Package ‘rrcov3way’

June 22, 2017

Date 2017-06-22

Title Robust Methods for Multiway Data Analysis, Applicable also for Compositional Data

Version 0.1-10

Author
Valentin Todorov <valentin.todorov@chello.at>, Maria Anna Di Palma <madipalma@unior.it> and Michele Gallo \{

Description Provides methods for multiway data analysis by means of Parafac and Tucker 3 models. Robust versions (Engelen and Hubert (2011) <doi:10.1016/j.aca.2011.04.043>) and versions for compositional data are also provided (Gallo (2015) <doi:10.1080/03610926.2013.798664>, Di Palma et al. (in press)).

Maintainer Valentin Todorov <valentin.todorov@chello.at>

Depends ThreeWay, rrcov

Imports robustbase

LazyLoad no

LazyData no

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-22 11:30:57 UTC

R topics documented:

`Arno` *Chemical composition of water in the main stream of Arno river*

Description

Chemical composition of water in the main stream of Arno river.

Usage

```
data("Arno")
```

Format

A three-way array with dimension 23x11x4. The first dimension refers to 23 distances from the spring. The second dimension refers to the 11 chemical compositions. The third dimension refers to the time of collection - four occasions.

Details

The Arno data example was used in Gallo and Buccinati (2013) to illustrate a particular version of the Tucker model, known as the weighted principal component analysis. The Tucker3 results are usually given in the form of tables or plots and in this work for the representation of the Tucker3 results of logratio data, is proposed to use one-mode plots, clr-joint biplots (Gallo, 2015), and trajectory plots.

Source

Nisi B., Vaselli O., Buccianti A., Minissale A., Delgado-Huertas A., Tassi F., Montegrossi G. (2008). Geochemical and isotopic investigation of the dissolved load in the running waters from the Arno valley: evaluation of the natural and anthropogenic input. In Memorie Descrittive della Carta Geologica d'Italia, Nisi (eds.), 79: 1-91.

Nisi B., Buccianti A., Vaselli O., Perini G., Tassi F., Minissale A., Montegrossi G. (2008) Hydrogeochemistry and strontium isotopes in the Arno river basin (Tuscany, Italy): Constraints on natural controls by statistical modeling. *Journal of Hydrology* 360: 166-183.

References

Gallo M. and Buccianti A. (2013). Weighted principal component analysis for compositional data: application example for the water chemistry of the Arno river (Tuscany, central Italy), *Environmetrics*, 24(4):269-277.

Gallo M. (2015). Tucker3 model for compositional data. *Communications in Statistics-Theory and Methods*, 44(21):4441-4453.

Examples

```
data(Arno)
dim(Arno)               # [1] 23 11  4
dim(Arno[,,1])          # [1] 23 11
rownames(Arno[,,1])     # the 23 distances from the spring
colnames(Arno[,,1])     # the 11 chemical compositions
dim(Arno[,1,])          # [1] 23  4
colnames(Arno[,1,])     # the four occasions

res <- Tucker3(Arno, robust=FALSE, ilr=TRUE)
res

## Distance-distance plot
plot(res, which="dd", main="Distance-distance plot")

## Paired component plot, mode A
plot(res, which="comp", main="Paired component plot (mode A)")

## Paired component plot, mode B
plot(res, which="comp", mode="B", main="Paired component plot (mode B)")

## Joint biplot
plot(res, which="jbplot", main="Joint biplot")

## Trajectory
plot(res, which="tjplot", main="Trajectory biplot")
```

---

congruence *Coefficient of factor congruence (phi)*

Description

Computes the Tucker's congruence (phi) coefficients among two sets of factors.

Usage

```
congruence(x, y=NULL)
```

Arguments

`x`	A vector or matrix of factor loadings
`y`	A vector or matrix of factor loadings (may be empty)

Details

Find the Tucker's coefficient of congruence between two sets of factor loadings.

Factor congruences are the cosines of pairs of vectors defined by the loadings matrix and based at the origin. Thus, for loadings that differ only by a scaler (e.g. the size of the eigen value), the factor congruences will be 1.

For factor loading vectors of X and Y the measure of factor congruence, phi, is

$$\phi = \frac{\sum XY}{\sqrt{\sum(X^2)\sum(Y^2)}}.$$

Value

A matrix of factor congruences.

