Package ‘CUB’

February 8, 2018

Title A Class of Mixture Models for Ordinal Data
Version 1.1.2
Author Maria Iannario <maria.iannario@unina.it>, Domenico Piccolo
       <domenico.piccolo@unina.it>, Rosaria Simone <rosaria.simone@unina.it>
Maintainer Rosaria Simone <rosaria.simone@unina.it>
Description For ordinal rating data, estimate and test models within the family of
        CUB models and their extensions (where CUB stands for Combination of a
        discrete Uniform and a shifted Binomial distributions). Simulation routines, plotting facilities
        and fitting measures are also provided.
Depends R (>= 2.15.2), Formula
License GPL-2 | GPL-3
Encoding UTF-8
Imports methods
LazyData true
RoxygenNote 6.0.1
NeedsCompilation no
Suggests knitr, digest
VignetteBuilder knitr
Repository CRAN
Date/Publication 2018-02-08 19:26:58 UTC

R topics documented:

  betabinomial ............................................. 3
  betabinomialcsi ......................................... 4
  betar ................................................................ 5
  BIC.GEM ......................................................... 6
  bitcsi .......................................................... 7
  bitgama ......................................................... 8
  chi2cub ........................................................ 9
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>coef.GEM</td>
</tr>
</tbody>
</table>
betabinomial

**Description**

Compute the Beta-Binomial probabilities of ordinal responses, given feeling and overdispersion parameters for each observation.

**Usage**

\[ \text{betabinomial}(m, \text{ordinal}, \text{csivett}, \text{phivett}) \]

**Arguments**

- `m`: Number of ordinal categories
- `ordinal`: Vector of ordinal responses. Missing values are not allowed: they should be preliminarily deleted or imputed
- `csivett`: Vector of feeling parameters of the Beta-Binomial distribution for given ordinal responses
- `phivett`: Vector of overdispersion parameters of the Beta-Binomial distribution for given ordinal responses

**Details**

The Beta-Binomial distribution is the Binomial distribution in which the probability of success at each trial is random and follows the Beta distribution. It is frequently used in Bayesian statistics, empirical Bayes methods and classical statistics as an overdispersed binomial distribution.

**Value**

A vector of the same length as ordinal, containing the Beta-Binomial probabilities of each observation, for the corresponding feeling and overdispersion parameters.
References


See Also

`betar, betabinomialcsi`

Examples

```r
data(relgoods)
m<-10
ordinal<-relgoods$T
age<-2014-relgoods$BirthYear
no_na<-na.omit(cbind(ordinal,age))
ordinal<-no_na[,1]; age<-no_na[,2]
lage<-log(age)-mean(log(age))
gama<-c(-0.6, -0.3)
csivett<-logis(lage,gama)
alpha<-c(-2.3, 0.92);
ZZ<-cbind(1,lage)
phivett<-exp(ZZ%*%alpha)
pr<-betabinomial(m,ordinal,csivett,phivett)
plot(densitypr)
```

**betabinomialcsi**

*Beta-Binomial probabilities of ordinal responses, given feeling parameter for each observation*

Description

Compute the Beta-Binomial probabilities of given ordinal responses, with feeling parameter specified for each observation, and with the same overdispersion parameter for all the responses.

Usage

`betabinomialcsi(m,ordinal,csivett,phi)`

Arguments

- `m` Number of ordinal categories
- `ordinal` Vector of ordinal responses. Missing values are not allowed: they should be preliminarily deleted or imputed
- `csivett` Vector of feeling parameters of the Beta-Binomial distribution for given ordinal responses
- `phi` Overdispersion parameter of the Beta-Binomial distribution
Description

Return the Beta-Binomial distribution with parameters \( m \), \( csi \) and \( phi \).

Usage

\[
\text{betar}(m, csi, phi)
\]

Arguments

- \( m \)  
  Number of ordinal categories
- \( csi \)  
  Feeling parameter of the Beta-Binomial distribution
- \( phi \)  
  Overdispersion parameter of the Beta-Binomial distribution

Value

A vector of the same length as ordinal: each entry is the Beta-Binomial probability for the given observation for the corresponding feeling and overdispersion parameters.

References


See Also

betar, betabinomial

Examples

```r
data(relgoods)
m<-10
ordinal<-relgoods$Tv
age<-relgoods$BirthYear
no_na<-na.omit(cbind(ordinal, age))
ordinal<-no_na[,1]; age<-no_na[,2]
lage<-log(age)-mean(log(age))
gama<-c(-0.61,-0.31)
phi<-0.16
csivett<-logis(lage, gama)
pr<-betabinomialcsi(m, ordinal, csivett, phi)
plot(density(pr))
```
Value

The vector of length \( m \) of the Beta-Binomial distribution.

References


See Also

`betabinomial`

Examples

```r
m<-9
csi<-.8
phi<-.2
pr<-betar(m,csi,phi)
plot(1:m,pr,type="h", main="Beta-Binomial distribution",xlab="Ordinal categories")
points(1:m,pr,pch=19)
```

---

**BIC.GEM**

*S3 BIC method for class "GEM"*

Description

S3 BIC method for objects of class `GEM`.

Usage

```r
## S3 method for class 'GEM'
BIC(object, ...)
```

Arguments

- **object**
  - An object of class "GEM"

- **...**
  - Other arguments

Value

BIC index for the fitted model.

See Also

`logLik, GEM`
Description

Compute the shifted Binomial probabilities of ordinal responses.

Usage

bitcsi(m, ordinal, csi)

Arguments

m        Number of ordinal categories
ordinal   Vector of ordinal responses
csi       Feeling parameter of the shifted Binomial distribution

Value

A vector of the same length as ordinal, where each entry is the shifted Binomial probability of the corresponding observation.

References


See Also

probcub00, probcubp0, probcub0q

Examples

data(univer)
m<-7
csi<-0.7
ordinal<-univer$informat
pr<-bitcsi(m, ordinal, csi)
Description

Return the shifted Binomial probabilities of ordinal responses where the feeling component is explained by covariates via a logistic link.

Usage

\texttt{bitgama(m, ordinal, W, gama)}

Arguments

- \texttt{m}: Number of ordinal categories
- \texttt{ordinal}: Vector of ordinal responses
- \texttt{W}: Matrix of covariates for the feeling component
- \texttt{gama}: Vector of parameters for the feeling component, with length equal to NCOL(W)+1 to account for an intercept term (first entry of \texttt{gama})

Value

A vector of the same length as \texttt{ordinal}, where each entry is the shifted Binomial probability for the corresponding observation and feeling value.

See Also

\texttt{logis, probcubq, probcubpq}

Examples

\begin{verbatim}
    n<-100
    m<-7
    W<-sample(c(0,1),n,replace=TRUE)
    gama<-c(0.2,-0.2)
    csivett<-logis(W,gama)
    ordinal<-rbinom(n,m-1,csivett)+1
    pr<-bitgama(m,ordinal,W,gama)
\end{verbatim}
Description

Compute the $X^2$ statistic of Pearson for CUB models with one or two discrete covariates for the feeling component.

Usage

chi2cub(m, ordinal, W, pai, gama)

Arguments

m
Number of ordinal categories

ordinal
Vector of ordinal responses

W
Matrix of covariates for the feeling component

pai
Uncertainty parameter

gama
Vector of parameters for the feeling component, with length equal to NCOL(W)+1 to account for an intercept term (first entry of gama)

Details

No missing value should be present neither for ordinal nor for covariate matrices: thus, deletion or imputation procedures should be preliminarily run.

Value

A list with the following components:

df
Degrees of freedom

chi2
Value of the Pearson fitting measure

dev
Deviance indicator

References


Examples

data(univer)
m<-7
pai<-0.3
gama<-c(0.1, 0.7)
ordinal<-univer$informat; W<-univer$gender;
pearson<-chi2cub(m, ordinal, W, pai, gama)
degfree<-pearson$df
statvalue<-pearson$chi2
deviance<-pearson$dev
**Description**

S3 method: coef for class "GEM".

**Usage**

```r
## S3 method for class 'GEM'
coef(object, ...)
```

**Arguments**

- `object`: An object of class `GEM`
- `...`: Other arguments

**Details**

Returns estimated values of coefficients of the fitted model.

**Value**

ML estimates of parameters of the fitted GEM model.

**See Also**

`GEM`, `summary`

---

**Description**

Compute parameter correlation matrix for estimated model as returned by an object of class "GEM".

**Usage**

```r
cormat(object, digits=options()$digits)
```

**Arguments**

- `object`: An object of class "GEM"
- `digits`: Number of significant digits to be printed. Default is `options()$digits`
Value
Parameters correlation matrix for fitted GEM models.

See Also
GEM, vcov

cubevisual
Plot an estimated CUBE model

Description
Plotting facility for the CUBE estimation of ordinal responses.

Usage
cubevisual(ordinal,csiplot=FALSE,paiplot=FALSE,...)

Arguments
ordinal Vector of ordinal responses
csiplot Logical: should $\xi$ or $1 - \xi$ be the $y$ coordinate
paiplot Logical: should $\pi$ or $1 - \pi$ be the $x$ coordinate
...
Additional arguments to be passed to plot() and text(). Optionally, the number $m$ of ordinal categories may be passed: this is recommended if some category has zero frequency.

Details
It represents an estimated CUBE model as a point in the parameter space with the overdispersion being labeled.

Value
For a CUBE model fitted to ordinal, by default it returns a plot of the estimated $(1 - \pi, 1 - \xi)$ as a point in the parameter space, labeled with the estimated overdispersion $\phi$. Depending on csiplot and paiplot and on desired output, $x$ and $y$ coordinates may be set to $\pi$ and $\xi$, respectively.

Examples
data(univer)
ordinal<-univer$global
cubevisual(ordinal,xlim=c(0,0.5),main="Global Satisfaction",
          ylim=c(0.5,1),cex=0.8,digits=3,col="red")
Plot an estimated CUB model with shelter

Description

Plotting facility for the CUB estimation of ordinal responses when a shelter effect is included.

Usage

cubshevisual(ordinal, shelter, csiplot=FALSE, paiplot=FALSE, ...)

Arguments

- **ordinal**: Vector of ordinal responses
- **shelter**: Category corresponding to the shelter choice
- **csiplot**: Logical: should $\xi$ or $1 - \xi$ be the $y$ coordinate
- **paiplot**: Logical: should $\pi$ or $1 - \pi$ be the $x$ coordinate
- **...**: Additional arguments to be passed to `plot()` and `text()`. Optionally, the number $m$ of ordinal categories may be passed: this is recommended if some category has zero frequency.

