	Convergence criterion
A	Optional (necessary if start=2) starting value for A

B	Optional (necessary if start=2) starting value for B
C	Optional (necessary if start=2) starting value for C
H	Optional (necessary if start=2) starting value for the matricized core array (frontal slices)

Value

A list including the following components:

A	Orthonormal component matrix for the A-mode
B	Orthonormal component matrix for the B-mode
C	Orthonormal component matrix for the C-mode
H	Matricized core array (frontal slices)
f	Loss function value
fp	Fit percentage
iter	Number of iterations
cputime	Computation time
La	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for A-mode
Lb	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for B-mode
Lc	Matrix which should be diagonal, and if so, contain 'intrinsic eigenvalues' for C-mode

Note

The loss function to be minimized is $||X_A - AG_A kron(C', B')||^2$ where X_A and G_A denote the matricized (frontal slices) data array and core array, respectively, and kron stands for the Kronecker product.
T3funcrep is the same as T3func except that all printings are suppressed. Thus, T3funcrep can be helpful for simulation experiments.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers, P.M. Kroonenberg \& J.M.F. ten Berge (1992). An efficient algorithm for TUCKALS3 on data with large numbers of observation units. *Psychometrika 57:415–422*.
P.M. Kroonenberg \& J. de Leeuw (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika 45:69–97*.

See Also

T3, T3func

Examples

```
data(Bus)
# labels for Bus data
laba <- rownames(Bus)
labb <- substr(colnames(Bus)[1:5], 1, 1)
labc <- substr(colnames(Bus)[seq(1,ncol(Bus),5)], 3, 8)
# T3 solution using two components for all the modes
# (rational starting point by SVD [start=0])
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 0, 1e-6)
# T3 solution using two components for all the modes
# (random orthonormalized starting point [start=1])
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 1, 1e-6)
# T3 solution using two components for all the modes
# (user starting point [start=2])
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 1, 1e-6, matrix(rnorm(7*2),nrow=7),
 matrix(rnorm(5*2),nrow=5), matrix(rnorm(37*2),nrow=37),
 matrix(rnorm(2*4),nrow=2))
```

T3runsApproxFit *Approximated Tucker3 solutions*

Description

Computes all the approximated Tcker3 solutions using PCASup results with `r1` (from 1 to `maxa`), `r2` (from 1 to `maxb`) and `r3` (from 1 to `maxc`) components.

Usage

```
T3runsApproxFit(X, n, m, p, maxa, maxb, maxc)
```

Arguments

X	Matrix (or data.frame coerced to a matrix) of order (n x mp) containing the matricized array (frontal slices)
n	Number of A-mode entities
m	Number of B-mode entities
p	Number of C-mode entities
maxa	Maximum dimensionality for the A-mode
maxb	Maximum dimensionality for the B-mode
maxc	Maximum dimensionality for the C-mode

Value

`out` Matrix with columns: number of components for the A-mode, number of components for the B-mode, number of components for the C-mode, goodness of fit (%), total number of components

Note

Cumulative sum of eigenvalues and fits from PCAsup applied to the A-, B- and C-modes are automatically printed.

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1991). Hierarchical relations among three-way methods. *Psychometrika 56:449–470*.

See Also

DimSelector, LineCon, pcasup3, T3

Examples

```
data(Bus)
# Fit values of T3 with different numbers of components (from 1 to 4 for the A-mode,
# from 1 to 3 for the B-mode, from 1 to 5 for the C-mode)
FitT3 <- T3runsApproxFit(Bus, 7, 5, 37, 4, 3, 5)
```

---

threewayanova *Three-way ANOVA*

Description

Computation of three-way Analysis of Variance (ANOVA).