Author(s)

`<valentin.todorov@chello.t>`

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 <- getLoadings(PcaClassic(delivery))
Y <- getLoadings(PcaHubert(delivery, k=3))
round(congruence(X,Y),3)
```

`do3Postprocess` *Postprocessing: renormalization, reflection and reordering; access to some of the components of the model.*

Description

The estimated model will be renormalized, reflected (change of sign) or the components will be reordered. Functions that provide access to some components of the model: coordinates, weights.

Usage

```
    ## S3 method for class 'tucker3'
do3Postprocess(x, reflectA, reflectB, reflectC, reorderA, reorderB, reorderC, ...)
    ## S3 method for class 'parafac'
do3Postprocess(x, reflectA, reflectB, reflectC, reorder, ...)
    ## S3 method for class 'parafac'
coordinates(x, mode = c("A", "B", "C"), type = c("normalized", "unit", "principal"), ...)
    ## S3 method for class 'tucker3'
coordinates(x, mode = c("A", "B", "C"), type = c("normalized", "unit", "principal"), ...)
    ## S3 method for class 'parafac'
weights(object, ...)
    ## S3 method for class 'tucker3'
weights(object, mode = c("A", "B", "C"), ...)
    ## S3 method for class 'parafac'
reflect(x, mode = c("A", "B", "C"), rsign = 1, ...)
    ## S3 method for class 'tucker3'
reflect(x, mode = c("A", "B", "C"), rsign = 1, ...)
```

Arguments

`x`	Tucker3 or Parafac solution
`object`	Tucker3 or Parafac solution (alternative of `x` for the generic function `weights()`)
`reflectA`	How to handle the signs of the components of mode A - can be a single number or a vector with length of the number of components of A
`reflectB`	How to handle the signs of the components of mode B - can be a single number or a vector with length of the number of components of B
`reflectC`	How to handle the signs of the components of mode C - can be a single number or a vector with length of the number of components of C
`reorder`	How to reorder the components of a Parafac solution - a vector with length of the number of components
`reorderA`	How to reorder the components of mode A - a vector with length of the number of components of A giving the new order
`reorderB`	How to reorder the components of mode B - a vector with length of the number of components of B giving the new order
`reorderC`	How to reorder the components of mode C - a vector with length of the number of components of C giving the new order
`mode`	For which mode to provide the coordinates or weights. Default is mode A
`type`	Which type of coordinates to provide. Default is "normalized"
`rsign`	How to change the sign of the components of the given mode. Can be a single number or a vector with length of the number of components of the corresponding mode.
`...`	Potential further arguments passed to lower level functions.

Value

The output value of do3Postproces() is the postprocessed solution, Parafac or Tucker3. The output of `weights()` and `coordinates()` are the respoective values.

Author(s)

Valentin Todorov <valentin.todorov@chello.at> and Maria Anna Di Palma <madipalma@unior.it> and Michele Gallo <mgallo@unior.it>

References

REFERENCES

Examples

```
data(elind)
x1 <- do3Scale(elind, center=TRUE, scale=TRUE)
cp <- Parafac(x1, ncomp=3, orth="A")
cp$B
cp1 <- do3Postprocess(cp, reflectB=-1)      # change the sign of all components of B
cp$B
weights(cp1)
coordinates(cp1)
coordinates(cp1, type="principal")
```

`do3Scale` *Centering and scaling*

Description

Centering and/or normalization of a three way array or a matricized array across one mode (modes indicated by "A", "B" or "C").

Usage

```
    ## S3 method for class 'tucker3'
do3Scale(x, renorm.mode = c("A", "B", "C"), ...)
    ## S3 method for class 'parafac'
do3Scale(x, renorm.mode = c("A", "B", "C"), ...)
    ## Default S3 method:
do3Scale(x, center = FALSE, scale = FALSE,
        center.mode = c("A", "B", "C"), scale.mode = c("B", "A", "C"),
        only.data=TRUE, ...)
```

Arguments

`x`	Three dimensional array of order (I x J x K) or a matrix (or data.frame coerced to a matrix) of order (I x JK) containing the matricized array (frontal slices)
`center`	Whether and how to center the data. Can be `NULL`, logical `TRUE` or `FALSE`, function or a numeric vector with length corresponding to the number of columns in the corresponding mode. If `center=TRUE`, `mean()` is used; default is `center=FALSE`.
`scale`	Whether and how to scale the data. Can be `NULL`, logical `TRUE` or `FALSE`, function or a numeric vector with length corresponding to the number of columns in the corresponding mode. If `scale=TRUE`, `sd()` is used; default is `scale=FALSE`.
`center.mode`	Across which mode to center. Default is `center.mode="A"`
`scale.mode`	Within which mode to scale. Default is `scale.mode="B"`
`renorm.mode`	Within which mode to renormalize a Parafac or Tucker3 solution. See in Details how this is performed for the different models. Default is `renorm.mode="A"`
`only.data`	Whether to return only the centered/scaled data or also the center and the scale themselves. Default is `only.data=TRUE`
`...`	potential further arguments passed to lower level functions.

Value

A named list, consisting of the centered and/or scaled data, a center vector, a scale vector and the mode in which the data were centered/scaled.

Author(s)

Valentin Todorov <valentin.todorov@chello.at> and Maria Anna Di Palma <madipalma@unior.it> and Michele Gallo <mgallo@unior.it>

References

Kiers, H.A.L. (2000).Towards a standardizrd notationand terminology in multiway analysis. *Journal of Chemometrics*, 14:105-122.

Kroonenberg, P.M. (1983).Three-mode principal component analysis: Theory and applications (Vol. 2), DSWO press.

Examples