Details

It represents an estimated CUB model with shelter effect as a point in the parameter space with shelter estimate indicated as label.

Value

For a CUB model with shelter fitted to `ordinal`, by default it returns a plot of the estimated $(1 - \pi, 1 - \xi)$ as a point in the parameter space, labeled with the estimated shelter parameter $\delta$. Depending on `csiplot` and `paiplot` and on desired output, $x$ and $y$ coordinates may be set to $\pi$ and $\xi$, respectively.

See Also

cubvisual, multicub

Examples

data(univer)
ordinal<-univer$global
cubshevisual(ordinal, shelter=7, digits=3, col="blue", main="Global Satisfaction")
cubvisual

Plot an estimated CUB model

Description

Plotting facility for the CUB estimation of ordinal responses.

Usage

cubvisual(ordinal, csiplot=FALSE, paiplot=FALSE, ...)

Arguments

- **ordinal**: Vector of ordinal responses
- **csiplot**: Logical: should $\xi$ or $1 - \xi$ be the $y$ coordinate
- **paiplot**: Logical: should $\pi$ or $1 - \pi$ be the $x$ coordinate
- **...**: Additional arguments to be passed to `plot()` and `text()`. Optionally, the number $m$ of ordinal categories may be passed: this is recommended if some category has zero frequency.

Details

It represents an estimated CUB model as a point in the parameter space with some useful options.

Value

For a CUB model fit to `ordinal`, by default it returns a plot of the estimated $(1 - \pi, 1 - \xi)$ as a point in the parameter space. Depending on `csiplot` and `paiplot` and on desired output, $x$ and $y$ coordinates may be set to $\pi$ and $\xi$, respectively.

Examples

data(univer)
ordinal<-univer$global
cubvisual(ordinal, xlim=c(0,0.5), ylim=c(0.5,1), cex=0.8, main="Global Satisfaction")
Description

The analysis of human perceptions is often carried out by resorting to questionnaires, where respondents are asked to express ratings about the items being evaluated. The standard goal of the statistical framework proposed for this kind of data (e.g. cumulative models) is to explicitly characterize the respondents’ perceptions about a latent trait, by taking into account, at the same time, the ordinal categorical scale of measurement of the involved statistical variables.

The new class of models starts from a particular assumption about the unconscious mechanism leading individuals’ responses to choose an ordinal category on a rating scale. The basic idea derives from the awareness that two latent components move the psychological process of selection among discrete alternatives: attractiveness towards the item and uncertainty in the response. Both components of models concern the stochastic mechanism in term of feeling, which is an internal/personal movement of the subject towards the item, and uncertainty pertaining to the final choice.

Thus, on the basis of experimental data and statistical motivations, the response distribution is modelled as the convex Combination of a discrete Uniform and a shifted Binomial random variable (denoted as CUB model) whose parameters may be consistently estimated and validated by maximum likelihood inference. In addition, subjects’ and objects’ covariates can be included in the model in order to assess how the characteristics of the respondents may affect the ordinal score.

CUB models have been firstly introduced by Piccolo (2003) and implemented on real datasets concerning ratings and rankings by D’Elia and Piccolo (2005), Iannario and Piccolo (2012). The CUB package allows the user to estimate and test CUB models and their extensions by using maximum likelihood methods. The package covers the main models of the class of Generalized Mixture Models with uncertainty (GEM - Iannario and Piccolo (2016a)), a comprehensive framework for modelling ordinal data. The accompanying vignettes supplies the user with detailed usage instructions and examples.

Acknowledgements: The Authors are grateful to Maria Antonietta Del Ferraro, Francesco Miranda and Giuseppe Porpora for their preliminary support in the implementation of the first version of the package.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>CUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.1.1</td>
</tr>
<tr>
<td>Date:</td>
<td>2017-10-11</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>

Author(s)

Maria Iannario, Domenico Piccolo, Rosaria Simone
deltaprob

Source

http://www.labstat.it/home/research/resources/cub-data-sets-2/

References

Piccolo D. (2003). On the moments of a mixture of uniform and shifted binomial random variables, Quaderni di Statistica, 5, 85–104

deltaprob

Mean difference of a discrete random variable

Description

Compute the Gini mean difference of a discrete distribution

Usage

deltaprob(prob)

Arguments

prob Vector of the probability distribution

Value

Numeric value of the Gini mean difference of the input probability distribution, computed according to the de Finetti-Paciello formulation.

Examples

prob<-c(0.04,0.04,0.05,0.10,0.21,0.32,0.24)
deltaprob(prob)
Normalized dissimilarity measure

Description

Compute the normalized dissimilarity measure between observed relative frequencies and estimated (theoretical) probabilities of a discrete distribution.

Usage

dissim(proba, probb)

Arguments

proba Vector of observed relative frequencies
probb Vector of estimated (theoretical) probabilities

Value

Numeric value of the dissimilarity index, assessing the distance to a perfect fit.

Examples

proba<McHPNPQLPNPSLPNPXLPNPWLPNRWLPNSWLPNQWI
probb<McHPNPTLPNPTLPNPULPNQPLPNRQLPNSRLPNRTI
dissim(proba, probb)

Log-likelihood function of a CUB model without covariates

Description

Compute the log-likelihood function of a CUB model without covariates fitting ordinal responses, possibly with subjects’ specific parameters.

Usage

ellecub(m, ordinal, assepai, assecsi)

Arguments

m Number of ordinal categories
ordinal Vector of ordinal responses
assepai Vector of uncertainty parameters for given observations (with the same length as ordinal)
assecsi Vector of feeling parameters for given observations (with the same length as ordinal)
### Description

Compute the expectation of a CUB model without covariates.

#### Usage

```r
expcub00(m, pai, csi)
```

#### Arguments

- `m`: Number of ordinal categories
- `pai`: Uncertainty parameter
- `csi`: Feeling parameter

#### References


#### See Also

- `varcub00`, `expcube`, `varcube`
### Examples

```r
m <- 10
pai <- 0.3
csi <- 0.7
meancub <- expcub00(m, pai, csi)
```

---

### Description

Compute the expectation of a CUBE model without covariates.

### Usage

```r
expcube(m, pai, csi, phi)
```

### Arguments

- **m**: Number of ordinal categories
- **pai**: Uncertainty parameter
- **csi**: Feeling parameter
- **phi**: Overdispersion parameter

### References


### See Also

`varcube, varcub00, expcub00`

### Examples

```r
m <- 10
pai <- 0.1
csi <- 0.7
phi <- 0.2
meancube <- expcube(m, pai, csi, phi)
```
**fitted.GEM**

### S3 method "fitted" for class "GEM"

**Description**

S3 method fitted for objects of class GEM.

**Usage**

```r
## S3 method for class 'GEM'
fitted(object, ...)
```

**Arguments**

- `object`: An object of class GEM
- `...`: Other arguments

**Details**

Returns the fitted probability distribution for GEM models with no covariates. If only one dichotomous covariate is included in the model to explain some components, it returns the fitted probability distribution for each profile.

**See Also**

GEM

**Examples**

```r
fitcub <- GEM(Formula(global ~ freqserv | 0), family = "cub", data = univer)
fitted(fitcub, digits = 4)
```

---

**GEM**

### Main function for GEM models

**Description**

Main function to estimate and validate GEneralized Mixture models with uncertainty.

**Usage**

```r
GEM(Formula, family = c("cub", "cube", "ihg", "cush"), data, ...)
```
Arguments

- **Formula**: Object of class Formula. Response variable is the vector of ordinal observations, see Details.
- **family**: Character string indicating which class of GEM models to fit.
- **data**: an optional data frame (or object coercible by as.data.frame to a data frame) containing the variables in the model. If missing, the variables are taken from environment(Formula).
- **...**: Additional arguments to be passed for the specification of the model. See details and examples.

Details

It is the main function for GEM models estimation, calling for the corresponding function for the specified subclass. The number of categories \( m \) is internally retrieved but it is advisable to pass it as an argument to the call if some category has zero frequency.

If `family="cub"`, then a CUB mixture model is fitted to the data to explain uncertainty, feeling and possible shelter effect by further passing the extra argument `shelter` for the corresponding category. Subjects' covariates can be included by specifying covariates matrices in the Formula as `ordinal~\gamma|w|x`, to explain uncertainty (`\gamma`), feeling (`w`) or shelter (`x`). Notice that covariates for shelter effect can be included only if specified for both feeling and uncertain (GeCUB models).

If `family="cube"`, then a CUBE mixture model (Combination of Uniform and Beta-Binomial) is fitted to the data to explain uncertainty, feeling and overdispersion. Subjects’ covariates can be also included to explain the feeling component or all the three components by specifying covariates matrices in the Formula as `ordinal~\gamma|w|z` to explain uncertainty (`\gamma`), feeling (`w`) or overdispersion (`z`). An extra logical argument `expinform` indicates whether or not to use the expected or the observed information matrix (default is FALSE).

If `family="ihg"`, then an IHG model is fitted to the data. IHG models (Inverse Hypergeometric) are nested into CUBE models (see the references below). The parameter \( \theta \) gives the probability of observing the first category and is therefore a direct measure of preference, attraction, pleasantness toward the investigated item. This is the reason why \( \theta \) is customarily referred to as the preference parameter of the IHG model. Covariates for the preference parameter \( \theta \) have to be specified in matrix form in the Formula as `ordinal~u`.

If `family="cush"`, then a CUSH model is fitted to the data (Combination of Uniform and SHElter effect). The category corresponding to the inflation should be passed via argument `shelter`. Covariates for the shelter parameter \( \delta \) are specified in matrix form Formula as `ordinal~x`.

Even if no covariate is included in the model for a given component, the corresponding model matrix needs always to be specified: in this case, it should be set to 0 (see examples below). Extra arguments include the maximum number of iterations (`maxiter`, default: `maxiter=500`) for the optimization algorithm and the required error tolerance (`toler`, default: `toler=1e-6`).

Standard methods: `logLik()`, `BIC()`, `vcov()`, `fitted()`, `coef()`, `print()`, `summary()` are implemented.

The optimization procedure is run via `optim()` when required. If the estimated variance-covariance matrix is not positive definite, the function returns a warning message and produces a matrix with NA entries.

