Package ‘extRC’

January 13, 2020

Type Package
Title Extended RC Models for Contingency Tables
Version 1.0
Date 2019-12-24
Author Francesco Bartolucci, Antonio Forcina
Maintainer Francesco Bartolucci <francesco.bartolucci@unipg.it>
Description Maximum likelihood estimation of an extended class of row-column (RC) association models for two-dimensional contingency tables, which are formulated by a condition of reduced rank on a matrix of extended association parameters; see Forcina (2019) <arXiv:1910.13848>. These parameters are defined by choosing the logit type for the row and column variables among four different options and a transformation derived from suitable divergence measures.
License GPL (>= 2)
Imports MASS
NeedsCompilation no
Repository CRAN
Date/Publication 2020-01-13 15:40:02 UTC

R topics documented:

extRC-package ......................................................... 2
cuby ................................................................. 3
Deta ................................................................. 4
dfm ................................................................. 4
Drank ............................................................... 5
extRC ............................................................... 6
Hmat ................................................................. 7
MainRC ............................................................. 8
MatIn ............................................................... 8
mobility ............................................................ 9
plot ................................................................. 10
PraD ............................................................... 10
Overview of the Package extRC

Description

Estimation of extended RC models, which are formulate by constraining different types of association parameters to have a reduced rank.

Details

The package contains functions for maximum likelihood (ML) estimation of an extended class of row-column (RC) association models for two-dimensional contingency tables, as described in Forcina (2019). These models are formulated by a condition of reduced rank on a matrix of extended association parameters, which are defined by choosing the logit type for the row and column variables among four different options and a transformation derived from Cressie and Read (1984). Among the available alternatives, it is possible to use log-odds ratio based on different types of aggregation of the joint probabilities. The class of models generalizes that proposed in Kateri and Papaioannou (1994), Bartolucci and Forcina (2002), and Espendiller (2017), and includes the original RC association models of Goodman (1979) and the correspondence analysis model, as formulated in Goodman (1981) and Gilula et al. (1988). Maximum likelihood estimation is based on an algorithm that is an adaptation of the Aitchison and Silvey (1958) algorithm for constrained ML estimation and is related to the algorithm described in Evans and Forcina (2013) for fitting constrained marginal models.

The main function in the package is extRC that provides an output that may be shown by usual R commands print, summary, and plot.

Author(s)

Francesco Bartolucci, Antonio Forcina

Maintainer: Francesco Bartolucci <francesco.bartolucci@unipg.it>

References


**Examples**

```r
# load data
data(mobility)

# fit model for a single la
out = extRC(mobility, mod=c("l","l"), k=1, la=0.6)
summary(out)
```

cuby  

<table>
<thead>
<tr>
<th>Step length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

cuby

**Description**

Internal function that computes step length of the estimation algorithm in `extRC` by fitting a cubic polynomial.

**Usage**

cuby(g)

**Arguments**

g vector of likelihood values at different step lengths

**Value**

comp1 optimal length

**Author(s)**

Francesco Bartolucci, Antonio Forcina
**Deta**

*Computation of marginal parameters*

**Description**

Given a vector of canonical parameters coding distribution for an \( I \times J \) contingency table and the RC model specification in list `Model`, it computes vector of marginal and joint parameters and matrix of its derivatives with respect the canonical parameters.

**Usage**

```r
Deta(the, Model, der = FALSE)
```

**Arguments**

- `the`: vector of canonical parameters
- `Model`: list specifying all model components
- `der`: to require derivative computation (optional)

**Value**

- `eta`: vector of marginal parameters
- `Der`: derivative matrix with respect to canonical parameters

**Author(s)**

Francesco Bartolucci, Antonio Forcina

---

**dfm**

*First difference matrix*

**Description**

It creates a matrix of first differences of order \( k \).

**Usage**

```r
dfm(k)
```

**Arguments**

- `k`: size of the matrix

**Value**

- `D`: first difference matrix
Drank

Author(s)
Francesco Bartolucci, Antonio Forcina

Examples
D = dfm(5)
x = runif(5)
(D%*%x)

Drank Check matrix rank

Description
Given the row vectorized matrix, it computes the vector of discrepancies with respect to a certain rank and its derivative.