Usage

```
threewayanova(Y, n, m, p)
```

Arguments

- `Y` — Matrix (or data.frame coerced to a matrix) of order (`n` x `mp`) containing the matricized array (frontal slices)
- `n` — Number of `A`-mode entities
- `m` — Number of `B`-mode entities
- `p` — Number of `C`-mode entities

Value

A list including the following components:

- `SS.a` — Main effect for the `A`-mode
- `SS.b` — Main effect for the `B`-mode
- `SS.c` — Main effect for the `C`-mode
- `SS.ab` — Second order interaction (`A`- and `B`-mode)
- `SS.bc` — Second order interaction (`B`- and `C`-mode)
- `SS.ac` — Second order interaction (`A`- and `C`-mode)
- `SS.abc` — Residual sum of squares after subtraction of second order interactions

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers \& I. Van Mechelen (2001). Three-way component analysis: principles and illustrative applications. *Psychological Methods 6:84–110.*

Examples

```
data(TV)
TVdata=TV[[1]]
anova3 <- threewayanova(TVdata, 16, 15, 30)
```

`tr` *Trace*

Description

Computes the trace of a matrix.

Usage

```
tr(a)
```

Arguments

`a`	Matrix

Value

`t`	Trace of `A`

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

Examples

```
X <- matrix(rnorm(6*6),ncol=6)
trace <- tr(X)
```

`TV` *TV data*

Description

Three-way data about ratings of 15 American television shows on 16 bipolar scales made by 30 students.

Usage

```
data(TV)
```

Format

A list containing one data.frame and three character vectors.
TV[[1]] is a data.frame with 16 rows and 450 (15 x 30) columns.
The rows refer to the American television shows.
The columns refer to the combinations of scales and students with the sclaes nested within the students.
The data.frame contains the frontal slices next to each other of the original array.
The labels for the bipolar scales are in the character vector TV[[2]].
The labels for the TV programs are in the character vector TV[[3]].
The labels for the students are in the character vector TV[[4]].

Details

The original data set consists of ratings made by 40 subjects (psychology students at the University of Western Ontario in 1981). To avoid missing data, only 30 students are considered. The ratings are made on 13-point bipolar scales. Lundy et al. (1989) perform Candecomp/Parafac on the preprocessed data. Details on preprocessing are not reported, but should be centered within TV programs and scales. Three real components are extracted. However, the unconstrained Candecomp/Parafac solution with three components suffers from the so-called degeneracy (obtained solution with highly correlated and uninterpretable dimensions). Degeneracy (see, for instance, Harshman \& Lundy, 1984; Stegeman, 2006, 2007; De Silva \& Lim, 2008; Rocci \& Giordani, 2010) can be overcome by imposing orthogonal constraints in one of the component matrices. The so-obtained solution with three components is meaningful and interpretable as described in Lundy et al. (1989).

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

V. De Silva \& L.-H. Lim (2008). Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications 30:1084–1127*.
R.A. Harshman \& M.E. Lundy (1984). Data preprocessing and the extended PARAFAC model. In *H.G. Law, C.W. Snyder Jr, J.A. Hattie, \& R.P. McDonald (Eds.): Research methods for multimode data analysis*. Praeger, New York (pp. 216–284).
M.E. Lundy, R.A. Harshman \& J.B. Kruskal (1989). A two-stage procedure incorporating good features of both trilinear and quadrilinear models. In *R. Coppi, S. Bolasco (Eds.): Multiway Data Analysis*. Elsevier, North Holland (pp. 123–130).
R. Rocci R \& P. Giordani (2010). A weak degeneracy decomposition for the CANDECOMP/PARAFAC model. *Journal of Chemometrics 24:57–66*.
A. Stegeman (2006). Degeneracy in Candecomp/Parafac explained for pxpx2 arrays of rank p+1 or higher. *Psychometrika 71:483–501*.
A. Stegeman (2007). Degeneracy in Candecomp/Parafac and Indscal explained for several three-sliced arrays with a two-valued typical rank. *Psychometrika 72:601–619*.

Examples

```
# to perform stability check and produce bootstrap confidence intervals
# it is useful to permute the modes so that the A-mode refers to students
data(TV)
TVdata=TV[[1]]
labSCALE=TV[[2]]
labPROGRAM=TV[[3]]
labSTUDENT=TV[[4]]
TVdata <- permnew(TVdata, 16, 15, 30)
TVdata <- permnew(TVdata, 15, 30, 16)
```

---

varim *Varimax roation*

Description

Produces varimax rotated version of A and rotation matrix T.