Value

An object of the class "GEM" is a list containing the following elements:
estimates  Maximum likelihood estimates of parameters
loglik    Log-likelihood function at the final estimates
varmat    Variance-covariance matrix of final estimates
niter     Number of executed iterations
BIC       BIC index for the estimated model
ordinal   Vector of ordinal responses on which the model has been fitted
time      Processor time for execution
ellipsis  Retrieve the arguments passed to the call and extra arguments generated via the call
family    Character string indicating the sub-class of the fitted model
formula   Returns the Formula of the call for the fitted model
call      Returns the executed call

References


See Also

`logLik, coef, BIC, makeplot, summary, vcov, fitted, cormat`

Examples

```r
library(CUB)
## CUB models with no covariates
model<-GEM(Formula(Walking~0|0|0),family="cub",data=relgoods)
coef(model,digits=5)  # Estimated parameter vector (pai,csi)
logLik(model)          # Log-likelihood function at ML estimates
vcov(model,digits=4)  # Estimated Variance-Covariance matrix
cormat(model)          # Parameter Correlation matrix
```
fitted(model)  # Fitted probability distribution
makeplot(model)

# CUB model with shelter effect
model <- GEM(office~0|0, family="cub", shelter=7, data=univer)
BICshe <- BIC(model, digits=4)

# CUB model with covariate for all components - GeCUB
modelgcub <- GEM(office~gender|dichoage|freqserv, family="cub", shelter=7, data=univer)
BICgcub <- BIC(modelgcub)

# CUB model with covariate for uncertainty
modelcovpai <- GEM(Parents~Smoking|0|0, family="cub", data=relgoods)
fitted(modelcovpai)
makeplot(modelcovpai)

# CUB model with covariate for feeling
# all methods are available for this variable specification
modelcovcsi <- GEM(ReIFriends~WalkAlone|0, family="cub", data=relgoods, maxiter=10)
fitted(modelcovcsi)
makeplot(modelcovcsi)

# CUB model with covariates for both uncertainty and feeling components
lage <- log(univer$age)-mean(log(univer$age))
model <- GEM(global~gender|lage|0, family="cub", data=univer, maxiter=150)

# ML estimates of coefficients for uncertainty covariate: gender
bet <- param[1:2]
# ML estimates of coefficients for feeling covariate: lage
gama <- param[3:4]

# CUBE models with no covariates
model <- GEM(MeetRelatives~0|0, family="cube", starting=c(0.5,0.5,0.1),
data=relgoods, expinform=TRUE)

# Final ML estimates
coef(model, digits=4)
loglik(model)  # Maximum value of the log-likelihood function
vcov(model)
print(model)
BIC(model)

fitted(model)
makeplot(model)
summary(model)

# CUBE with covariate 'WalkAlone' only for the feeling component
modelcovcsi <- GEM(MeetRelatives~0|WalkAlone|0, family="cube", data=relgoods)
summary(modelcovcsi)

# CUBE with covariates for all components
modelcov <- GEM(MeetRelatives=WalkAlone|WalkAlone|WalkAlone, family="cube",
maxiter=50, toler=1e-2, data=relgoods)
BIC(modelcov)
BIC(modelcov)
### Description

Compute the normalized Gini heterogeneity index for a given discrete probability distribution.

#### Usage

```r
gini(prob)
```

#### Arguments

- `prob`  
  Vector of probability distribution or relative frequencies

#### See Also

`laakso`
Examples

```r
prob <- c(0.04, 0.04, 0.15, 0.10, 0.21, 0.32, 0.24)
gini(prob)
```

---

**inibest**

*Preliminary estimators for CUB models without covariates*

**Description**

Compute preliminary parameter estimates of a CUB model without covariates for given ordinal responses. These preliminary estimators are used within the package code to start the E-M algorithm.

**Usage**

`inibest(m, freq)`

**Arguments**

- `m`: Number of ordinal categories
- `freq`: Vector of the absolute frequencies of given ordinal responses

**Value**

A vector \((\pi, \xi)\) of the initial parameter estimates for a CUB model without covariates, given the absolute frequency distribution of ordinal responses

**References**


**See Also**

`inibestgama`

**Examples**

```r
m <- 9
c <- c(10, 24, 28, 36, 50, 43, 23, 12, 5)
estim <- inibest(m, c)
pai <- estim[1]
csi <- estim[2]
```
inibestcube

Naive estimates for CUBE models without covariates

Description
Compute naive parameter estimates of a CUBE model without covariates for given ordinal responses. These preliminary estimators are used within the package code to start the E-M algorithm.

Usage
inibestcube(m, ordinal)

Arguments
m
Number of ordinal categories
ordinal
Vector of ordinal responses

Value
A vector \((\pi, \xi, \phi)\) of parameter estimates of a CUBE model without covariates.

See Also
inibestcubecov, inibestcubecsi

Examples
```r
data(relgoods)
m<-10
ordinal<-relgoods$SocialNetwork
estim<-inibestcube(m, ordinal)  # Preliminary estimates \((p_1, c_1, \phi)\)
```

inibestcubecov

Preliminary parameter estimates for CUBE models with covariates

Description
Compute preliminary parameter estimates for a CUBE model with covariates for all the three parameters. These estimates are set as initial values to start the E-M algorithm within maximum likelihood estimation.

Usage
inibestcubecov(m, ordinal, Y, W, Z)
Arguments

- **m**: Number of ordinal categories
- **ordinal**: Vector of ordinal responses
- **Y**: Matrix of selected covariates to explain the uncertainty parameter
- **W**: Matrix of selected covariates to explain the feeling parameter
- **Z**: Matrix of selected covariates to explain the overdispersion parameter

Value

A vector \( \text{inibet}, \text{inigama}, \text{inialpha} \) of preliminary estimates of parameter vectors for \( \pi = \pi(\beta), \xi = \xi(\gamma), \phi = \phi(\alpha) \), respectively, of a CUBE model with covariates for all the three parameters. In details, \( \text{inibet}, \text{inigama} \) and \( \text{inialpha} \) have length equal to \( \text{NCOL}(Y)+1, \text{NCOL}(W)+1 \) and \( \text{NCOL}(Z)+1 \), respectively, to account for an intercept term for each component.

See Also

- `inibestcube`, `inibestcubecsi`, `inibestgama`

Examples

```r
library(relgoods)
m<-10
naord<-which(is.na(relgoods$TV))
nacovpai<-which(is.na(relgoods$Gender))
nacovcsi<-which(is.na(relgoods$year.12))
nacovphi<-which(is.na(relgoods$EducationDegree))
na<-union(union(naord,nacovpai),union(nacovcsi,nacovphi))
ordinal<-relgoods$TV[-na]
Y<-relgoods$Gender[-na]
W<-relgoods$year.12[-na]
Z<-relgoods$EducationDegree[-na]
ini<-inibestcubecov(m,ordinal,Y,W,Z)
p<-NCOL(Y)
q<-NCOL(W)
inibet<-ini[1:(p+1)]  # Preliminary estimates for uncertainty
inigama<-ini[(p+2):(p+q+2)]  # Preliminary estimates for feeling
inialpha<-ini[(p+q+3):length(ini)]  # Preliminary estimates for overdispersion
```

---

**inibestcubecsi**  
*Preliminary estimates of parameters for CUBE models with covariates only for feeling*

Description

Compute preliminary parameter estimates of a CUBE model with covariates only for feeling, given ordinal responses. These estimates are set as initial values to start the corresponding E-M algorithm within the package.
Usage

`inibestcubecsi(m, ordinal, W, starting, maxiter, toler)`

Arguments

- `m` Number of ordinal categories
- `ordinal` Vector of ordinal responses
- `W` Matrix of selected covariates to explain the feeling component
- `starting` Starting values for preliminary estimation of a CUBE without covariate
- `maxiter` Maximum number of iterations allowed for preliminary iterations
- `toler` Fixed error tolerance for final estimates for preliminary iterations

Details

Preliminary estimates for the uncertainty and the overdispersion parameters are computed by short runs of EM. As to the feeling component, it considers the nested CUB model with covariates and calls `inibestgama` to derive initial estimates for the coefficients of the selected covariates for feeling.

Value

A vector (\(\hat{\pi}\), \(\hat{\gamma}\), \(\hat{\phi}\)), where \(\hat{\pi}\) is the initial estimate for the uncertainty parameter, \(\hat{\gamma}\) is the vector of initial estimates for the feeling component (including an intercept term in the first entry), and \(\hat{\phi}\) is the initial estimate for the overdispersion parameter.

See Also

`inibestcube`, `inibestcubecov`, `inibestgama`

Examples

data(relgoods)
isnacov <- which(is.na(relgoods$Gender))
isnaord <- which(is.na(relgoods$Tv))
na <- union(isnacov, isnaord)
ordinal <- relgoods$Tv[!na]; W <- relgoods$Gender[!na]
m <- 10
starting <- rep(0.1, 3)
ini <- inibestcubecsi(m, ordinal, W, starting, maxiter = 100, toler = 1e-3)
nparam <- length(ini)

pai <- ini[1] # Preliminary estimates for uncertainty component
gamaest <- ini[2:(nparam - 1)] # Preliminary estimates for coefficients of feeling covariates
phi <- ini[nparam] # Preliminary estimates for overdispersion component
inibestgama

Preliminary parameter estimates of a CUB model with covariates for feeling

Description
Compute preliminary parameter estimates for the feeling component of a CUB model fitted to ordinal responses. These estimates are set as initial values for parameters to start the E-M algorithm.

Usage
inibestgama(m, ordinal, W)

Arguments
- m: Number of ordinal categories
- ordinal: Vector of ordinal responses
- W: Matrix of selected covariates for explaining the feeling component

Value
A vector of length equal to NCOL(W)+1, whose entries are the preliminary estimates of the parameters for the feeling component, including an intercept term as first entry.

References

See Also
inibest, inibestcubecsi

Examples
data(univer)
m<-7; ordinal<-univer$global; cov<-univer$diploma
ini<-inibestgama(m, ordinal, W=cov)
inigrid  

Grid-based preliminary parameter estimates for CUB models

Description

Compute the log-likelihood function of a CUB model with parameter vector \((\pi, \xi)\) ranging in the Cartesian product between \(x\) and \(y\), for a given absolute frequency distribution.