Usage
Drank(ga, lev, k, der = FALSE)

Arguments
ga row vectorized matrix of interaction
lev vector of the number of row and column categories in the original table (the numbers of rows and columns of the input matrix must be increased by 1)
k matrix rank
der to require derivative

Value
fr vector of discrepancies with respect to the rank
Dfr derivative of fr

References

Examples
A = matrix(rnorm(12),4) # matrix the rank of which must be checked
a = as.vector(t(A))
out = Drank(a,c(5,4),1,der=TRUE)
(out$fr)
(out$Dfr)
**Description**

Main function that fits extended RC models based on different types of aggregation (continuation, local, global) and different divergence functions defined by a suitable value of lambda.

**Usage**

```r
extRC(N, mod, k, la, marg.cons = c("free","equal","shift"))
```

**Arguments**

- `N`: observed contingency table
- `mod`: vector indicating the types of aggregation for row and column variables ("c" for continuation, "l" for local, "g" for global)
- `k`: rank required for the matrix of interaction parameters
- `la`: value of lambda parameter
- `marg.cons`: type of constraint on the marginal distributions

**Value**

- `la`: vector of lambda values (when a vector is in input)
- `dev`: deviance of the fitted model (when only one lambda value is in input) or vector of deviances (when a vector of lambda values is in input)
- `df`: degrees of freedom (when only one lambda value is in input)
- `it`: number of iterations (when only one lambda value is in input)
- `dis`: final discrepancy (when only one lambda value is in input)
- `pj`: vector of joint probabilities under the fitted model (when only one lambda value is in input)
- `eta`: full vector of marginal parameters (when only one lambda value is in input)
- `etaX`: vector of row marginal parameters (when only one lambda value is in input)
- `etaY`: vector of column marginal parameters (when only one lambda value is in input)
- `Eta`: matrix of association parameters (when only one lambda value is in input)
- `la`: vector of lambda values (when more lambda values are in input)
- `dev`: vector of deviance values (when more lambda values are in input)

**Author(s)**

Francesco Bartolucci, Antonio Forcina
Examples

# load data
data(mobility)

# for a single value of lambda, fit model with constraints of rank 1 on
# local-local logits and without constraints on the marginal distributions
out = extRC(mobility,mod=c("l","l"),k=1,la=0.6)
summary(out)

# for a single value of lambda, fit model with constraints of rank 1 on
# local-local logits and under constrain of equal marginal distributions
out = extRC(mobility,mod=c("l","l"),k=1,la=0.6,marg.cons="equal")
summary(out)

# for a single value of lambda, fit model with constraints of rank 2 on
# global-global logits and under constraint that marginal distributions
# are equal up to a constant shift
out = extRC(mobility,mod=c("g","g"),k=2,la=0.6,marg.cons="shift")
summary(out)

# fit model for a vector of lambdas
la = seq(-1.8,0.6,length.out=10)
out1 = extRC(mobility,mod=c("l","l"),k=1,la=la)
plot(out1)

Hmat

Matrix algebra transformation

Description

Internal function that performs a matrix algebra transformation that is used for estimation in extRC.

Usage

Hmat(G)

Arguments

G  input matrix

Value

H  transformed matrix

Author(s)

Francesco Bartolucci, Antonio Forcina
MainRC  *Estimation of extended RC models*

**Description**

Internal function that implements the Aitchinson-Silvey algorithm to estimate extended RC models.

**Usage**

```r
MainRC(y, Model, the0 = NULL, output = FALSE)
```

**Arguments**

- `y` row vectorized vector of frequencies of the contingency table
- `Model` list of model components
- `the0` initial vector of canonical parameters (optional)
- `output` to require full output (optional)

**Value**

- `dev` final deviance
- `df` degrees of freedom
- `pj` vector of joint probabilities
- `it` number of iterations
- `dis` final discrepancy

**Author(s)**

Francesco Bartolucci, Antonio Forcina

---

MatIn  *Aggregation matrices*

**Description**

Computation of aggregation matrices for generalized interactions that are used in `codeextRC` to estimate extended RC models.

