Comunicaciones en Estadística
Diciembre 2018, Vol. 11, No. 2, pp. 219–238

cubm package in R to fit CUB models

cubm package

Freddy Hernández Barajas[a]
fhernanb@unal.edu.co

Olga Cecilia Usuga Manco[b]
olga.usuga@udea.edu.co

Sebastián García Muñoz[c]
sebastiangm01@gmail.com

Abstract

The class of CUB models is commonly used by practitioners to model ordinal data, in this paper we propose the **cubm** package which provides the class of CUB models in the R system for statistical computing. The **cubm** package allows to specify a formula for each parameter of the model, the Maximum Likelihood (ML) estimation is performed by optimization via the functions `nlminb`, `optim` and `DEoptim` and the variance-covariance matrix can be obtained by numerical approximation of the Hessian matrix or by bootstrap method. The utility of the package is illustrated by an application and a simulation study.

***Keywords*:** CUB models, Feeling and uncertainty, Ordinal data, R.

Resumen

La clase de modelos CUB es usada comunmente por investigadores para modelar datos ordinales. En este artículo se describe el paquete **cubm** que proporciona la clase de modelos CUB en el sistema de computación estadística R. El paquete **cubm** permite especificar una fórmula para cada parámetro del modelo, las estimaciones de máxima verosimilitud se obtienen por medio de optimización através de las funciones `nlminb`, `optim` y `DEoptim` y la matriz de varianza-covarianza se puede obtener por medio de aproximación numérica de la matriz Hessiana o por medio del método bootstrap. La utilidad del paquete se ilustra mediante una aplicación y un estudio de simulación.

***Palabras clave*:** Modelos CUB, Sentimiento e incertidumbre, Datos ordinales, R.

[a] Profesor asistente, Universidad Nacional de Colombia, Sede Medellín.
[b] Profesora asociada, Universidad de Antioquia, Medellín.
[c] Ingeniero Industrial, Universidad de Antioquia, Medellín.

1. Introduction

The usual statistical approach for modeling ordinal data has been Generalized Linear Models (GLM). When data are collected as ordered responses to a sequence of items (concerning preferences, evaluations, proficiency, etc), current literature has developed a vast amount of results known as Item Response Theory, Iannario (2010). In the last thirteen years, a different approach has been introduced for explaining the behavior of respondents when faced to a single item characterized by ordinal choices, this class of statistical models known as CUB (Combined Uniform and Binomial) has been derived by Piccolo (2003*b*), Piccolo (2003*a*), and D'Elia & Piccolo (2005). The model is based on a mixture model that is able to express the stated evaluation via the subject's covariates. Specifically, it examines and compares the uncertainty of the answer and the feeling towards the items.

The recent interest in well-being measurements, initially developed in behavioral contexts, has inspired the application of CUB models to the selection of response categories in a number of research areas. In the food industry, for example, studies have been developed to evaluate preferences and satisfaction from the use of CUB models. Iannario et al. (2012) performed a sensory analysis in the food industry in order to obtain useful information for marketing management; Piccolo et al. (2013) studied the importance that respondents assign to a list of intrinsic and extrinsic attributes and the level of agreement that consumers express with a number of statements concerning extra virgin olive oil; Boatto et al. (2016) carried out a study to detect segments of markets based on consumption opinions, purchase characteristics and price of Parmesan cheese, and Arboretti & Bordignon (2016) conducted an investigation with the objective of evaluating the preferences of fresh food packaging. Other studies have been developed in areas such as education and work, Cafarelli & Crocetta (2016) evaluated the student satisfaction in a Faculty of Economics in Foggia, Gambacorta & Iannario (2013) modeled job satisfaction based on data collected in a Survey in Italy and Capecchi (2015) measured the experience of conflict between personal and organizational ethnics in a large sample of respondents.

Due to the use of the model in many contexts, several authors have developed generalizations, and have improved the initial model. Iannario (2008) specified the statistical implications of dummy covariates by emphasizing the interpretation of the estimated parameters, Iannario (2010) studied the identifiability of the CUB model, Corduas (2011) proposed a test procedure in order to compare CUB models, Innario (2012) and Iannario (2014) studied ordered categorical data with overdispersion in the framework of CUB models, Capecchi & Piccolo (2014) and Grilli et al. (2014) studied and applied latent class CUB models, Oberski & Vermunt (2015) showed the equivalence of loglinear latent class models and CUB models and Piccolo (2015) studied inferential issues on CUB models with covariate.

Some generalizations of the CUB models have been introduced, Iannario (2012) proposed a Hierarchical CUB models which is a generalization in which parameters are allowed to be random, and Manisera & Zuccolotto (2013, 2014, 2015, 2016) generalized CUB models by introducing a new class of models, called Nonlinear CUB, which are able to describe precision processes. The CUB models was implemented in the **CUB** package (Iannario et al. 2016) written in R system for statistical computing (R Core Team (2017)). The implementation of CUB models relies on one Formula interface (Zeileis & Croissant, 2010), the Maximum Likelihood (ML) estimation is performed by classical EM procedures (McLachlan & Krishnan, 1997) and the optimization procedure is run via the `optim` function.