Usage

\[
inigrid(m, freq, x, y)
\]

Arguments

- \(m\): Number of ordinal categories
- \(freq\): Vector of length \(m\) of the absolute frequency distribution
- \(x\): A set of values to assign to the uncertainty parameter \(\pi\)
- \(y\): A set of values to assign to the feeling parameter \(\xi\)

Value

It returns the parameter vector corresponding to the maximum value of the log-likelihood for a CUB model without covariates for given frequencies.

See Also

- \texttt{inibest}

Examples

\[
\begin{align*}
m &<- 9 \\
x &<- c(0.1, 0.4, 0.6, 0.8) \\
y &<- c(0.2, 0.5, 0.7) \\
freq &<- c(10, 24, 28, 36, 50, 43, 23, 12, 5) \\
inigrid &<- inigrid(m, freq, x, y) \\
pai &<- ini[1] \\
ksi &<- ini[2]
\end{align*}
\]
iniihg

$\hat{\theta}$

**Moment estimate for the preference parameter of the IHG distribution**

**Description**

Compute the moment estimate of the preference parameter of the IHG distribution. This preliminary estimate is set as initial value within the optimization procedure for an IHG model fitting the observed frequencies.

**Usage**

```r
iniihg(m,freq)
```

**Arguments**

- `m` Number of ordinal categories
- `freq` Vector of the absolute frequency distribution of the categories

**Value**

Moment estimator of the preference parameter $\theta$.

**References**


**See Also**

`inibest, inibestcube`

**Examples**

```r
m<-9
defreq<-c(70,51,48,38,29,23,12,10,5)
initheta<-iniihg(m,freq)
```
**laakso**

*Normalized Laakso and Taagepera heterogeneity index*

**Description**

Compute the normalized Laakso and Taagepera heterogeneity index for a given discrete probability distribution.

**Usage**

`laakso(prob)`

**Arguments**

- `prob` Vector of a probability or relative frequency distribution

**References**


**See Also**

`gini`

**Examples**

```r
c(0.04, 0.04, 0.05, 0.10, 0.21, 0.32, 0.24)
lakso(prob)
```

---

**logis**

*The logistic transform*

**Description**

Create a matrix YY binding array Y with a vector of ones, placed as the first column of YY. It applies the logistic transform componentwise to the standard matrix multiplication between YY and param.

**Usage**

`logis(Y, param)`

**Arguments**

- `Y` A generic matrix or one dimensional array
- `param` Vector of coefficients, whose length is NCOL(Y) + 1 (to consider also an intercept term)
Value

Return a vector whose length is NROW(Y) and whose i-th component is the logistic function at the scalar product between the i-th row of YY and the vector param.

Examples

n<-50
Y<-sample(c(1,2,3),n,replace=TRUE)
param<-c(0.2,0.7)
logis(Y,param)

logLik.GEM

logLik S3 Method for class "GEM"

Description

S3 method: logLik() for objects of class "GEM".

Usage

## S3 method for class 'GEM'
logLik(object, ...)

Arguments

object An object of class "GEM"

... Other arguments

Value

Log-likelihood at the final ML estimates for parameters of the fitted GEM model.

See Also

loglikCUB, loglikCUBE, GEM, loglikIHG, loglikCUSH, BIC
loglikCUB

Description

Compute the log-likelihood value of a CUB model fitting given data, with or without covariates to explain the feeling and uncertainty components, or for extended CUB models with shelter effect.

Usage

loglikCUB(ordinal, m, param, Y = NULL, W = NULL, X = NULL, shelter = NULL)

Arguments

- ordinal: Vector of ordinal responses
- m: Number of ordinal categories
- param: Vector of parameters for the specified CUB model
- Y: Matrix of selected covariates to explain the uncertainty component (default: no covariate is included in the model)
- W: Matrix of selected covariates to explain the feeling component (default: no covariate is included in the model)
- X: Matrix of selected covariates to explain the shelter effect (default: no covariate is included in the model)
- shelter: Category corresponding to the shelter choice (default: no shelter effect is included in the model)

Details

If no covariate is included in the model, then param should be given in the form \((\pi, \xi)\). More generally, it should have the form \((\beta, \gamma)\) where, respectively, \(\beta\) and \(\gamma\) are the vectors of coefficients explaining the uncertainty and the feeling components, with length \(\text{NCOL}(Y)+1\) and \(\text{NCOL}(W)+1\) to account for an intercept term in the first entry. When shelter effect is considered, param corresponds to the first possible parameterization and hence should be given as \((p1, p2, c1)\). No missing value should be present neither for ordinal nor for covariate matrices: thus, deletion or imputation procedures should be preliminarily run.

See Also

logLik
Examples

```r
## Log-likelihood of a CUB model with no covariate
m<9; n<300
pai<0.6; csi<0.4
ordinal<-simcub(n,m,pai,csi)
param<-c(pai,csi)
loglikcub<-loglikCUB(ordinal,m,param)

## Log-likelihood of a CUB model with covariate for uncertainty

data(relgoods)
m<10
naord<-which(is.na(relgoods$Physician))
nacov<-which(is.na(relgoods$Gender))
na<-union(naord,nacov)
ordinal<-relgoods$Physician[-na]; Y<-relgoods$Gender[-na]
bbet<-c(-0.81,0.93); ccsi<-0.2
param<-c(bbet,ccsi)
loglikcubp<loglikCUB(ordinal,m,param,Y=Y)

## Log-likelihood of a CUB model with covariate for feeling

data(relgoods)
m<10
naord<-which(is.na(relgoods$Physician))
nacov<-which(is.na(relgoods$Gender))
na<-union(naord,nacov)
ordinal<-relgoods$Physician[-na]; W<-relgoods$Gender[-na]
pai<-0.44; gama<-c(-0.91,-0.7)
param<-c(pai,gama)
loglikcubpq<loglikCUB(ordinal,m,param,W=W)

## Log-likelihood of a CUB model with covariates for both parameters

data(relgoods)
m<10
naord<-which(is.na(relgoods$Walking))
nacovpai<-which(is.na(relgoods$Gender))
nacovcsi<-which(is.na(relgoods$Smoking))
na<-union(naord,union(nacovpai,nacovcsi))
ordinal<-relgoods$Walking[-na]
Y<-relgoods$Gender[-na]; W<-relgoods$Smoking[-na]
bet<-c(-0.45,-0.48); gama<-c(-0.55,-0.43)
param<-c(bet,gama)
loglikcubpq<loglikCUB(ordinal,m,param,Y=Y,W=W)

## Log-likelihood of a CUB model with shelter effect

m<7; n<400
pai<0.7; csi<0.16; delta<0.15
shelter<5
ordinal<-simcubshe(n,m,pai,csi,delta,shelter)
pai<(-pai*(1-delta); pai2<1-pai-delta
param<-c(pai1,pai2,csi)
loglik<loglikCUB(ordinal,m,param,shelter=shelter)
```
loglikCUBE

Log-likelihood function for CUBE models

Description

Compute the log-likelihood function for CUBE models. It is possible to include covariates in the model for explaining the feeling component or all the three parameters.

Usage

loglikCUBE(ordinal, m, param, Y=0, W=0, Z=0)

Arguments

ordinal Vector of ordinal responses
m Number of ordinal categories
param Vector of parameters for the specified CUBE model
Y Matrix of selected covariates to explain the uncertainty component (default: no covariate is included in the model)
W Matrix of selected covariates to explain the feeling component (default: no covariate is included in the model)
Z Matrix of selected covariates to explain the overdispersion component (default: no covariate is included in the model)

Details

If no covariate is included in the model, then param has the form \((\pi, \xi, \phi)\). More generally, it has the form \((\beta, \gamma, \alpha)\) where, respectively, \(\beta, \gamma, \alpha\) are the vectors of coefficients explaining the uncertainty, the feeling and the overdispersion components, with length NCOL(Y)+1, NCOL(W)+1, NCOL(Z)+1 to account for an intercept term in the first entry. No missing value should be present neither for ordinal nor for covariate matrices: thus, deletion or imputation procedures should be preliminarily run.

See Also

logLik
### Log-likelihood of a CUBE model with no covariate

```r
m<-7; n<-400
pai<-0.83; csi<-0.19; phi<-0.045
ordinal<-simcube(n,m,pai,csi,phi)
loglik<-loglikCUBE(ordinal,m,param=c(pai,csi,phi))
```

### Log-likelihood of a CUBE model with covariate for feeling data

```r
m<-10
nacov<-which(is.na(relgoods$BirthYear))
nacord<-which(is.na(relgoods$TV))
na<-union(nacov,nacord)
age<-2014-relgoods$BirthYear[-na]
lage<-log(age)-mean(log(age))
ordinal<-relgoods$TV[-na]; W<-lage
pai<-0.63; gama<-c(-0.61,-0.31); phi<-0.16
param<-c(pai,gama,phi)
loglik<-loglikCUBE(ordinal,m,param,W=W)
```

### Log-likelihood of a CUBE model with covariates for all parameters

```r
Y<-W-Z-age
bet<-c(0.18, 1.03); gama<-c(-0.6, -0.3); alpha<-c(-2.3,0.92)
param<-c(bet,gama,alpha)
loglik<-loglikCUBE(ordinal,m,param,Y=W,W,Z=Z)
```

---

**Description**

Compute the log-likelihood function for CUSH models with or without covariates to explain the shelter effect.

**Usage**

```r
loglikCUSH(ordinal,m,param,shelter,X=0)
```

**Arguments**

- `ordinal` Vector of ordinal responses
- `m` Number of ordinal categories
- `param` Vector of parameters for the specified CUSH model
- `shelter` Category corresponding to the shelter choice
- `X` Matrix of selected covariates to explain the shelter effect (default: no covariate is included in the model)
Details
If no covariate is included in the model, then param is the estimate of the shelter parameter (delta), otherwise param has length equal to NCOL(X) + 1 to account for an intercept term (first entry). No missing value should be present neither for ordinal nor for X.

See Also
GEM, logLik

Examples

```r
## Log-likelihood of CUSH model without covariates
m<-300
m<-7
shelter<-2; delta<-0.4
ordinal<-rnorm(m,delta)
loglik<-loglikCUSH(ordinal,m,theta=delta,shelter)

## Log-likelihood of CUSH model with covariates
data(relgoods)
m<-10
naord<-which(is.na(relgoods$SocialNetwork))
nacov<-which(is.na(relgoods$Gender))
na<-union(nacov,naord)
ordinal<-relgoods$SocialNetwork[-na]; cov<-relgoods$Gender[-na]
omega<-c(-2.29, 0.62)
loglikcov<-loglikCUSH(ordinal,m,theta=omega,shelter=1,X=cov)
```

---

**Description**
Compute the log-likelihood function for IHG models with or without covariates to explain the preference parameter.