```
data(elind)
(x1 <- do3Scale(elind, center=TRUE, scale=TRUE))
(x2 <- do3Scale(elind, center=TRUE, scale=TRUE, center.mode="B"))
(x3 <- do3Scale(elind, center=TRUE, scale=TRUE, center.mode="C", scale.mode="C"))
```

elind *OECD Electronics Industries Data*

Description

OECD publishes comparative statistics of the export size of various sectors of the electronics industry:

1. information science,
2. telecommunication products,
3. radio and television equipment,
4. components and parts,
5. electromedical equipment, and
6. scientific equipment.

The data consist of specialisation indices of electronics industries of 23 European countries for the years 1973–1979. The specialization index is defined as the proportion of the monetary value of an electronic industry compared to the total export value of manufactured goods of a country compared to the similar proportion for the world as a whole (see D'Ambra, 1985, p. 249 and Kroonenberg, 2008, p.282).

Usage

```
data(elind)
```

Format

A three-way array with dimension 23x6x7. The first dimension refers to 23 countries. The second dimension refers to the six indices of electronics industries. The third dimension refers to the years in the period 1978–1985.

Source

The data set is available from Pieter Kroonenberg's web site at: http://three-mode.leidenuniv.nl/data/electronicindustriesinfo.htm

References

D'Ambra, L. (1985). Alcune estensione dell'analisi in componenti principali per lo studio dei sistemi evolutivi. Uno studio sul commercio internazionale dell'elettronica. In: Ricerche Economiche. 2. del Dipartimento di Scienze Economiche Ca'Foscari, Venezia.

Kroonenberg PM (2008). Applied multiway data analysis. Wiley series in probability and statistics. John Wiley and Sons, Hoboken, NJ, p.282.

Examples

```
data(elind)

res <- Parafac(elind, robust=FALSE, ilr=FALSE)

## Distance-distance plot
plot(res, which="dd", main="Distance-distance plot")

## Paired component plot, mode A
plot(res, which="comp", main="Paired component plot (mode A)")

## Paired component plot, mode B
plot(res, which="comp", mode="B", main="Paired component plot (mode B)")

## Per-component plot
plot(res, which="percomp", comp=1, main="Per component plot")

## all components plot
plot(res, which="allcomp", main="All components plot", legend.position="topright")
```

`girls` *Sempe girls' growth curves data*

Description

Thirty girls selected from a French auxiological study (1953-1975) to get insight into the physical growth patterns of children from ages four to fifteen, Sempe (1987). They were measured yearly between the ages 4 and 15 on the following eight variables:

1. weight = Weight
2. length = Length
3. crump = Crown-rump length
4. head = Head circumference
5. chest = Chest circumference
6. arm = Arm
7. calf = Calf
8. pelvis = Pelvis

The data set is three way data array of size 30 (girls) x 8 (variables) x 12 (years).

Usage

```
data("girls")
```

Format

The format is a three way array with the following dimensions: The first dimension refers to 30 girls. The second dimension refers to the eight variables measured on the girls. The third dimension refers to the years – 4 to 15.

Details

The data are generally preprocessed as standard multiway profile data. For details see Kroonenberg (2008), Chapters 6 and 15.

Source

The data sets are available from Pieter Kroonenberg's web site at: http://www.leidenuniv.nl/fsw/three-mode/data/girlsgrowthcurvesinfo.htm

References

Sempe, M. (1987). Multivariate and longitudinal data on growing children: Presentation of the French auxiological survey. In J.Janssen et al. Data analysis. The Ins and Outs of solving real problems (pp. 3-6). New York: Plenum Press.

Kroonenberg (2008). Applied multiway data analysis. Wiley series in probability and statistics. Hoboken NJ, Wiley.

Examples

```
data(girls)
str(girls)
## Center the data in mode A and find the "average girl"
center.girls <- do3Scale(girls, center=TRUE, only.data=FALSE)
X <- center.girls$x
center <- center.girls$center
average.girl <- as.data.frame(matrix(center, ncol=8, byrow=TRUE))
dimnames(average.girl) <- list(dimnames(X)[[3]], dimnames(X)[[2]])

## Divide these variables by 10 to reduce their range
average.girl$weight <- average.girl$weight/10
average.girl$length <- average.girl$length/10
average.girl$crrump <- average.girl$crrump/10

average.girl
p <- ncol(average.girl)
plot(rownames(average.girl), average.girl[,1], ylim=c(min(average.girl),
    max(average.girl)), type="n", xlab="Age", ylab="")
for(i in 1: p)
{
    lines(rownames(average.girl), average.girl[,i], lty=i, col=i)
    points(rownames(average.girl), average.girl[,i], pch=i, col=i)
}
legend <- colnames(average.girl)
legend[1] <- paste0(legend[1], "*")
legend[2] <- paste0(legend[3], "*")
```