Although there is already a package in R for the analysis of CUB models, the **cubm** package proposed in this paper gives other options to the user to estimate the parameters and variance-covariance matrix, and also allows the user to define a formula for each parameter of the model. In this paper, we describe the **cubm** package which can be used to estimate parameters. The package is implemented in R system for statistical computing (R Core Team (2017)) and it is available from the `GitHub` repository. Implementation of CUB models relies on the formula interface of GAMLSS package (Rigby & Stasinopoulos, 2005), allowing to specify a formula for each parameter, the first one for the feeling and the second one for the uncertainty. **cubm** package fits the CUB models using ML estimation trough different optimization methods: a bounds constrained quasi-Newton method (`nlminb`), the Differential Evolution algorithm for global optimization of a real-valued function of a real-valued parameter vector (`DEoptim`, (Mullen et al., 2011)), a relatively robust method that does not require derivatives (`optim`: Nelder-Mead), a low-memory optimizer for unconstrained problems with large numbers of parameters (`optim`: CG), a simple unconstrained variable metric/quasi-Newton method (`optim`: BFGS), a modest-memory optimizer for bounds constrained problems (`optim`: L-BFGS-B) and a stochastic method that does not require derivatives (`optim`: SANN). The variance-covariance matrix can be obtained by numerical approximation to the Hessian matrix (`Hessian`: numDeriv, (Gilbert & Varadhan, 2016)). If the variance-covariance matrix is not positive definite the procedure uses bootstrap method to estimate the standard deviation of the parameters.

In the remainder of this manuscript we elaborate on **cubm**'s implementation and use. In Section 2 the notation for CUB models is introduced. Section 3 describes the **cubm** package. Then, Section 4 shows an application. A simulation study is presented in Section 5. Some concluding remarks end the paper.

2. CUB models

The class of CUB models is built on the basic assumption that, when a subject is asked to express a rating about a given issue on an ordered response scale with m categories, his/her response derives from the combination of a *feeling* attitude towards the evaluated issue and an intrinsic *uncertainty* component surrounding the discrete choice. CUB models fit rating data by means of a mixture of two random variables, namely a Shifted Binomial $V(m, \xi)$ with trial parameter m and success probability $1 - \xi$, modelling the feeling component, and a discrete

Uniform $U(m)$ defined over the support $\{1,\dots,m\}$, aimed to model the uncertainty component (Manisera & Zuccolotto 2015). The random variable R represents the observed ratings and has probability mass function given by

$$\Pr(R=r|\boldsymbol{\theta}) = \pi\binom{m-1}{r-1}\xi^{m-r}(1-\xi)^{r-1} + (1-\pi)\frac{1}{m}, \qquad (1)$$

where $r=1,\dots,m$, with $\boldsymbol{\theta}=(\pi,\xi)^\top$, $\pi\in(0,1]$ and $\xi\in[0,1]$.

On left panel of Figure 1 we can find the probability mass function for four π values and $\xi=0.70$ with $m=5$. The $1-\pi$ quantity measures uncertainty that accompanies the choice, for small values of π, the value of $1-\pi$ increases and the form of the distribution tends to be uniform. For the case of $\pi=0.01$ we can see that black line is constant around 0.2. As π increases, $1-\pi$ decreases and the shape of the probability mass function changes, when $\pi=0.99$ the cub distribution in expression (1) corresponds to a shifted binomial distribution with success probability $1-\xi$. On right panel of Figure 1 we can observe the probability mass function for four ξ values and $\pi=0.99$. For small values of ξ, the value of $1-\xi$ increases and the distribution tends to give more probability for high values of R, at the contrary, for high values of ξ, the value of $1-\xi$ decreases and the distribution tends to give more probability for lower values of R.

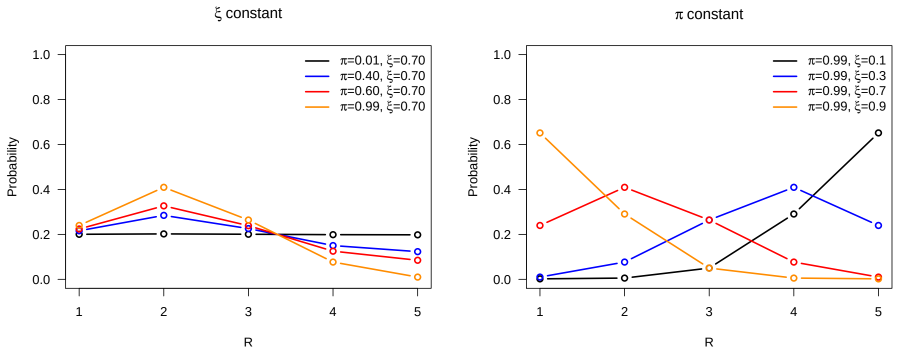

Figure 1: Probability mass function for $\xi=0.70$, different values of π and $m=5$ on left panel, probability mass function for $\pi=0.99$, different values of ξ and $m=5$ on right panel.