**Usage**

```r
loglikIHG(ordinal,m,param,U=0)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ordinal</td>
<td>Vector of ordinal responses</td>
</tr>
<tr>
<td>m</td>
<td>Number of ordinal categories</td>
</tr>
<tr>
<td>param</td>
<td>Vector of parameters for the specified IHG model</td>
</tr>
<tr>
<td>U</td>
<td>Matrix of selected covariates to explain the preference parameter (default: no covariate is included in the model)</td>
</tr>
</tbody>
</table>
Details

If no covariate is included in the model, then `param` is the estimate of the preference parameter (`\theta`), otherwise `param` has length equal to `NCOL(U) + 1` to account for an intercept term (first entry). No missing value should be present neither for `ordinal` nor for `U`.

See Also

`GEM, logLik`

Examples

```r
### Log-likelihood of an IHG model with no covariate
m<-10; theta<-0.14; m<-300
ordinal<-simihg(n,m,theta)
loglik<-loglikIHG(ordinal,m,param=theta)

### Log-likelihood of a IHG model with covariate
data(relgoods)
m<-10
naord<-which(is.na(relgoods$HandWork))
nacov<-which(is.na(relgoods$Gender))
na<-union(naord,nacov)
ordinal<-relgoods[-na]; U<-relgoods$Gender[-na]
nu<-c(-1.55,-0.11) # first entry: intercept term
loglik<-loglikIHG(ordinal,m,param=nu,U=U); loglik
```

Description

Compute the logarithmic score of a CUB model with covariates both for the uncertainty and the feeling parameters.

Usage

```r
logscore(m,ordinal,Y,W,bet,gama)
```

Arguments

- `m` Number of ordinal categories
- `ordinal` Vector of ordinal responses
- `Y` Matrix of covariates for explaining the uncertainty component
- `W` Matrix of covariates for explaining the feeling component
- `bet` Vector of parameters for the uncertainty component, with length `NCOL(Y)+1` to account for an intercept term (first entry of `bet`)
- `gama` Vector of parameters for the feeling component, with length `NCOL(W)+1` to account for an intercept term (first entry of `gama`)
Details

No missing value should be present neither for ordinal nor for covariate matrices: thus, deletion or imputation procedures should be preliminarily run.

References


Examples

data(relgoods)
m<-10
naord<-which(is.na(relgoods$Walking))
naovpae<-which(is.na(relgoods$Gender))
naovcesi<-which(is.na(relgoods$Smoking))
na<-union(naord,union(naovpae,naovcesi))
ordinal<-relgoods$Walking[-na]
Y<-relgoods$Gender[-na]
W<-relgoods$Smoking[-na]
bet<-c(-0.45,-0.48)
gama<-c(-0.55,-0.43)
logscore(m,ordinal,Y=Y,W=W,bet,gama)

makeplot  Plot facilities for GEM objects

Description

Plot facilities for objects of class "GEM".

Usage

makeplot(object)

Arguments

object An object of class "GEM"

Details

Returns a plot comparing fitted probabilities and observed relative frequencies for GEM models without covariates. If only one explanatory dichotomous variable is included in the model for one or all components, then the function returns a plot comparing the distributions of the responses conditioned to the value of the covariate.

See Also
cubvisual, cubevisual, cubshevisual, multicub, multicube
multicub

Joint plot of estimated CUB models in the parameter space

Description
Return a plot of estimated CUB models represented as points in the parameter space.

Usage
```
multicub(listord,mvett,csiplot=FALSE,paiplot=FALSE,...)
```

Arguments
- `listord`: A data matrix, data frame, or list of vectors of ordinal observations (for variables with different number of observations)
- `mvett`: Vector of number of categories for ordinal variables in `listord` (optional: if missing, the number of categories is retrieved from data: it is advisable to specify it in case some category has zero frequency)
- `csiplot`: Logical: should $\xi$ or $1 - \xi$ be the $y$ coordinate
- `paiplot`: Logical: should $\pi$ or $1 - \pi$ be the $x$ coordinate
- `...`: Additional arguments to be passed to `plot`, `text`, and `GEM`

Value
Fit a CUB model to list elements, and then by default it returns a plot of the estimated $(1 - \pi, 1 - \xi)$ as points in the parameter space. Depending on `csiplot` and `paiplot` and on desired output, $x$ and $y$ coordinates may be set to $\pi$ and $\xi$, respectively.

Examples
```
data(univer)
listord<-univer[,8:12]
multicub(listord,colours=rep("red",5),cex=c(0.4,0.6,0.8,1,1.2),
      pch=c(1,2,3,4,5),xlim=c(0,0.4),ylim=c(0.75,1),pos=c(1,3,3,3,3))
m1<-5; m2<-7; m3<-9
pai<-.7;csi<-.6
n1<-1000; n2<-500; n3<-1500
ord1<-simcub(n1,m1,pai,csi)
ord2<-simcub(n2,m2,pai,csi)
ord3<-simcub(n3,m3,pai,csi)
listord<-list(ord1,ord2,ord3)
multicub(listord,labels=c("m=5","m=7","m=9"),pos=c(3,1,4))
```
multicube

Joint plot of estimated CUBE models in the parameter space

Description

Return a plot of estimated CUBE models represented as points in the parameter space, where the overdispersion is labeled.

Usage

```
multicube(listord,mvett,csiplot=FALSE,paiplot=FALSE,...)
```

Arguments

- **listord**: A data matrix, data frame, or list of vectors of ordinal observations (for variables with different number of observations)
- **mvett**: Vector of number of categories for ordinal variables in `listord` (optional: if missing, the number of categories is retrieved from data: it is advisable to specify it in case some category has zero frequency)
- **csiplot**: Logical: should $\xi$ or $1 - \xi$ be the $y$ coordinate
- **paiplot**: Logical: should $\pi$ or $1 - \pi$ be the $x$ coordinate
- **...**: Additional arguments to be passed to `plot`, `text`, and `GEM`

Value

Fit a CUBE model to list elements, and then by default it returns a plot of the estimated $(1 - \pi, 1 - \xi)$ as points in the parameter space, labeled with the estimated overdispersion. Depending on `csiplot` and `paiplot` and on desired output, $x$ and $y$ coordinates may be set to $\pi$ and $\xi$, respectively.

Examples

```r
ml<-5; m2<-7; m3<-9
pai<-0.7;csi<-0.6;phi=0.1
n1<-1000; n2<-500; n3<-1500
ord1<-simcube(n1,ml,pai,csi,phi)
ord2<-simcube(n2,m2,pai,csi,phi)
ord3<-simcube(n3,m3,pai,csi,phi)
listord<-list(ord1,ord2,ord3)
multicube(listord,labels=c("m=5","m=7","m=9"),pos=c(3,1,4),expinform=TRUE)
```
plotloglikihg

Plot of the log-likelihood function of the IHG distribution

Description
Plot the log-likelihood function of an IHG model fitted to a given absolute frequency distribution, over the whole support of the preference parameter. It returns also the ML estimate.

Usage
plotloglikihg(m, freq)

Arguments
m
Number of ordinal categories
freq
Vector of the absolute frequency distribution

See Also
loglikIHG

Examples
m <- 7
freq <- c(828, 275, 202, 178, 143, 110, 101)
max <- plotloglikihg(m, freq)

print.GEM

S3 method: print for class "GEM"

Description
S3 method print for objects of class GEM.

Usage
## S3 method for class 'GEM'
print(x, ...)

Arguments
x
An object of class GEM
...
Other arguments

Value
Brief summary results of the fitting procedure, including parameter estimates, their standard errors and the executed call.
**probit**  

*Probability distribution of a shifted Binomial random variable*

---

**Description**

Return the shifted Binomial probability distribution.

**Usage**

`probit(m,csi)`

**Arguments**

- `m`: Number of ordinal categories
- `csi`: Feeling parameter

**Value**

The vector of the probability distribution of a shifted Binomial model.

**See Also**

`bitcsi`, `probcubPP`

**Examples**

```r
m<-7
csi<-0.7
pr<-probit(m,csi)
plot(1:m,pr,type="h",main="Shifted Binomial probability distribution",xlab="Categories")
points(1:m,pr,pch=19)
```

---

**probcubPP**  

*Probability distribution of a CUB model without covariates*

---

**Description**

Compute the probability distribution of a CUB model without covariates.

**Usage**

`probcubPP(m,pai,csi)`
Arguments

Arguments

- `m`: Number of ordinal categories
- `pai`: Uncertainty parameter
- `csi`: Feeling parameter

Value

The vector of the probability distribution of a CUB model.

References


See Also

bitcsi, probcub0q, probcubp0, probcubpq

Examples

```r
m <- 9
pai <- 0.3
csi <- 0.8
pr <- probcub0q(m, pai, csi)
plot(1:m, pr, type = "h", main = "CUB probability distribution", xlab = "Ordinal categories")
points(1:m, pr, pch = 19)
```

probcub0q

Probability distribution of a CUB model with covariates for the feeling component

Description

Compute the probability distribution of a CUB model with covariates for the feeling component.

Usage

`probcub0q(m, ordinal, w, pai, gama)`

Arguments

- `m`: Number of ordinal categories
- `ordinal`: Vector of ordinal responses
- `w`: Matrix of covariates for explaining the feeling component. NCOL(Y)+1 to include an intercept term in the model (first entry)
probcube

\texttt{pai} Uncertainty parameter
\texttt{gama} Vector of parameters for the feeling component, whose length equals \texttt{NCOL(W)+1} to include an intercept term in the model (first entry)

\textbf{Value}

A vector of the same length as \texttt{ordinal}, whose i-th component is the probability of the i-th observation according to a CUB distribution with the corresponding values of the covariates for the feeling component and coefficients specified in \texttt{gama}.

\textbf{References}


\textbf{See Also}

\texttt{bitgama, probcub00, probcubp0, probcubpq}

\textbf{Examples}

\begin{verbatim}
data(relgoods)
m<-10 naord<-which(is.na(relgoods$Physician)) nacov<-which(is.na(relgoods$Gender)) na<-union(naord,nacov) ordinal<-relgoods$Physician[-na] W<-relgoods$Gender[-na] pai<-0.44; gama<-c(-0.91,-0.7) pr<-probcub0q(m,ordinal,W,pai,gama)
\end{verbatim}

\begin{verbatim}
probcube
\end{verbatim}

\textit{Probability distribution of a CUBE model without covariates}

\textbf{Description}

Compute the probability distribution of a CUBE model without covariates.

\textbf{Usage}

\texttt{probcube(m,pai,csi,phi)}
Arguments

- `m` Number of ordinal categories
- `pai` Uncertainty parameter
- `csi` Feeling parameter
- `phi` Overdispersion parameter

Value

The vector of the probability distribution of a CUBE model without covariates.

References


See Also

`betar`, `betabinomial`

Examples