**Usage**

```r
MatIn(lev, mod)
```
**mobility**

**Arguments**

`lev` vector number of rows and columns

`mod` type of logit for each dimension

**Value**

- **R0**: aggregation matrix for the row margin upper level
- **R1**: aggregation matrix for the row margin lower level
- **C0**: aggregation matrix for the column margin upper level
- **C1**: aggregation matrix for the column margin lower level
- **J00**: aggregation matrix for the left upper quadrant
- **J01**: aggregation matrix for the right upper quadrant
- **J10**: aggregation matrix for the left lower quadrant
- **J11**: aggregation matrix for the right lower quadrant

**Author(s)**

Francesco Bartolucci, Antonio Forcina

---

**Description**

Social mobility table of 3,500 British individuals, who are cross-classified according to their occupational status and the occupation status of their fathers.

**Usage**

```r
data("mobility")
```

**Format**

The format is: `num [1:5, 1:5] 50 28 11 3 45 174 78 150 42 ... - attr(*, "dimnames")=List of 2 ..$ : chr [1:5] "F1" "F2" "F3" "F4" ... ..$ : chr [1:5] "S1" "S2" "S3" "S4"`

**References**


**plot**                   *Plot the output*

**Description**

It plots the output of code `extRC` function for a vector of lambda values.

**Usage**

```r
## S3 method for class 'extRC'
plot(x, ...)
```

**Arguments**

- `x`          output from `extRC`
- `...`        further arguments passed to or from other methods

**Value**

None

**Author(s)**

Francesco Bartolucci, Antonio Forcina

---

**PraD**                   *Discrepancy with respect to inequality constraints*

**Description**

Internal function that, given a vector of joint probabilities from an $I \times J$ table (vectorized by row) and the RC model specification in list `Model`, computes vector of discrepancies and matrix of its derivatives with respect to the canonical parameters.

**Usage**

```r
PraD(the, Model, der = FALSE)
```

**Arguments**

- `the`     vector of canonical parameters
- `Model`   list of model components
- `der`     to require the derivative (optional)
Value

hdis    vector of discrepancies
Hdis    matrix of derivatives of discrepancies with respect to the canonical parameter (optional)

Author(s)

Francesco Bartolucci, Antonio Forcina

Description

Given the output of codeextRC function, it is written in a readable form.

Usage

```r
## S3 method for class 'extRC'
print(x, ...)
```

Arguments

x       output from `extRC`
...

Value

None

Author(s)

Francesco Bartolucci, Antonio Forcina
**summary**  
*Summary of extRC fits*

**Description**  
Summary method for the output of code `extRC` function.

**Usage**  
```r  
## S3 method for class 'extRC'  
summary(object, ...)  
```

**Arguments**  
- `object`: output from `extRC`  
- `...`: further arguments passed to or from other methods

**Value**  
None

**Author(s)**  
Francesco Bartolucci, Antonio Forcina

---

**tril**  
*Lower triangular matrix*

**Description**  
Given a square matrix, it provides the lower triangular part, including the main diagonal.

**Usage**  
```
tril(M)  
```

**Arguments**  
- `M`: square matrix

**Value**  
- `N`: transformed matrix

**Author(s)**  
Francesco Bartolucci, Antonio Forcina
Examples

\[ M = \text{matrix}(1:9,3) \]
\[ N = \text{tril}(M) \]
Index

*Topic **algebra**
  Drank, 5
  Hmat, 7
*Topic **array**
  tril, 12
*Topic **datasets**
  mobility, 9
*Topic **multivariate**
  extRC-package, 2

cuby, 3
Deta, 4
dfm, 4
Drank, 5
extRC, 2, 3, 6, 7, 8, 10–12
extRC-package, 2
Hmat, 7
MainRC, 8
MatIn, 8
mobility, 9
plot, 2, 10
PraD, 10
print, 2, 11
summary, 2, 12
tril, 12