2.1. Parameter estimation without covariates

Suppose a group of n individuals are asked to rate a product service in a scale from 1 to m. Let R a random variable and $r_1,r_2,\dots,r_n$ the ratings given by the individuals. If $R\sim\text{CUB}(\pi,\xi,m)$, the likelihood function of the CUB model is:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \Pr(R = r_i|\boldsymbol{\theta}), \tag{2}$$

where $\boldsymbol{\theta} = (\pi, \xi,)^\top$ and the associated log-likelihood function is:

$$\begin{aligned} \ell(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log \Pr(R = r_i|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \log \left[\pi \binom{m-1}{r_i - 1} \xi^{m-ri} (1-\xi)^{ri-1} + (1-\pi)\frac{1}{m} \right] \end{aligned} \tag{3}$$

There are not closed forms for maximum likelihood estimates of π and ξ, therefore we need to use numerical methods (Iannario & Piccolo 2012).

2.2. Parameter estimation with covariates

Suppose a group of n individuals are asked to rate a product service in a scale from 1 to m. Suppose also that for each individual there is additional information such as age, marital status, salary, among others. This additional information corresponds to t variables denoted by $X_1, X_2, X_3, \ldots, X_t$. The Table 1 shows a summary of the information available in a CUB model with covariates.

Individual	Score	X_1	X_2	X_3	$\cdots$	X_t
1	r_1	x_{11}	x_{21}	x_{31}	$\cdots$	x_{t1}
2	r_2	x_{12}	x_{22}	x_{32}	$\cdots$	x_{t2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
n	r_n	x_{1n}	x_{2n}	x_{3n}	$\cdots$	x_{tn}

Table 1: Illustration of the information of a CUB model with covariates

Assuming that the responses of the n individuals are distributed CUB(π_i, ξ_i, m), it is possible to model the parameters π and ξ using subsets of the t covariates shown in the Table 1.

The parameter π for the i-th individual can be modeled using p of the t covariates as follows:

$$g(\pi_i) = \boldsymbol{\beta}^\top \boldsymbol{Z_i}, \tag{4}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^\top$ and $\boldsymbol{Z_i} = (1, x_{1i}, x_{2i}, \ldots, x_{pi})^\top$. The link function $g(\cdot)$ ensures that the values of $\boldsymbol{\beta}^\top \boldsymbol{Z_i}$ lies within the $(0, 1]$ interval. The most commonly choices for the link functions are logit and probit.

In a similar way, the parameter ξ for the i-th individual can be modeled using q of the t covariates as follows:

$$g(\xi_i) = \boldsymbol{\gamma}^\top \boldsymbol{W_i}, \tag{5}$$

where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_q)^\top$ and $\boldsymbol{W_i} = (1, x_{1i}, x_{2i}, \ldots, x_{qi})^\top$.

Substituting the expressions 4 and 5 in the expression 1, we have the following probability mass function:

$$\begin{aligned} P(R = r_i|\boldsymbol{Z}_i, \boldsymbol{W}_i, \boldsymbol{\beta}, \boldsymbol{\gamma}) &= \pi_i \binom{m-1}{r_i-1} (1-\xi)^{r_i-1} \xi^{m-r_i} + (1-\pi_i)\frac{1}{m} \\ &= g^{-1}\left(\boldsymbol{\beta}^\top \boldsymbol{Z_i}\right) \binom{m-1}{r_i-1} \left(1 - g^{-1}\left(\boldsymbol{\gamma}^\top \boldsymbol{W_i}\right)\right)^{r_i-1} \\ &\quad g^{-1}\left(\boldsymbol{\gamma}^\top \boldsymbol{W_i}\right)^{m-r_i} + \left(1 - g^{-1}\left(\boldsymbol{\beta}^\top \boldsymbol{Z_i}\right)\right)\frac{1}{m} \end{aligned} \tag{6}$$

The likelihood function of the CUB model with covariates is given by:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \Pr(R = r_i|\boldsymbol{Z}_i, \boldsymbol{W}_i, \boldsymbol{\beta}, \boldsymbol{\gamma}), \tag{7}$$

with $\Pr(R = r_i|\boldsymbol{Z}_i, \boldsymbol{W}_i, \boldsymbol{\beta}, \boldsymbol{\gamma})$ defined in 6 and $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\gamma})^\top$.

Again for this case we do not have closed forms to estimate $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, so we need to use numerical methods.

3. cubm package

In this section we present the **cubm** package and some useful functions made in R to fit cub models through maximum likelihood estimation.

3.1. Installation

The current version of the **cubm** package is hosted in github which is a web-based Git repository hosting service. For installation the user need to use the next code that automatically install the **devtools** package necessary to download the **cubm** package.

```
if (!require('devtools')) install.packages('devtools')
devtools::install_github('fhernanb/cubm', force=TRUE)
require(cubm)  # To load the package
```

3.2. `dcub` function

The `dcub` function is used to obtain the probabilities for a cub model given the parameters π, ξ and m as in expression (1). The structure of the function is as follows

```
dcub(x, pi, xi, m, log = FALSE)
```

The arguments for the function are:

- `x`: vector of quantiles.
- `pi`: uncertainty parameter belongs to $(0, 1]$.
- `xi`: feeling parameter belongs to $[0, 1]$.
- `m`: the maximum value for the response variable.