```
legend[3] <- paste0(legend[4], "*")
legend("topleft", legend=legend, col=1:p, lty=1:p, pch=1:p)
```

Kojima   *Parental behaviour in Japan*

Description

The data are drawn from a study (Kojima, 1975) of the perception of parental behaviour by parents and their children. Two data sets, boys and girls are available as Kojima.boys and Kojima.girls.

- Boys data were analysed in Kroonenberg (2008)
- Girls data were analysed in Kroonenberg, Harshman, & Murakami (2009).

Usage

```
data(Kojima)
```

Format

Both data sets are three dimensional arrays:

- boys: 150 x 18 x 4
- girls: 153 x 18 x 4

The rows (1st mode) are 150 Japanese sons/153 Japanese daughters. The columns (2nd mode) are 18 scales (Acceptance, Child centerness, Possesiveness, etc.). The slices (3rd mode) are the 4 judgements (See Details for explanation).

Details

The `boys` data are ratings expressing the judgments of parents with respect to their own behaviour toward their sons, and the judgments of their sons with respect to their parents. Thus, there are four conditions:

- Father-Own behaviour (F-F),
- Mother-Own behaviour (M-M),
- Son-Father (B-F),
- Son-Mother (B-M).

The judgments involved 150 middle-class Japanese eighth-grade boys on the 18 subscales of the inventory. Thus, the data set consists of a 150 (Sons) x 18 (scales) x 4 (judgment combinations) data array.

Similarly, the girls data are ratings expressing the judgments of parents with respect to their own behaviour toward their daughters, and the judgments of their daughters with respect to their parents. Thus, there are four conditions:

- Father-Own behaviour (F-F),
- Mother-Own behaviour (M-M),
- Daughter-Father (G-F),
- Daughter-Mother (G-M).

The judgments involved 153 middle-class Japanese eighth-grade girls on the 18 subscales of the inventory. Thus, the data set consists of a 153 (Daughters) x 18 (scales) x 4 (judgment combinations) data array.

Preprocessing Given that the data are three-way profile data they are treated in the standard manner: centering per occasion-variable combination and by normalising the data after centring per lateral slice i.e. per scale over all sons/daughters x judges combinations. For details see Kroonenberg (2008), Chapter 13.

Source

The data sets are available from the Pieter Kroonenberg's web site at `http://three-mode.leidenuniv.nl/`.

References

Kojima, H. (1975). Inter-battery factor analysis of parents' and children's reports of parental behavior. Japanese Psychological Bulletin, 17, 33-48 (in Japanese).

Kroonenberg, P. M. (2008). Applied multiway data analysis. Wiley series in probability and statistics. Wiley, Hoboken NJ.

Kroonenberg, P. M., Harshman, R. A, & Murakami, T. (2009). Analysing three-way profile data using the Parafac and Tucker3 models illustrated with views on parenting. Applied Multivariate Research, 13:5-41. PDF available at: http://www.phaenex.uwindsor.ca/ojs/leddy/index.php/AMR/article/viewFile/2833/2271

Examples

```
data(Kojima)
dim(Kojima.boys)
dim(Kojima.girls)
```

---

krp — *Function to compute Khatri-Rao product*

Description

The function `krp(A,B)` returns the Khatri-Rao product of two matrices `A` and `B`, of dimensions I x K and J x K respectively. The result is an IJ x K matrix formed by the matching column-wise Kronecker products, i.e. the k-th column of the Khatri-Rao product is defined as `kronecker(A[, k], B[, k])`.

Usage

```
krp(A, B)
```

Arguments

A	Matrix of order I x K.
B	Matrix of order J x K.

Value

The IJ x K matrix of columnwise Kronecker products.

Note

A and B must have the same number of columns.

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References

Khatri, C.G. and Radhakrishna Rao, C. (1968). Solutions to Some Functional Equations and Their Applications to Characterization of Probability Distributions. *Sankhya: Indian J. Statistics (Series A)*, 30:167-180. Smilde, A., Bro R. and Gelardi, P. (2004). Multi-way Analysis: Applications in Chemical Sciences, Chichester:Wiley

Examples

```
A <- matrix(1, nrow=5, ncol=2)
B <- matrix(1:4, ncol=2)
krp(A,B)

A <- matrix(1:10, ncol=2, byrow=TRUE)
B <- diag(1,2)
krp(A,B)
```

Parafac — *Robust Parafac estimator for compositional data*

Description

Compute a robust Parafac model for compositional data

Usage