Usage

```
varim(A)
```

Arguments

A	Matrix to be to be rotated

Value

A list including the following components:

B	Rotated version of A (B=AT)
T	Rotation matrix
f	Varimax function value

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H. Kaiser (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika 23:187–200*.
K. Nevels (1986). A direct solution for pairwise rotations in Kaiser's varimax method. *Psychometrika 51:327–329*.

See Also

normvari

Examples

```
X <- matrix(rnorm(6*3),ncol=3)
Y <- varim(X)
# varimax rotated version of X
Y$B
# rotation matrix
Y$T
```

---

varimcoco *Varimax Rotation for Tucker3 and Tucker2*

Description

Performs varimax rotation of the core and component matrix rotations to simple structure.

Usage

```
varimcoco(A, B, C, H, wa_rel, wb_rel, wc_rel, rot1, rot2, rot3, nanal)
```

Arguments

A	Columnwise orthomornal component matrix for the A-mode
B	Columnwise orthomornal component matrix for the B-mode
C	Columnwise orthomornal component matrix for the C-mode
H	Matricized core array (frontal slices)
wa_rel	relative weight (>=0) for the simplicity of A
wb_rel	relative weight (>=0) for the simplicity of B
wc_rel	relative weight (>=0) for the simplicity of C
rot1	binary indicator (1 if the A-mode is rotated, 0 otherwise, default 1)
rot2	binary indicator (1 if the B-mode is rotated, 0 otherwise, default 1)
rot3	binary indicator (1 if the C-mode is rotated, 0 otherwise, default 1)
nanal	Number of random starts, default 5

Value

A list including the following components:

AS	Rotated component matrix for the A-mode
BT	Rotated component matrix for the B-mode
CU	Rotated component matrix for the C-mode

K	Rotated matricized core array (frontal slices)
S	Rotation matrix for the A-mode
T	Rotation matrix for the B-mode
U	Rotation matrix for the C-mode
f	Best solution for three-way orthomax function value
f1	Varimax value of H
f2a	Varimax value of AS
f2b	Varimax value of BT
f2c	Varimax value of CU
func	Function values upon convergence for all the runs of the orthomax algorithm

Note

The simplicity values f1, f2a, f2b, f2c are based on 'natural' weigths and therefore comparable across matrices. When multiplied by the relative weights, they give the contribution to the overall simplicity value (they are I^2/p, J^2/q or K^2/r, respectively, times the sum of the variances of squared values).

Author(s)

Maria Antonietta Del Ferraro <mariaantonietta.delferraro@yahoo.it>
Henk A.L. Kiers <h.a.l.kiers@rug.nl>
Paolo Giordani <paolo.giordani@uniroma1.it>

References

H.A.L. Kiers (1998). Joint orthomax rotation of the core and component matrices resulting from three-mode principal components analysis. *Journal of Classification 15:245–263*.

See Also

orthmax2, varim

Examples

```
data(Bus)
# T3 solution
BusT3 <- T3funcrep(Bus, 7, 5, 37, 2, 2, 2, 0, 1e-6)
# Simplicity of A (with weight = 2.5), B (with weight = 2) and C (with weight = 1.5)
T3vmABC <- varimcoco(BusT3$A, BusT3$B, BusT3$C, BusT3$H, 2.5, 2, 1.5)
# Simplicity of only A (with weight = 2.5) and B (with weight = 2)
# rot3=0; the value of wc_rel (= 0) does not play an active role
T3vmAB <- varimcoco(BusT3$A, BusT3$B, BusT3$C, BusT3$H, 2.5, 2, 0, 1, 1, 0)
# simplicity repeatedly with different relative weights for A, B and C
T3vm <- list()
weight.a <- c(1, 3, 6)
weight.b <- c(0, 2, 5)
weight.c <- c(1, 4)
```

```
i <- 1
for (wa_rel in weight.a){
 for (wb_rel in weight.b){
  for (wc_rel in weight.c){
   T3vm[[i]] <- varimcoco(BusT3$A, BusT3$B, BusT3$C,
    BusT3$H, wa_rel, wb_rel, wc_rel)
   i <- i+1
  }
 }
}
```

Index