The following code calculates the probability for the cub($\pi = 0.4$, $\xi = 0.7$, $m = 5$) distribution shown in Figure 1.

```
dcub(x=1:5, pi=0.4, xi=0.7, m=5)
## [1] 0.21604 0.28464 0.22584 0.15024 0.12324
```

3.3. `rcub` function

The `rcub` function is used to generate random values from a cub distribution given the parameters π, ξ and m. The structure of the function is as follows

```
rcub(n, pi, xi, m = 5)
```

The arguments for the `cub` function are:

- `n`: number of observations.
- `pi`: uncertainty parameter belongs to $(0, 1]$.
- `xi`: feeling parameter belongs to $[0, 1]$.
- `m`: the maximum value for the response variable.

The following code generates 20 observations for cub($\pi = 0.4$, $\xi = 0.7$, $m = 5$) distribution. The `set.seed` function is used to create random numbers that can be reproduced by the user.

```
set.seed(12345)
rcub(n=20, pi=0.4, xi=0.7, m=5)
## [1] 3 2 5 4 2 1 2 2 3 3 4 1 3 1 2 2 5 2 4 4
```

3.4. `cub` function

The `cub` function is used to estimate the parameters via maximum likelihood for a cub model with or without explanatory variables. The structure of the `cub` function is as follows:

```
cub(pi.fo, xi.fo, m, data=NULL, optimizer="nlminb",
    pi.link="probit", xi.link="probit")
```

The arguments for the `cub` function are:

- `pi.fo`: a "formula" object for π parameter with two parts using a tilde operator to separate the dependent variable y from the independent variables. For example, `y` $\sim$ `x1 + x2` is interpreted as modeling `y` as a linear function of `x1` and `x2`, it can be included interactions and polynomials.
- `xi.fo`: a "formula" object for ξ parameter without left part. For example, $\sim$ `x1 + x2` means that we want to model ξ parameter as a linear function of `x1` and `x2`.
- `m`: maximum value for the response variable, it must be an integer.
- `shift`: minimum value for the response variable, by default is 1.
- `data`: an optional data frame with the response and independent variables.
- `optimizer`: optimizer to find the parameter vector, by default is `nlminb` but are available `optim` and `DEoptim` from **DEoptim** package created by Mullen et al. (2011) that implements the global optimization by differential evolution (Ardia et al. 2011).
- `pi.link`: link function for model π parameter, could be `"probit"` or `"logit"`, by default is `"probit"`.
- `xi.link`: link function for model ξ parameter, could be `"probit"` or `"logit"`, by default is `"probit"`.

3.4.1. Example 1

In the next example we simulate 1000 observations from a cub($\pi = 0.4$, $\xi = 0.7$, $m = 5$) using the seed 1234. The `cub` function is used with the formula `pi.fo = y` $\sim$ `1` to indicate the response variable is `y`, the $\sim$ `1` in the formula of `pi.fo` and `xi.fo` indicates that we do not have explanatory variables to model π neither ξ. The parameter vector for this example is $\mathbf{\Theta} = (\pi = 0.4, \, \xi = 0.7)^\top$.

```
set.seed(1234)
y <- rcub(n=1000, pi=0.4, xi=0.7, m=5)
mod <- cub(pi.fo=y~1, xi.fo=~1, m=5, optimizer='nlminb')
```

To obtain the summary table for the model `mod` we can use:

```
summary(mod)
## ---------------------------------------------------------------
## Fixed effects for probit(pi)
## ---------------------------------------------------------------
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.29616    0.11592 -2.5549  0.01062 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---------------------------------------------------------------
## Fixed effects for probit(xi)
## ---------------------------------------------------------------
## Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.55050    0.06407  8.5922 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---------------------------------------------------------------
```

From the last result we obtain the estimates -0.29616 and 0.55050 for π and ξ respectively. Note that in the summary table there is the information about the link function used, for this reason, to obtain the estimated values we need to calculate $\Phi(-0.29616) = 0.38355$ and $\Phi(0.55050) = 0.70901$ that match with the true values $\pi = 0.4$ and $\xi = 0.7$, respectively.

3.4.2. Example 2

In the next example we are going to simulate 1000 random variables y_i following the cub model:

$$
\begin{aligned}
y_i &\sim \text{cub}(\pi_i, \, \xi_i, \, m = 5) \\
\Phi^{-1}(\pi_i) &= \beta_0 + \beta_1 x_1 \\
\Phi^{-1}(\xi_i) &= \gamma_0 + \gamma_1 x_2 \\
x_1 &\sim U(0, 1) \\
x_2 &\sim U(0, 1)
\end{aligned}
\tag{8}
$$

where the parameter vector is $\mathbf{\Theta} = (\beta_0 = -1, \beta_1 = 1, \gamma_0 = -2, \gamma_1 = 1.5)^\top$. To simulate the 1000 random variables we fixed the seed to obtain the same outputs, the code below can be used to simulate the y_i.

```
n <- 1000; m <- 5
b0 <- -1; b1 <- 1
g0 <- -2; g1 <- 1.5
set.seed(123) ; x1 <- runif(n)
set.seed(124) ; x2 <- runif(n)
pi <- pnorm(b0 + b1 * x1) # Using probit link function
xi <- pnorm(g0 + g1 * x2) # Using probit link function
set.seed(12345); y <- rcub(n=n, pi=pi, xi=xi, m=m)
```