```
Parafac(X, ncomp = 2, conv = 1e-06, center = FALSE,
    scale=FALSE, scale.mode=c("B", "A", "C"),
    orth=c(), robust = FALSE, ilr = FALSE, ncomp.rpca = 2,
    alpha = 0.75, maxiter = 100, crit=0.975, trace = FALSE)
```

Arguments

`X`	3-way array of data
`ncomp`	Number of components
`conv`	Convergence criterion, defaults to `1e-6`
`center`	Whether to center the data
`scale`	Whether to scale the data
`scale.mode`	If centering and/or scaling the data, on which mode to do this
`orth`	Orthogonality constraints
`robust`	Whether to apply a robust estimation
`ilr`	If the data are a composition, use an ilr transformation
`ncomp.rpca`	Number of components for robust PCA
`alpha`	Measures the fraction of outliers the algorithm should resist. Allowed values are between 0.5 and 1 and the default is 0.75
`maxiter`	Maximal number of iterations
`crit`	Cut-off for identifying outliers, default `crit=0.975`
`trace`	Logical, provide trace output

Details

The function can compute four versions of the Parafac model:

1. Classical Parafac,
2. Parafac for compositional data,
3. Robust Parafac and
4. Robust Parafac for compositional data.

This is controlled though the paramters `robust=TRUE` and `ilr=TRUE`.

Value

An object of class "parafac" which is basically a list with components:

`fit`	Fit value
`fp`	Fit percentage
`A`	Orthogonal loading matrix for the A-mode
`B`	Orthogonal loading matrix for the A-mode
`Bclr`	Orthogonal loading matrix for the B-mode, clr transformed. Available only if `ilr=TRUE` (default), otherwise NULL
`C`	Orthogonal loading matrix for the C-mode
`Xhat`	Robust reconstructed array
`iter`	Number of iterations
`RD`	Residual distances

`flag`	The observations whose residual distance RD is larger than cutoff.RD can be considered as outliers and receive a flag equal to zero. The regular observations receive a flag 1
`robust`	The paramater `robust`, whether robust method is used or not
`ilr`	The paramater `ilr`, whether ilr transformation is used or not

Author(s)

Valentin Todorov <valentin.todorov@chello.at> and Maria Anna Di Palma <madipalma@unior.it> and Michele Gallo <mgallo@unior.it>

References

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

Engelen, S., Frosch, S. and Jorgensen, B.M. (2009). A fully robust PARAFAC method analyzing fluorescence data. *Journal of Chemometrics*, 23(3): 124–131.

Kroonenberg, P.M. (1983).Three-mode principal component analysis: Theory and applications (Vol. 2), DSWO press.

Rousseeuw, P.J. and Driessen, K.V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3): 212–223.

Egozcue J.J., Pawlowsky-Glahn V., Mateu-Figueras G. and Barcel'o-Vidal, C. (2003). Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3): 279-300

Examples

```
#############
##
## Example with the UNIDO Manufacturing value added data

data(va3way)
dim(va3way)

## Treat quickly and dirty the zeros in the data set (if any)
va3way[va3way==0] <- 0.001

##
res <- Parafac(va3way)
res
print(res$fit)
print(res$A)

## Distance-distance plot
plot(res, which="dd", main="Distance-distance plot")

data(ulabor)
res <- Parafac(ulabor, robust=TRUE, ilr=TRUE)
res
```

```
## Plot Orthonormalized A-mode component plot
plot(res, which="comp", mode="A", main="Component plot, A-mode")

## Plot Orthonormalized B-mode component plot
plot(res, which="comp", mode="B", main="Component plot, B-mode")

## Plot Orthonormalized B-mode component plot
plot(res, which="comp", mode="C", main="Component plot, C-mode")
```

---

permute *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

```
permute(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

References

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

Examples

```
X <- array(1:24, c(4,3,2))
dim(X)

## matricize the array

##matricized X with the A-mode entities in its rows
Xa <- unfold(X)
dim(Xa)
```

```
Xa

## matricized X with the B-mode entities in its rows
Xb <- permute(Xa, 4, 3, 2)
dim(Xb)
Xb

## matricized X with the C-mode entities in its rows
Xc <- permute(Xb, 3, 2, 4)
dim(Xc)
Xc
```

---

plot.tucker3 *Plot a tucker3 object*

Description

Different plots for the results of Tucker3 analysis, stored in a tucker3 object, see Details.

Usage

```
    ## S3 method for class 'tucker3'
plot(x, which = c("dd", "comp", "allcomp", "jbplot",
   "tjplot", "all"), ask = (which == "all" && dev.interactive(TRUE)), id.n, ...)
```

Arguments

x	A tucker3 object
which	Which plot to select (see Details). Default is dd, Distance-distance plot.
ask	Generates all plots in interactive mode
id.n	Number of items to highlight
...	Other parameters to be passed to the lower level functions.

Details

Different plots for a tucker3 object will be produced. Use the parameter which to select which plot to produce:

dd Distance-distance plot

comp Paired component plot

jbplot Joint biplot

tjplot Trajectory plot

Author(s)

Valentin Todorov <valentin.todorov@chello.at> and Maria Anna Di Palma <madipalma@unior.it> and Michele Gallo <mgallo@unior.it>

References

Kiers, H.A. (2000).Some procedures for displaying results from three-way methods. *Journal of Chemometrics*. 14(3): 151-170.