```r
m<-9
pai<-0.3
csi<-0.8
phi<-0.1
pr<-probcube(m,pai,csi,phi)
plot(1:m,pr,type="h", main="CUBE probability distribution",xlab="Ordinal categories")
points(1:m,pr,pch=19)
```

---

`probcubp0`  
Probability distribution of a CUB model with covariates for the uncertainty component

Description

Compute the probability distribution of a CUB model with covariates for the uncertainty component.

Usage

`probcubp0(m,ordinal,Y,bet,csi)`
Arguments

- **m**
  Number of ordinal categories

- **ordinal**
  Vector of ordinal responses

- **Y**
  Matrix of covariates for explaining the uncertainty component

- **bet**
  Vector of parameters for the uncertainty component, whose length equals NCOL(Y) + 1 to include an intercept term in the model (first entry)

- **csi**
  Feeling parameter

Value

A vector of the same length as `ordinal`, whose i-th component is the probability of the i-th observation according to a CUB model with the corresponding values of the covariates for the uncertainty component and coefficients for the covariates specified in `bet`.

References


See Also

`bitgama`, `probcub00`, `probcubpq`, `probcub0q`

Examples

```r
data(relgoods)
m<10
naord<which(is.na(relgoods$Physician))
nacov<which(is.na(relgoods$Gender))
na<union(naord,nacov)
ordinal<relgoods$Physician[-na]
Y<relgoods$Gender[-na]
bet<-c(-0.81,0.93); csi<-0.20
probi<-probcubp0(m,ordinal,Y,bet,csi)
```
**probcubpq**  
*Probability distribution of a CUB model with covariates for both feeling and uncertainty*

**Description**

Compute the probability distribution of a CUB model with covariates for both the feeling and the uncertainty components.

**Usage**

```
probcubpq(m, ordinal, Y, W, bet, gama)
```

**Arguments**

- `m`: Number of ordinal categories  
- `ordinal`: Vector of ordinal responses  
- `Y`: Matrix of covariates for explaining the uncertainty component  
- `W`: Matrix of covariates for explaining the feeling component  
- `bet`: Vector of parameters for the uncertainty component, whose length equals `NCOL(Y)` + 1 to include an intercept term in the model (first entry)  
- `gama`: Vector of parameters for the feeling component, whose length equals `NCOL(W)`+1 to include an intercept term in the model (first entry)

**Value**

A vector of the same length as `ordinal`, whose i-th component is the probability of the i-th rating according to a CUB distribution with given covariates for both uncertainty and feeling, and specified coefficients vectors `bet` and `gama`, respectively.

**References**


**See Also**

`bitgama`, `probcub00`, `probcub0p`, `probcub0q`
Examples

```r
data(relgoods)
m<-10
naord<-which(is.na(relgoods$Physician))
nacov<-which(is.na(relgoods$Gender))
na<-union(naord,nacov)
ordinal<-relgoods$Physician[-na]
W<-Y<-relgoods$Gender[-na]
gama<-c(-0.91,-0.7); bet<-c(-0.81,0.93)
probi<-probcubpq(m,ordinal,Y,W,bet,gama)
```

Description

Probability distribution of an extended CUB model with a shelter effect.

Usage

`probcubshe1(m,pai1,pai2,csi,shelter)`

Arguments

- **m**: Number of ordinal categories
- **pai1**: Mixing coefficient for the shifted Binomial component of the mixture distribution
- **pai2**: Mixing coefficient for the discrete Uniform component of the mixture distribution
- **csi**: Feeling parameter
- **shelter**: Category corresponding to the shelter choice

Details

An extended CUB model is a mixture of three components: a shifted Binomial distribution with probability of success $\xi$, a discrete uniform distribution with support \{1,...,m\}, and a degenerate distribution with unit mass at the shelter category (shelter).

Value

The vector of the probability distribution of an extended CUB model with a shelter effect at the shelter category

References

See Also

probcubshe2, probcubshe3

Examples

```r
m<-8
pai1<-0.5
pai2<-0.3
csi<-0.4
shelter<-6
pr<-probcubshe2(m,pai1,pai2,csi,shelter)
plot(1:m,pr,type="h",main="Extended CUB probability distribution with shelter effect",
 xlab="Ordinal categories")
points(1:m,pr,pch=19)
```

Description

Probability distribution of a CUB model with explicit shelter effect

Usage

```r
probcubshe2(m,pai,csi,delta,shelter)
```

Arguments

- `m` Number of ordinal categories
- `pai` Uncertainty parameter
- `csi` Feeling parameter
- `delta` Shelter parameter
- `shelter` Category corresponding to the shelter choice

Details

A CUB model with explicit shelter effect is a mixture of two components: a CUB distribution with uncertainty parameter \( \pi \) and feeling parameter \( \xi \), and a degenerate distribution with unit mass at the shelter category (\( \text{shelter} \)) with mixing coefficient specified by \( \delta \).

Value

The vector of the probability distribution of a CUB model with explicit shelter effect.
### References


### See Also

`probcubshe1`, `probcubshe3`

### Examples

```r
m<-8
pai1<-0.5
pai2<-0.3
csi<-0.4
shelter<-6
delta<-1-pai1-pai2
pai<-pai1/(1-delta)
pr2<-probcubshe2(m,pai,csi,delta,shelter)
plot(1:m,pr2,type="h", main="CUB probability distribution with explicit shelter effect",xlab="Ordinal categories")
points(1:m,pr2,pch=19)
```

### Description

Probability distribution of a CUB model with explicit shelter effect: satisficing interpretation

### Usage

```r
probcubshe3(m,lambda,eta,csi,shelter)
```

### Arguments

- `m` Number of ordinal categories
- `lambda` Mixing coefficient for the shifted Binomial component
- `eta` Mixing coefficient for the mixture of the uncertainty component and the shelter effect
- `csi` Feeling parameter
- `shelter` Category corresponding to the shelter choice
Details
The "satisficing interpretation" provides a parametrization for CUB models with explicit shelter effect as a mixture of two components: a shifted Binomial distribution with feeling parameter $\xi$ (meditated choice), and a mixture of a degenerate distribution with unit mass at the shelter category ($\text{shelter}$) and a discrete uniform distribution over $m$ categories, with mixing coefficient specified by $\eta$ (lazy selection of a category).

Value
The vector of the probability distribution of a CUB model with shelter effect.

References

See Also
probcubshe1, probcubshe2

Examples
```r
m<-8
pai1<-0.5
pai2<-0.3
csi<-0.4
shelter<-6
lambda<-pai1
et<-1-pai2/(1-pai1)
pr3<-probcubshe3(m,lambda,eta,csi,shelter)
plot(1:m,pr3,type="h",main="CUB probability distribution with explicit shelter effect",xlab="Ordinal categories")
points(1:m,pr3,pch=19)
```

Description
Compute the probability distribution of a CUSH model without covariates, that is a mixture of a degenerate random variable with mass at the shelter category and the Uniform distribution.

Usage
```r
probcush(m,delta,shelter)
```
Arguments

- \( m \)  Number of ordinal categories
- \( \delta \)  Shelter parameter
- \( \text{shelter} \)  Category corresponding to the shelter choice

Value

The vector of the probability distribution of a CUSH model without covariates.

References


Examples

```r
m<-10
shelter<-1
\( \delta \)<-0.4
pr<-probcush(m,\( \delta \),shelter)
plot(1:m,pr,type="h",xlab="Number of categories")
points(1:m,pr,pch=19)
```

probgecub  

*Probability distribution of a GeCUB model*

Description

Compute the probability distribution of a GeCUB model, that is a CUB model with shelter effect with covariates specified for all component.

Usage

```r
probgecub(\text{ordinal},Y,\text{w},X,\text{bet},gama,omega,\text{shelter})
```

Arguments

- \text{ordinal}  Vector of ordinal responses
- \( Y \)  Matrix of covariates for explaining the uncertainty component
- \( \text{w} \)  Matrix of covariates for explaining the feeling component
- \( X \)  Matrix of covariates for explaining the shelter effect
- \text{bet}  Vector of parameters for the uncertainty component, whose length equals NCOL(\( Y \))+1 to include an intercept term in the model (first entry)
Vector of parameters for the feeling component, whose length equals NCOL(W)+1 to include an intercept term in the model (first entry)

Vector of parameters for the shelter effect, whose length equals NCOL(X)+1 to include an intercept term in the model (first entry)

Category corresponding to the shelter choice

A vector of the same length as ordinal, whose i-th component is the probability of the i-th observation according to a GeCUB model with the corresponding values of the covariates for all the components and coefficients specified in bet, gama, omega.


---

**probihg**

*Probability distribution of an IHG model*

**Description**

Compute the probability distribution of an IHG model (Inverse Hypergeometric) without covariates.

**Usage**

`probihg(m,theta)`

**Arguments**

- `m` Number of ordinal categories
- `theta` Preference parameter

**Value**

The vector of the probability distribution of an IHG model.

**References**

**Examples**

```r
m<-10
theta<-0.30
pr<-probihg(m,theta)
plot(1:m,pr,type="h",xlab="Ordinal categories")
points(1:m,pr,pch=19)
```

**Description**

Given a vector of \( n \) ratings over \( m \) categories, it returns a vector of length \( n \) whose \( i \)-th element is the probability of observing the \( i \)-th rating for the corresponding IHG model with parameter \( \theta_i \), obtained via logistic link with covariates and coefficients.

**Usage**

`probihgcvn(m, ordinal, U, nu)`

**Arguments**

- `m` Number of ordinal categories
- `ordinal` Vector of ordinal responses
- `U` Matrix of selected covariates for explaining the preference parameter
- `nu` Vector of coefficients for covariates, whose length equals `NCOL(U)+1` to include an intercept term in the model (first entry)

**Details**

The matrix \( U \) is expanded with a vector with entries equal to 1 in the first column to include an intercept term in the model.

**See Also**

`probihg`

**Examples**