Now, to estimate the parameter vector $\mathbf{\Theta}$ for the cub model (8) we can use the next code.

```
mod <- cub(pi.fo=y~x1, xi.fo=~x2, m=5, optimizer='optim')
summary(mod)
## ---------------------------------------------------------------
## Fixed effects for probit(pi)
## ---------------------------------------------------------------
## Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -0.96030    0.16687 -5.7547 8.680e-09 ***
## x1           1.22862    0.25633  4.7931 1.642e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---------------------------------------------------------------
## Fixed effects for probit(xi)
## ---------------------------------------------------------------
## Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -2.29608    0.26155 -8.7789 < 2.2e-16 ***
## x2           1.84114    0.35999  5.1145 3.147e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---------------------------------------------------------------
```

The `cub` function returns an object of class cub and the `summary` function can be applied to this object to obtain an usual summary table. From the summary we obtain that $\hat{\mathbf{\Theta}} = (-0.96, 1.23, -2.30, 1.84)^\top$ which is close to the true parameter vector $\mathbf{\Theta} = (-1, 1, -2, 1.5)^\top$.

Note that in last examples we used two different optimizers, `nlminb` and `optim`, by default `cub` function uses `nlminb`.

4. Simulation study

In this section we present the results from a simulation study to explore the parameter estimation procedures for a model without and with covariates. The scenarios considered here were taken from the structure given in the application from last section.

4.1. Simulation study without covariates

To study the estimation procedure without covariates we considered the next model.

$$\begin{aligned} global_i &\sim \text{cub}(\pi,\, \xi,\, m=7),\ \text{ with } i = 1, 2, \ldots, n \\ \pi &= 0.87 \\ \xi &= 0.17 \end{aligned} \tag{9}$$

We considered sample size values of $n = 10, 20, \ldots, 490, 500$. For each n we simulated 1000 samples to estimate π and ξ, then we calculated the mean for $\hat{\pi}$ and $\hat{\xi}$. To estimate the values of π and ξ we considered the optimizers `nlminb`, `optim` and `DEoptim` available in the `cub` function.

The results from this simulation study are shown in the Figure 2. From this figure we note that the sequence of $\hat{\pi}$ and $\hat{\xi}$ obtain by the `cub` function goes to the real value very quickly. We observe that even for small samples, the mean estimated parameter is very close the target value given by the dotted line. It can be observed that for $n \leq 30$, the blue and orange lines for `nlminb` and `DEoptim` respectively, are quite similar.

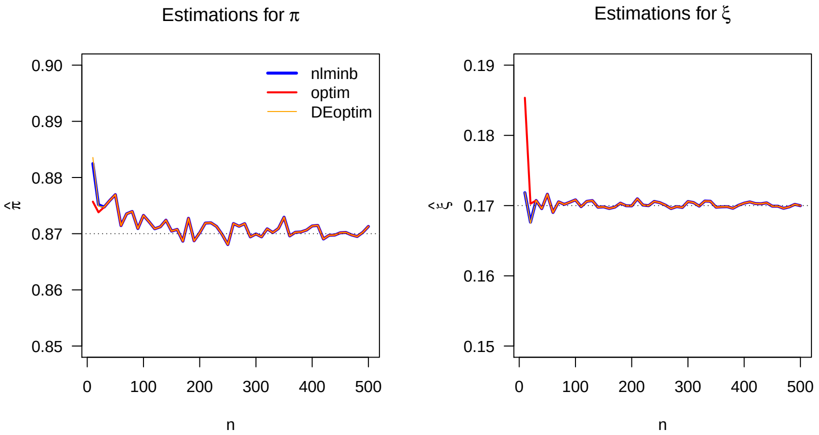

Figure 2: Mean value for $\hat{\pi}$ and $\hat{\xi}$ versus sample size n given the optimizer `nlminb`, `optim` and `DEoptim`. Dotted lines represent true parameter values.

4.2. Simulation study with covariates

To study the estimation procedure with covariates we considered the next model

$$\begin{aligned} global_i &\sim \text{cub}(\pi_i,\, \xi_i,\, m=7),\ \text{ with } i=1,2,\ldots,n \\ \Phi^{-1}(\pi_i) &= \beta_0 + \beta_1 \times gender_i \\ \Phi^{-1}(\xi_i) &= \gamma_0 + \gamma_1 \times lage_i \end{aligned} \tag{10}$$

The fixed parameters assumed values of $\beta_0 = 0.93$, $\beta_1 = 0.47$, $\gamma_0 = -0.95$ and $\gamma_1 = -0.33$. The values for the covariates $gender_i$ and $lage_i$ where taken from the original dataset *univer*. We considered sample size values of $n = 10, 20, \ldots, 490, 500$. For each n we simulated 1000 samples to estimate the parameters, then we calculatde the mean for each estimated parameter. In the same way as in the previous case, we considered the optimizers `nlminb`, `optim` and `DEoptim` available in `cub` function. The results from this simulation study are shown in the Figure 3.