Kroonenberg, P.M. (1983).Three-mode principal component analysis: Theory and applications (Vol. 2), DSWO press.

Examples

```
##
## The Bus data from package ThreeWay
##
data(Bus)       # from ThreeWay: 7x5x37, unfolded to 7x185
dim(Bus)

## Preprocessing
##
max.scale <- c(47, 10, 10, 15, 15)
maxBus <- rep(max.scale, 37)
BusN <- t(t(Bus)/maxBus)

## Use rarray to turn back to a 3-way array
##
X <- rarray(BusN, 7, 5, 37)
dimnames(X) <- list(dimnames(Bus)[[1]], c("L-Occ", "P-Occ", "Q-Occ",
    "S-Occ", "R-Occ"), paste("V",1:37,sep=""))

## Perform classical Tucker3 (nor robust option and no
##  ilr transformation for compositional data)
res <- Tucker3(X, robust=FALSE, ilr=FALSE)

## Distance-distance plot
plot(res, which="dd", main="Distance-distance plot")

## Paired component plot, mode A
plot(res, which="comp", main="Paired component plot (mode A)")

## Paired component plot, mode B
plot(res, which="comp", mode="B", main="Paired component plot (mode B)")

## Joint biplot
plot(res, which="jbplot", main="Joint biplot")

## Trajectory plot
plot(res, which="tjplot", main="Trajectory biplot")
```

`toArray` *Matrix to array conversion*

Description

Restore an array from its matricization with all the frontal slices of the array next to each other (mode="A")

Usage

```
toArray(x, n, m, r, mode = c("A", "B", "C"))
```

Arguments

`x`	Matrix (or data.frame coerced to a matrix) containing the elements of the frontal slices of an array
`n`	number of A-mode elements
`m`	number of B-mode elements
`r`	number of C-mode elements
`mode`	in which mode is the matricized array

Value

Three way array

Author(s)

Valentin Todorov <valentin.todorov@chello.at> and Maria Anna Di Palma <madipalma@unior.it> and Michele Gallo <mgallo@unior.it>

References

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

Examples

```
data(elind)
di <- dim(elind)
toArray(unfold(elind), di[1], di[2], di[3])
```

Tucker3 — *Robust Tucker3 estimator for compositional data*

Description

Compute a robust Tucker3 model for compositional data

Usage

```
Tucker3(X, P = 2, Q = 2, R = 2, conv = 1e-06,
    center = FALSE, scale = FALSE, scale.mode = c("B", "A", "C"),
    robust = FALSE, ilr = FALSE, ncomp.rpca = 2, alpha = 0.75,
    maxiter=100, crit=0.975, trace = FALSE)
```

Arguments

`X`	3-way array of data
`P`	Number of A-mode components
`Q`	Number of B-mode components
`R`	Number of C-mode components
`conv`	Convergence criterion, defaults to `1e-6`
`center`	Whether to center the data
`scale`	Whether to scale the data
`scale.mode`	If centering and/or scaling the data, on which mode to do this
`robust`	Whether to apply a robust estimation
`ilr`	If the data are a composition, use an ilr transformation
`ncomp.rpca`	Number of components for robust PCA
`alpha`	Measures the fraction of outliers the algorithm should resist. Allowed values are between 0.5 and 1 and the default is 0.75
`maxiter`	Maximal number of iterations
`crit`	Cut-off for identifying outliers, default `crit=0.975`
`trace`	Logical, provide trace output

Details

The function can compute four versions of the Tucker3 model:

1. Classical Tucker3,
2. Tucker3 for compositional data,
3. Robust Tucker3 and
4. Robust Tucker3 for compositional data.

This is controlled through the parameters `robust=TRUE` and `ilr=TRUE`.

Value

An object of class "tucker3" which is basically a list with components:

`fit`	Fit value
`fp`	Fit percentage
`A`	Orthogonal loading matrix for the A-mode

`B`	Orthogonal loading matrix for the B-mode
`Bclr`	Orthogonal loading matrix for the B-mode, clr transformed. Available only if `ilr=TRUE` (default), otherwise NULL
`C`	Orthogonal loading matrix for the C-mode
`GA`	Core matrix, which describes the relation between `A`, `B` and `C`, unfolded in A-form. The largest squared elements of the core matrix indicate the most important factors in the model of `X`.
`iter`	Number of iterations
`RD`	Residual distances
`flag`	The observations whose residual distance `RD` is larger than cutoff.RD can be considered as outliers and receive a flag equal to zero. The regular observations receive a flag 1

Author(s)

Valentin Todorov <valentin.todorov@chello.at> and Maria Anna Di Palma <madipalma@unior.it> and Michele Gallo <mgallo@unior.it>

References

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

Egozcue J.J., Pawlowsky-Glahn, V., Mateu-Figueras G. and Barcel'o-Vidal, C. (2003). Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3): 279–300.