```r
n<-100
m<-7
theta<-0.30
ordinal<simihg(n,m,theta)
U<-sample(c(0,1),n,replace=TRUE)
nu<-c(0.12,-0.5)
pr<-probihgcvn(m,ordinal,U,nu)
```
Relational goods and Leisure time dataset

Description

Dataset consists of the results of a survey aimed at measuring the evaluation of people living in the metropolitan area of Naples, Italy, with respect to relational goods and leisure time collected in December 2014. Every participant was asked to assess on a 10 point ordinal scale his/her personal score for several relational goods (for instance, time dedicated to friends and family) and to leisure time. In addition, the survey asked respondents to self-evaluate their level of happiness by marking a sign along a horizontal line of 110 millimeters according to their feeling, with the left-most extremity standing for "extremely unhappy", and the right-most extremity corresponding to the status "extremely happy".

Usage
data(relgoods)

Format

The description of subjects' covariates is the following:

- ID: An identification number
- Gender: A factor with levels: 0 = man, 1 = woman
- BirthMonth: A variable indicating the month of birth of the respondent
- BirthYear: A variable indicating the year of birth of the respondent
- Family: A factor variable indicating the number of members of the family
- Year: A factor with levels: 1 = if there is any child aged less than 12 in the family, 0 = otherwise
- EducationDegree: A factor with levels: 1 = compulsory school, 2 = high school diploma, 3 = Graduated-Bachelor degree, 4 = Graduated-Master degree, 5 = Post graduated
- MaritalStatus: A factor with levels: 1 = Unmarried, 2 = Married/Cohabitee, 3 = Separated/Divorced, 4 = Widower
- Residence: A factor with levels: 1 = City of Naples, 2 = District of Naples, 3 = Others Campania, 4 = Others Italia, 5 = Foreign countries
- Glasses: A factor with levels: 1 = wearing glasses or contact lenses, 0 = otherwise
- RightHand: A factor with levels: 1 = right-handed, 0 = left-handed
- Smoking: A factor with levels: 1 = smoker, 0 = not smoker
- Walk Alone: A factor with levels: 1 = usually walking alone, 0 = usually walking in company
- Job: A factor with levels: 1 = Not working, 2 = Retired, 3 = occasionally, 4 = fixed-term job, 5 = permanent job
- PlaySport: A factor with levels: 1 = Not playing any sport, 2 = Yes, individual sport, 3 = Yes, team sport
- Pets: A factor with levels: 1 = owning a pet, 0 = not owning any pet
1) Respondents were asked to evaluate the following items on a 10 point Likert scale, ranging from 1 = "never, at all" to 10 = "always, a lot":

**WalkOut**  How often the respondent goes out for a walk
**Parents**  How often respondent talks at least to one of his/her parents
**Meet Relatives**  How often respondent meets his/her relatives
**Association**  Frequency of involvement in volunteering or different kinds of associations/parties, etc
**Rel Friends**  Quality of respondent's relationships with friends
**Rel Neighbours**  Quality of the relationships with neighbors
**Need Help**  Easiness in asking help whenever in need
**Environment**  Level of comfort with the surrounding environment
**Safety**  Level of safety in the streets
**End of Month**  Family making ends meet
**Meet Friend**  Number of times the respondent met his/her friends during the month preceding the interview
**Physician**  Importance of the kindness/sympathy in the selection of respondent's physician

**Happiness**  Each respondent was asked to mark a sign on a 110mm horizontal line according to his/her feeling of happiness (left endpoint corresponding to completely unhappy, right-most endpoint corresponding to extremely happy)

2) The same respondents were asked to score the activities for leisure time listed below, according to their involvement/degree of amusement, on a 10 point Likert scale ranging from 1 = "At all, nothing, never" to 10 = "Totally, extremely important, always":

**Videogames**
**Reading**
**Cinema**
**Drawing**
**Shopping**
**Writing**
**Bicycle**
**Tv**
**Stay With Friend**  Spending time with friends
**Groups**  Taking part to associations, meetings, etc.
**Walking**
**Hand Work**  Hobby, gardening, sewing, etc.
**Internet**
**Sport**
**Social Network**
**Gym**
Quiz  Crosswords, sudoku, etc.
MusicInstr  Playing a musical instrument
GoAroundCar  Hanging out by car
Dog  Walking out the dog
GoOutEat  Go to restaurants/pubs

Details

Period of data collection: December 2014
Mode of collection: questionnaire
Number of observations: 2459
Number of subjects’ covariates: 16
Number of analyzed items: 34
Warning: with a limited number of missing values

Source


---

**simcub**

*Simulation routine for CUB models*

**Description**

Generate $n$ pseudo-random observations following the given CUB distribution.

**Usage**

`simcub(n,m,pai,csi)`

**Arguments**

- `n`  Number of simulated observations
- `m`  Number of ordinal categories
- `pai`  Uncertainty parameter
- `csi`  Feeling parameter

**See Also**

`probcub00`
**Examples**

```r
n<-300
m<-9
pai<-0.4
csi<-0.7
simulation<-simcube(n,m,pai,csi)
plot(table(simulation),xlab="Ordinal categories",ylab="Frequencies")
```

---

**Description**

Generate \( n \) pseudo-random observations following the given CUBE distribution.

**Usage**

```r
simcube(n,m,pai,csi,phi)
```

**Arguments**

- \( n \): Number of simulated observations
- \( m \): Number of ordinal categories
- \( pai \): Uncertainty parameter
- \( csi \): Feeling parameter
- \( phi \): Overdispersion parameter

**See Also**

`probcube`

**Examples**

```r
n<-300
m<-9
pai<-0.7
csi<-0.4
phi<-0.1
simulation<-simcube(n,m,pai,csi,phi)
plot(table(simulation),xlab="Ordinal categories",ylab="Frequencies")
```
Simulation routine for CUB models with shelter effect

Description
Generate \( n \) pseudo-random observations following the given CUB distribution with shelter effect.

Usage
\[
simcubshe(n, m, pai, csi, delta, shelter)
\]

Arguments
- \( n \): Number of simulated observations
- \( m \): Number of ordinal categories
- \( pai \): Uncertainty parameter
- \( csi \): Feeling parameter
- \( delta \): Shelter parameter
- \( shelter \): Category corresponding to the shelter choice

See Also
- `probcubsheQ`, `probcubsheR`, `probcubsheS`

Examples
\[
\begin{align*}
n & \leftarrow 300 \\
m & \leftarrow 9 \\
pai & \leftarrow 0.7 \\
csi & \leftarrow 0.3 \\
delta & \leftarrow 0.2 \\
shelter & \leftarrow 3 \\
simulation & \leftarrow \text{simcubshe}(n, m, pai, csi, delta, shelter) \\
\text{plot(table(simulation),xlab="Ordinal categories",ylab="Frequencies")}
\end{align*}
\]

Simulation routine for CUSH models

Description
Generate \( n \) pseudo-random observations following the distribution of a CUSH model without co-
variates.

Usage
\[
simcush(n, m, delta, shelter)
\]
**Arguments**

- **n**  
  Number of simulated observations
- **m**  
  Number of ordinal categories
- **delta**  
  Shelter parameter
- **shelter**  
  Category corresponding to the shelter choice

**See Also**

- `probcush`

**Examples**

```r
n<-200
m<-7
delta<-0.3
shelter<-3
simulation<-simihg(n,m,delta,shelter)
plot(table(simulation),xlab="Ordinal categories",ylab="Frequencies")
```

---

**simihg**  
*Simulation routine for IHG models*

**Description**

Generate *n* pseudo-random observations following the given IHG distribution.

**Usage**

```r
simihg(n,m,theta)
```

**Arguments**

- **n**  
  Number of simulated observations
- **m**  
  Number of ordinal categories
- **theta**  
  Preference parameter

**See Also**

- `probihg`

**Examples**

```r
n<-300
m<-9
theta<-0.4
simulation<-simihg(n,m,theta)
plot(table(simulation),xlab="Number of categories",ylab="Frequencies")
```
**S3 method: summary for class "GEM"**

**Description**

S3 method summary for objects of class `GEM`.

**Usage**

```r
## S3 method for class 'GEM'
summary(object, correlation = FALSE, ...)
```

**Arguments**

- `object`: An object of class `GEM`
- `correlation`: Logical: should the estimated correlation matrix be returned? Default is FALSE
- `...`: Other arguments

**Value**

Extended summary results of the fitting procedure, including parameter estimates, their standard errors and Wald statistics, maximized log-likelihood compared with that of the saturated model and of a Uniform sample. AIC, BIC and ICOMP indeces are also displayed for model selection. Execution time and number of executed iterations for the fitting procedure are also returned.

**Examples**

```r
model <- GEM(Formula(Mee|tRelatives ~ 0|0), data = relgoods)
summary(model, correlation = TRUE, digits = 4)
```

---

**Evaluation of the Orientation Services 2002**

**Description**

A sample survey on students evaluation of the Orientation services was conducted across the 13 Faculties of University of Naples Federico II in five waves: participants were asked to express their ratings on a 7 point scale (1 = "very unsatisfied", 7 = "extremely satisfied"). Here dataset collected during 2002 is loaded.

**Usage**

```r
data(univer)
```
Format

The description of subjects’ covariates is:

Faculty  A factor variable, with levels ranging from 1 to 13 indicating the coding for the different university faculties
Freqserv A factor with levels: 0 = for not regular users, 1 = for regular users
Age  Variable indicating the age of the respondent in years
Gender  A factor with levels: 0 = man, 1 = woman
Diploma  A factor with levels: 1 = classic studies, 2 = scientific studies, 3 = linguistic, 4 = Professional, 5 = Technical/Accountancy, 6 = others
Residence  A factor with levels: 1 = city NA, 2 = district NA, 3 = others
ChangeFa  A factor with levels: 1 = changed faculty, 2 = not changed faculty

Analyzed ordinal variables (Likert ordinal scale): 1 = "extremely unsatisfied", 2 = "very unsatisfied", 3 = "unsatisfied", 4 = "indifferent", 5 = "satisfied", 6 = "very satisfied", 7 = "extremely satisfied"

Informat  Level of satisfaction about the collected information
Willingn  Level of satisfaction about the willingness of the staff
Officeho  Judgment about the Office hours
Competen  Judgement about the competence of the staff
Global  Global satisfaction

Details

Period of data collection: 2002
Mode of collection: questionnaire
Number of observations: 2179
Number of subjects’ covariates: 7
Number of analyzed items: 5

Source


varcub00  Variance of CUB models without covariates

Description

Compute the variance of a CUB model without covariates.

Usage

varcub00(m,pai,csi)
Arguments

m  Number of ordinal categories
pai  Uncertainty parameter
csi  Feeling parameter

References


See Also

expcube00, probcube00

Examples

```
m<-9
pai<-0.6
csi<-0.5
varcub<-varcub00(m,pai,csi)
```

```
varcube                  Variance of CUBE models without covariates

Description

Compute the variance of a CUBE model without covariates.