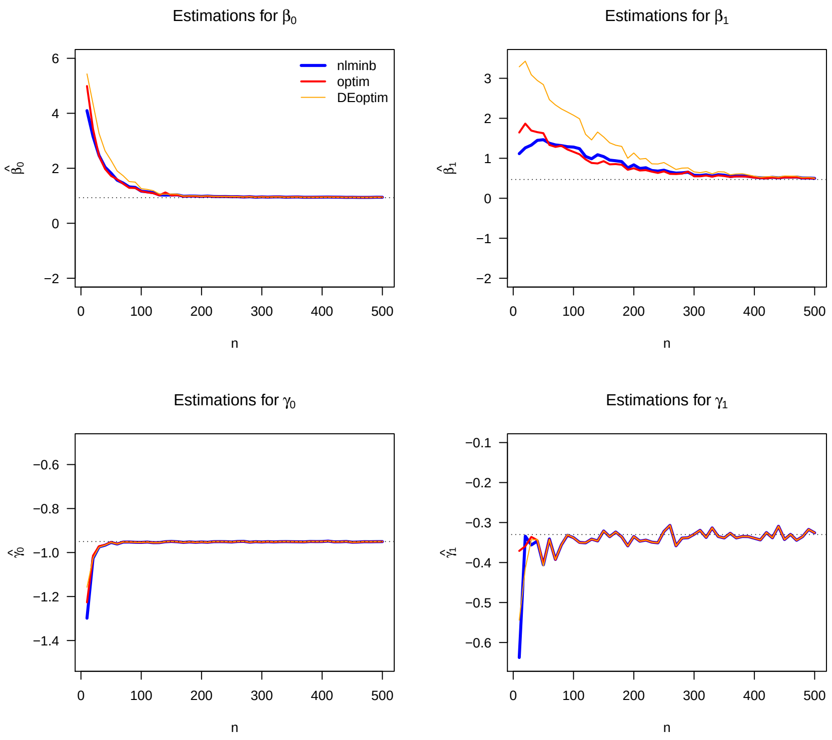

Figure 3: Mean value for $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\gamma}_0$ and $\hat{\gamma}_1$ versus sample size n given the optimizer `nlminb`, `optim` and `DEoptim`. Dotted lines represent true parameter values.

From the Figure 3 we note that the sequences for $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\gamma}_0$ and $\hat{\gamma}_1$ tend to go to the real value (in dotted line) as n increases. It can be observed that mean estimations

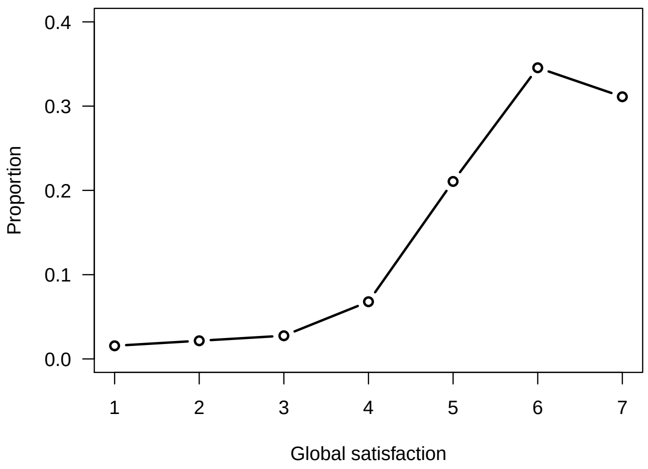

Figure 4: Observed proportion for global satisfaction.

for the intercepts in π and ξ have less variability than mean estimations for the slopes. From this figure is clear that maximum likelihood estimations for each parameter go to the real value as n increases.

5. Application

In this section we re-analyzed the *univer* data from Iannario et al. (2016) related to a sample survey conducted in 13 faculties of University of Naples in Italy. The participants were asked to express their opinion about orientation services on a 7 point scale (1 = very unsatisfied, 7 = extremely satisfied). The *univer* data has 12 variables (faculty, freqserv, age, gender, diploma, residenc, changefa, informat, willingn, officeho, compete, and global) and 2179 observations, the variable called *global* corresponds to the response variable related to global satisfaction. Figure 4 shows the observed proportion for global satisfaction, from this figure we note that almost 35 % of the participants rated the orientation services with 6 and 31 % of the participants rated the services with a 7, the participants tend to measure the service with high values.

The model considered here can be summarised as:

$$\begin{aligned} global_i &\sim \text{cub}(\pi,\, \xi,\, m = 7) \\ \Phi^{-1}(\pi) &= \beta_0 \\ \Phi^{-1}(\xi) &= \gamma_0 \end{aligned} \tag{11}$$

To estimate the π and ξ parameters for the global satisfaction variable with the proposed **cubm** package we created the model `mod0` using the next code.

```
require(cubm)  # Loading the cubm package
mod0 <- cub(pi.fo=global~1, xi.fo=~1, m=7, data=univer)
summary(mod0)
```

The results for `mod0` are shown below. From this output we found that $\Phi^{-1}(\hat{\pi}) = 1.118$ which implies that $\hat{\pi} = 0.868$, and in a similar way we found that $\hat{\xi} = 0.171$. The model `mod0` has a log-likelihood value of -3245.474 with two parameters. In Figure 5 we observed the proportion for global satisfaction in black line and the estimated proportion in red line, we can note that the estimated curve follows the observed curve.

```
## ---------------------------------------------------------------
## Fixed effects for probit(pi)
## ---------------------------------------------------------------
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  1.11810    0.05888  18.989 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---------------------------------------------------------------
## Fixed effects for probit(xi)
## ---------------------------------------------------------------
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -0.948689   0.016254 -58.367 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---------------------------------------------------------------
```

Iannario et al. (2016) considered a cub model to explain the parameters π and ξ as a function of *gender* and *lage* respectively, the model can be summarised as:

$$\begin{aligned} global_i &\sim \text{cub}(\pi_i,\, \xi_i,\, m = 7) \\ \Phi^{-1}(\pi_i) &= \beta_0 + \beta_1 \times gender_i \\ \Phi^{-1}(\xi_i) &= \gamma_0 + \gamma_1 \times lage_i \end{aligned} \tag{12}$$

where *global* is the response variable with values from 1 to 7, $gender = 0$ corresponds to man and $gender = 1$ corresponds to woman, *lage* is the transformation of age variable given by $lage_i = \log(age_i) - \overline{\log(age)}$.