Examples

```
##
## The Bus data from package ThreeWay
##
data(Bus)       # from ThreeWay: 7x5x37, unfolded to 7x185
dim(Bus)

## Preprocessing
##
max.scale <- c(47, 10, 10, 15, 15)
maxBus <- rep(max.scale, 37)
BusN <- t(t(Bus)/maxBus)

## Use toArray to turn to a 3-way array
##
X <- toArray(BusN, 7, 5, 37)
dimnames(X) <- list(dimnames(Bus)[[1]],
                    substr(dimnames(Bus)[[2]][1:5], 1, 1),
                    substr(dimnames(Bus)[[2]][seq(1, ncol(Bus), 5)], 3, 8))
dimnames(X)

## Perform classical Tucker3 (non-robust option and no
##  ilr transformation for compositional data)
```

```
res <- Tucker3(X, robust=FALSE, ilr=FALSE)

## Distance-distance plot
plot(res, which="dd", main="Distance-distance plot")

## Paired component plot, mode A
plot(res, which="comp", main="Paired component plot (mode A)")

## Paired component plot, mode B
plot(res, which="comp", mode="B", main="Paired component plot (mode B)")

## Joint biplot
plot(res, which="jbplot", main="Joint biplot")

## Trajectory plot
plot(res, which="tjplot", main="Trajectory biplot")


#############
##
## Example with the UNIDO Manufacturing value added data

data(va3way)
dim(va3way)

## Treat quickly and dirty the zeros in the data set (if any)
va3way[va3way==0] <- 0.001

##
res <- Tucker3(va3way)
res
print(res$fit)
print(res$A)

## Print the core matrix
print(res$GA)

## Distance-distance plot
plot(res, which="dd", main="Distance-distance plot")

## Paired component plot, mode A
plot(res, which="comp", main="Paired component plot (mode A)")

## Paired component plot, mode B
plot(res, which="comp", mode="B", main="Paired component plot (mode B)")

## Joint biplot
plot(res, which="jbplot", main="Joint biplot")

## Trajectory
plot(res, which="tjplot", main="Trajectory biplot")
```

ulabor *Undeclared labor by region in Italy*

Description

The dataset contains the undeclared labor in thousands work units. The data originate from Italy and are recorded at a regional level over a certain time horizon for five macroeconomic activities defined according to NACE Rev. 1.1 classification.

Usage

```
data("ulabor")
```

Format

A three-way array with dimension 22x5x5. The first dimension refers to 22 regions in Italy. The second dimension refers to the 5 economic activities. The third dimension refers to the years in the period 2001-2009.

Source

http://www.istat.it/it/archivio/39522

References

ISTAT (2011). Note metodologiche, la misura dell'occupazione non regolare nelle stime di contabilita nazionale [online]. Roma. Available at: http://www.istat.it/it/archivio/39522.

Di Palma M.A., Filzmoser P., Gallo M. and Hron, K. (2016). A robust CP model for compositional data, submitted.

Examples

```
data(ulabor)
dim(ulabor)
str(ulabor)

## Plot robust and non-robust DD-plots of the ilr-transformed data
usr <- par(mfrow=c(1,2))
res1 <- Parafac(ulabor, robust=TRUE, ilr=TRUE)
res2 <- Parafac(ulabor, ilr=TRUE)
plot(res1)
plot(res2)
par(usr)

## Plot Orthonormalized A-mode component plot
res <- Parafac(ulabor, robust=TRUE, ilr=TRUE)
plot(res, which="comp", mode="A", main="Component plot, A-mode")
```

```
## Plot Orthonormalized B-mode component plot
plot(res, which="comp", mode="B", main="Component plot, B-mode")

## Plot Orthonormalized B-mode component plot
plot(res, which="comp", mode="C", main="Component plot, C-mode")


## Per component plot
## adapted for the example and only for robust, ilr transformed model
##
##
res <- Parafac(ulabor, robust=TRUE, ilr=TRUE)

plot(res, which="percomp")              # component 1
plot(res, which="percomp", comp=2)      # component 2
```

---

unfold *Matrix unfolding*

Description

Conducts matricizations of a three-way array into matrices according to the selected mode.