Usage

```
varcube(m,pai,csi,phi)
```

Arguments

m  Number of ordinal categories
pai  Uncertainty parameter
csi  Feeling parameter
phi  Overdispersion parameter

References


See Also

probcube, expcube
Examples

```r
m<-7
pai<-0.6
csi<-0.2
phi<-0.05
varianceCUBE<-varcube(m,pai,csi,phi)
```

**Description**

Compute the variance-covariance matrix of parameter estimates for CUB models with or without covariates for the feeling and the uncertainty parameter, and for extended CUB models with shelter effect.

**Usage**

```r
varmatCUB(ordinal,m,param,Y=0,W=0,X=0,shelter=0)
```

**Arguments**

- `ordinal` Vector of ordinal responses
- `m` Number of ordinal categories
- `param` Vector of parameters for the specified CUB model
- `Y` Matrix of selected covariates to explain the uncertainty component (default: no covariate is included in the model)
- `W` Matrix of selected covariates to explain the feeling component (default: no covariate is included in the model)
- `X` Matrix of selected covariates to explain the shelter effect (default: no covariate is included in the model)
- `shelter` Category corresponding to the shelter choice (default: no shelter effect is included in the model)

**Details**

The function checks if the variance-covariance matrix is positive-definite: if not, it returns a warning message and produces a matrix with NA entries. No missing value should be present neither for `ordinal` nor for covariate matrices: thus, deletion or imputation procedures should be preliminarily run.
References


See Also

vcov, cormat

Examples

data(univer)
m<-7
### CUB model with no covariate
pai<-0.87; csi<-0.17
param<-c(pai,csi)
varmat<-varmatCUB(univer$global,m,param)

### and with covariates for feeling
data(univer)
m<-7
pai<-0.86; gama<-c(-1.94,-0.17)
param<-c(pai,gama)
ordinal<-univer$willingn; W<-univer$gender
varmat<-varmatCUB(ordinal,m,param,W)

### CUB model with uncertainty covariates
data(relgoods)
m<-10
naord<which(is.na(relgoods$Physician))
nacov<which(is.na(relgoods$Gender))
na<-union(naord,nacov)
ordinal<-relgoods$Physician[-na]
Y<-relgoods$Gender[-na]
bet<-c(-0.81,0.93); csi<-0.20
varmat<-varmatCUB(ordinal,m,param=c(bet,csi),Y=Y)

### and with covariates for both parameters
data(relgoods)
m<-10
naord<which(is.na(relgoods$Physician))
nacov<which(is.na(relgoods$Gender))
na<-union(naord,nacov)
ordinal<-relgoods$Physician[-na]
W<-Y<-relgoods$Gender[-na]
gama<-c(-0.91,-0.7); bet<-c(-0.81,0.93)
varmat<-varmatCUB(ordinal,m,param=c(bet,gama),Y=Y,W=W)

********************************************
### Description

Compute the variance-covariance matrix of parameter estimates for CUBE models when no covariate is specified, or when covariates are included for all the three parameters.

### Usage

```r
varmatCUBE(ordinal, m, param, Y = 0, W = 0, Z = 0, expinform = FALSE)
```

### Arguments

- `ordinal`: Vector of ordinal responses
- `m`: Number of ordinal categories
- `param`: Vector of parameters for the specified CUBE model
- `Y`: Matrix of selected covariates to explain the uncertainty component (default: no covariate is included in the model)
- `W`: Matrix of selected covariates to explain the feeling component (default: no covariate is included in the model)
- `Z`: Matrix of selected covariates to explain the overdispersion component (default: no covariate is included in the model)
- `expinform`: Logical: if TRUE and no covariate is included in the model, the function returns the expected variance-covariance matrix (default is FALSE: the function returns the observed variance-covariance matrix)

### Details

The function checks if the variance-covariance matrix is positive-definite: if not, it returns a warning message and produces a matrix with NA entries. No missing value should be present neither for `ordinal` nor for covariate matrices: thus, deletion or imputation procedures should be preliminarily run.
References


See Also

vcov, cormat

Examples

```r
m<7; n<500
pai<0.83; csi<0.19; phi<0.045
ordinal<simcube(n,m,pai,csi,phi)
param<c(pai,csi,phi)
varmat<varmatCUBE(ordinal,m,param)

### Including covariates

data(relgoods)
m<10
naord<which(is.na(relgoods$Tv))
nacov<which(is.na(relgoods$BirthYear))
na<union(naord,nacov)
age<2014-relgoods$BirthYear[-na]
lage<log(age)-mean(log(age))
Y<~W~Z~lage
ordinal<relgoods$TV[-na]
estbet<c(0.18,1.03); estgama<c(-0.6,-0.3); estalpha<c(-2.3,0.92)
param<c(estbet,estgama,estalpha)
varmat<varmatCUBE(ordinal,m,param,Y=W=W,Z=Z,expinform=TRUE)
```

vcov.GEM

*S3 method vcov() for class "GEM"*

Description

S3 method: vcov for objects of class GEM.

Usage

```r
## S3 method for class 'GEM'
vcov(object, ...)
```
vcov.GEM

Arguments

  object  An object of class \texttt{GEM}

  \ldots  Other arguments

Value

Variance-covariance matrix of the final ML estimates for parameters of the fitted GEM model. It returns the square of the estimated standard error for CUSH and IHG models with no covariates.

See Also

\texttt{varmatCUB, varmatCUBE, GEM}
Index

*Topic **datasets**
- relgoods, 56
- univer, 62

*Topic **device**
- cubevisual, 11
- cubeshevisual, 12
- cubvisual, 13
- makeplot, 39
- multicub, 40
- multicube, 41

*Topic **distribution**
- betabinomial, 3
- betabinomialcsi, 4
- betar, 5
- bitcsi, 7
- bitgama, 8
- expcub00, 17
- expcube, 18
- probbit, 43
- probcub00, 43
- probcub0q, 44
- probcube, 45
- probcubp0, 46
- probcubpqa, 48
- probcubshe1, 49
- probcubshe2, 50
- probcubshe3, 51
- probcush, 52
- probhcub, 53
- probihg, 54
- probihgcvn, 55
- simcub, 58
- simcube, 59
- simcubshe, 60
- simcush, 60
- simihg, 61
- varcub00, 63
- varcube, 64

*Topic **htest**
- chi2cub, 9
- ellecub, 16
- inibest, 24
- inibestcube, 25
- inibestcubecov, 25
- inibestcubecsi, 26
- inibestgama, 28
- inigrid, 29
- iniihg, 30
- loglikCUB, 33
- loglikCUBE, 35
- loglikCUSH, 36
- loglikIHG, 37
- logscore, 38
- varmatCUB, 65
- varmatCUBE, 67

*Topic **models**
- cormat, 10
- GEM, 19
- makeplot, 39

*Topic **package**
- BIC_.GEM, 6
- coef_.GEM, 10
- CUB_.package, 14
- fitted_.GEM, 19
- loglik_.GEM, 32
- makeplot, 39
- print_.GEM, 42
- summary_.GEM, 62
- vcov_.GEM, 68

*Topic **univar**
- deltaprob, 15
- dissim, 16
- gini, 23
- laakso, 31

*Topic **utilities**
- inibest, 24
- inibestcube, 25
- inibestcubecov, 25
INDEX

inibestcubecsi, 26
inibestgama, 28
inigrid, 29
inihg, 30
logis, 31

betabinomial, 3, 5, 6, 46
betabinomialcsi, 4, 4
betar, 4, 5, 5, 46
BIC, 21, 32
BIC.GEM, 6
bitcsi, 7, 43, 44
bitgama, 8, 45, 47, 48

chi2cub, 9
coeff, 21
coeff.GEM, 10
cormat, 10, 21, 66, 68
CUB_package, 14
cubevisual, 11, 39
cubshevisual, 12, 39
cubvisual, 12, 13, 39
deltaprob, 15
dissim, 16
ellecub, 16
expcub0, 17, 18, 64
expcube, 17, 18, 64

fitted, 21
fitted.GEM, 19

GEM, 6, 10, 19, 19, 32, 37, 38, 40–42, 62, 68, 69
gini, 23, 31

inibest, 24, 28–30
inibestcubecsi, 25, 26, 27, 30
inibestcubecov, 25, 25, 27
inibestcubecsi, 25, 26, 26, 28
inibestgama, 24, 26, 27, 28
inigrid, 29
inihg, 30

laakso, 23, 31
logis, 8, 31
logLik, 6, 21, 33, 35, 37, 38
logLik.GEM, 32
loglikCUB, 17, 32, 33
loglikCUBE, 32, 35

loglikCUSH, 32, 36
loglikGAM, 32, 37, 42
logscore, 38

makeplot, 21, 39
multicub, 12, 39, 40
multicube, 39, 41

plot, 40, 41
plotloglikhg, 42
print.GEM, 42
probit, 43
probcub0, 7, 43, 43, 45, 47, 48, 58, 64
probcub0q, 7, 8, 44, 44, 47, 48
probcube, 45, 59, 64
probcubp, 7, 44, 45, 46, 48
probcubpq, 8, 44, 45, 47, 48
probcubshe1, 49, 51, 52, 60
probcubshe2, 50, 50, 52, 60
probcubshe3, 50, 51, 51, 60
probcush, 52, 61
probgecub, 53
probihg, 54, 55, 61
probihgcn, 55

relgoods, 56

simcub, 58
simcube, 59
simcubshe, 60
simcush, 60
simihg, 61
summary, 10, 21
summary.GEM, 62

text, 40, 41

univer, 62

varcub0, 17, 18, 63
varcube, 17, 18, 64
varmatCUB, 65, 69
varmatCUBE, 67, 69
vcov, 21, 66, 68
vcov.GEM, 68