To fit the model (12) with the proposed **cubm** package we used the next code.

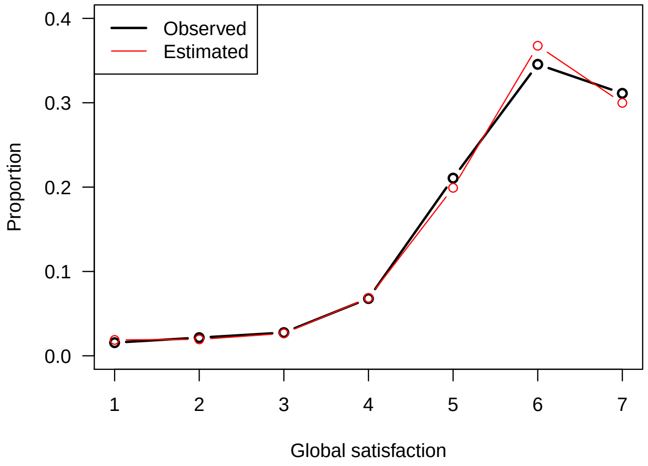

Figure 5: Observed and estimated proportion (with $\hat{\pi} = 0.868$ and $\hat{\xi} = 0.171$) for global satisfaction.

```
mod <- cub(pi.fo=global˜gender, xi.fo=˜lage, m=7,
           data=univer, optimizer='optim')
summary(mod)
```

The results obtained from the summary table for model `mod` are shown next. From this output we note that the two variables considered *gender* and *lage* were significant, this model has a log-likelihood value of -3233.465 with four parameters.

```
## ------------------------------------------------------------------
## Fixed effects for probit(pi)
## ------------------------------------------------------------------
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.926428   0.074291 12.4702 < 2.2e-16 ***
## gender1     0.467235   0.119935  3.8957  9.79e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ------------------------------------------------------------------
## Fixed effects for probit(xi)
## ------------------------------------------------------------------
##              Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) -0.949215   0.016243 -58.4399 < 2.2e-16 ***
## lage        -0.325423   0.108465  -3.0003  0.002698 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ------------------------------------------------------------------
```

From summary table we can write the fitted model as:

$$global_i \sim \mathrm{cub}(\hat{\pi}_i,\, \hat{\xi}_i,\, m = 7)$$

$$\Phi^{-1}(\hat{\pi}_i) = 0.926428 + 0.467235 \times gender_i \tag{13}$$

$$\Phi^{-1}(\hat{\xi}_i) = -0.949215 - 0.325423 \times lage_i \tag{14}$$

From expression (13) we obtain that the uncertain parameter $1 - \hat{\pi}$ is 0.1771118 for male and 0.0817097 for female, this means that female tends to respond the survey with less uncertain than male.

From expression (14) we obtain that the feeling parameter is $1-\hat{\xi} = \Phi(0.949215 + 0.325423 \times lage)$, this means that the age is related to feeling in a positive way, elder people tend to response the survey with more feeling than young people. Figure 6 shows the relation between feeling $(1 - \xi)$ with age (left panel) and the transformed age *lage* (right panel), from both panels we can note that the feeling increases as age increases.

Figure 7 shows estimated probabilities for global satisfaction given gender and two selected ages, 25 and 45 years old. From this figure we confirm that older participants tend to rate the service with upper values and we also note that there is not evidence of greater uncertain in the responses because the individual uncertains are 0.17 and 0.08 for male and women respectively.

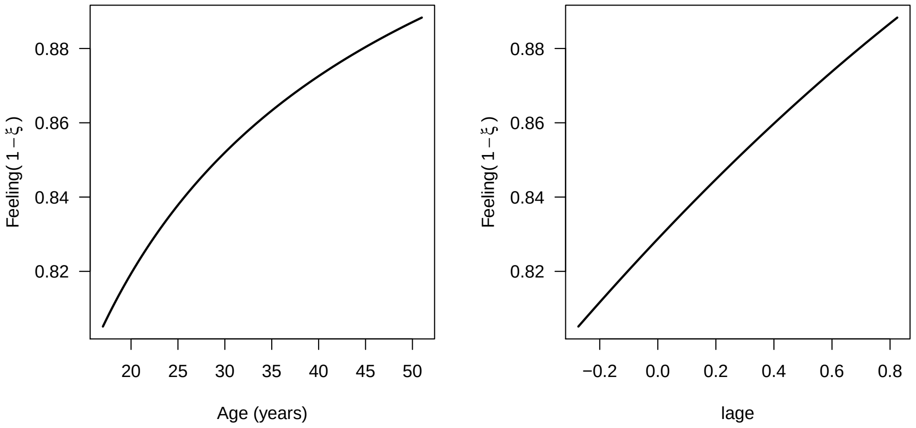

Figure 6: Relation between feeling parameter $(1 - \pi)$ with age (left panel) and with transformed age (rigth panel).