Usage

```
unfold(x, mode=c("A", "B", "C"))
```

Arguments

x	Array to be unfolded
mode	the selected mode for unfolding

Value

A matrix represnting the input array, according to the selected mode:

- Mode=A: B-mode entities are nested within C-mode entities (all the frontal slices of the array next to each other) item Mode=B: C-mode entities nested within A-mode entities (all the horizontal slices of the array next to each other) item Mode C: A-mode entities nested within B-mode entities (all the lateral slices of the array next to each other)

Author(s)

Valentin Todorov <valentin.todorov@chello.at>

References

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

Examples

```
(X <- array(1:24, c(4,3,2)))
dim(X)

## matricize the array

##  matricized X with the A-mode entities in its rows
##  all the frontal slices of the array next to each other
##
(Xa <- unfold(X))
dim(Xa)

## matricized X with the B-mode entities in its rows
##  all the horizontal slices of the array next to each other
##
(Xb <- unfold(X, mode="B"))
dim(Xb)

## matricized X with the C-mode entities in its rows
##  all the lateral slices of the array next to each other
##
(Xc <- unfold(X, mode="C"))
dim(Xc)
```

---

va3way *Manufacturing value added by technology intensity for several years*

Description

A three-way array containing manufacturing value added by technology intensity for 55 countries in the period 2000–2010. UNIDO maintains a unique database containing key industrial statistics indicators for more than 160 countries in the world in the period 1963-2011: INDSTAT 2, available at stat.unido.org. The data are organized according to the International Standard Industrial Classification of all economic activities (ISIC) Revision 3 at 2-digit level. The present data set was created by aggregating the 23 2-digit divisions into five groups according to technology intensity, using the UNIDO derived classification (Upadhyaya, 2011). Then 55 countries were selected which have relatively complete data in the period 2000–2010.

Usage

```
data(va3way)
```

Format

A three-way array with dimension 55x5x11. The first dimension refers to 55 countries. The second dimension refers to the five categories of technology intensity described above. The third dimension refers to the years in the period 2000–2010.

Details

Note that the values in the second mode (sectors) sum up to a constant - the total manufacturing value added of a country in a given year and thus the data set has a compositional character.

Source

stat.unido.org

References

Upahdyaya S (2011). Derived classifications for industrial performance indicators. In *Int. Statistical Inst.: Proc. 58th World Statistical Congress, 2011, Dublin (Session STS022)*.

Upadhyaya S, Todorov V (2008). UNIDO Data Quality. UNIDO Staff Working Paper, Vienna.

Examples

```
data(va3way)
ct <- 2
x <- va3way[ct,,]/1000000
plot(colnames(x), x[1,], ylim=c(min(x), max(x)), type="n", ylab="Manufacturing Value
    Added in million USD", xlab="Years")
for(i in 1:nrow(x))
    lines(colnames(x), x[i,], col=i)
legend("topleft", legend=rownames(x), col=1:nrow(x), lwd=1)
title(paste("Coutnry: ", rownames(va3way[,,1])[ct]))


## Treat quickly and dirty the zeros in the data set (if any)
## in order to be able to perform ilr transformation:

va3way[va3way==0] <- 0.001

res <- Tucker3(va3way)

##
## Not yet a print function
##
print(res$fit)
print(res$A)

## Print the core matrix
print(res$GA)

## Distance-distance plot
plot(res, which="dd", main="Distance-distance plot")

## Paired component plot, mode A
plot(res, which="comp", main="Paired component plot (mode A)")

## Paired component plot, mode B
plot(res, which="comp", mode="B", main="Paired component plot (mode B)")
```

```
## Joint biplot
plot(res, which="jbplot", main="Joint biplot")

## Trajectory
plot(res, which="tjplot", main="Trajectory biplot")
```

---

waterquality *Water quality data in Wyoming, USA*

Description

Water quality data for three years of seasonal compositional groundwater chemistry data for 14 wells at a study site in Wyoming, USA. Routine water quality monitoring typically involves measurement of J parameters and constituents measured at I number of static locations at K sets of seasonal occurrences.

Usage

```
data("waterquality")
```

Format

A three-way array with dimension 14x12x10. The first dimension refers to 14 wells at a study site in Wyoming, USA. The second dimension refers to the ten most reactive and indicative dissolved constituents at the site: B, Ba, Ca, Cl, K, Mg, Na, Si, Sr, and SO4. In addition, the concentration of water in each sample was calculated. The third dimension refers to the time of collection - ten occasions.

References

Engle, M.A., Gallo, M., Schroeder, K.T., Geboy, N.J., Zupancic, J.W., (2014). Three-way compositional analysis of water quality monitoring data. *Environmental and Ecological Statistics*, 21(3):565-581.

Examples

```
data(waterquality)
dim(waterquality)               # [1] 14 12 10
dim(waterquality[,,1])          # [1] 14 12
rownames(waterquality[,,1])     # the 14 wells
colnames(waterquality[,,1])     # the 12 chemical compositions
dim(waterquality[,1,])          # [1] 14 10
colnames(waterquality[,1,])     # the ten occasions

(res <- Tucker3(waterquality, robust=FALSE, ilr=TRUE))

## Distance-distance plot
```

```
plot(res, which="dd", main="Distance-distance plot")
## Paired component plot, mode A
plot(res, which="comp", main="Paired component plot (mode A)")

## Paired component plot, mode B
plot(res, which="comp", mode="B", main="Paired component plot (mode B)")

## Joint biplot
plot(res, which="jbplot", main="Joint biplot")

## Trajectory
plot(res, which="tjplot", main="Trajectory biplot")
```

Index