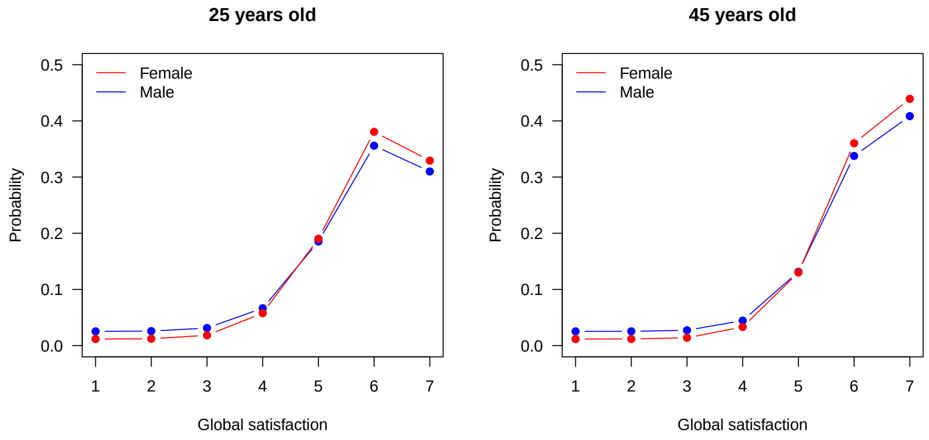

Figure 7: Estimated probabilities for participants with 25 years old (left) and 45 years old (right).

6. Conclusions

This paper addressed the CUB models by the **cubm** package. We have presented the CUB models, described how the user can model ordered responses using **cubm** package and we have presented practical examples. From the simulation study we conclude that **cubm** package fits CUB models in a correct way. Future researches and implementations in R will focus on residuals, adjustment measures, influence measures and Bayesian estimation.

Recibido: 2017-10-15
Aceptado: 2018-12-11

References

Arboretti, R. & Bordignon, P. (2016), 'Consumer preferences in food packaging: Cub models and conjoint analysis', *British Food Journal* **118**(3), 527–540.

Ardia, D., Boudt, K., Carl, P., Mullen, K. M. & Peterson, B. G. (2011), 'Differential Evolution with DEoptim: An application to non-convex portfolio optimization', *The R Journal* **3**(1), 27–34.

Boatto, V., Rossetto, L., Bordignon, P., Arboretti, R., Salmaso, L., Griffith, C. & Griffith, C. (2016), 'Cheese perception in the north american market: empirical evidence for domestic vs. imported parmesan', *British Food Journal* **118**(7).

Cafarelli, B. & Crocetta, C. (2016), An evaluation of the student satisfaction based on cub models, *in* 'Topics in Theoretical and Applied Statistics', Springer, pp. 73–83.

Capecchi, S. (2015), 'Modelling the perception of conflict in working conditions', *Electronic Journal of Applied Statistical Analysis* **8**(3), 298–311.

Capecchi, S. & Piccolo, D. (2014), Modelling the latent components of personal happiness, *in* 'Mathematical and Statistical Methods for Actuarial Sciences and Finance', Springer, pp. 49–52.

Corduas, M. (2011), Assessing similarity of rating distributions by kullback-leibler divergence, *in* 'Classification and Multivariate Analysis for Complex Data Structures', Springer, pp. 221–228.

D'Elia, A. & Piccolo, D. (2005), 'A mixture model for preference data analysis', *Computational Statistics Data Analysis* **49**(3), 917–934.

Gambacorta, R. & Iannario, M. (2013), 'Measuring job satisfaction with cub models', *Labour* **27**(2), 198–224.

Gambacorta, R., Iannario, M. & Valliant, R. (2014), ‘Design-based inference in a mixture model for ordinal variables for a two stage stratified design’, *Australian & New Zealand Journal of Statistics* **56**(2), 125–143.

Gilbert, P. & Varadhan, R. (2016), *numDeriv: Accurate Numerical Derivatives.* R package version 2016.8-1.
*https://CRAN.R-project.org/package=numDeriv

Grilli, L., Iannario, M., Piccolo, D. & Rampichini, C. (2014), ‘Latent class cub models’, *Advances in Data Analysis and Classification* **8**(1), 105–119.

Iannario, M. (2008), ‘Dummy covariates in cub models’, *Statistica* **68**(2), 179–200.

Iannario, M. (2010), ‘On the identifiability of a mixture model for ordinal data’, *Metron* **68**(1), 87–94.

Iannario, M. (2012), ‘Hierarchical cub models for ordinal variables’, *Communications in Statistics-Theory and Methods* **41**(16-17), 3110–3125.

Iannario, M. (2014), ‘Modelling uncertainty and overdispersion in ordinal data’, *Communications in Statistics-Theory and Methods* **43**(4), 771–786.

Iannario, M., Manisera, M., Piccolo, D. & Zuccolotto, P. (2012), ‘Sensory analysis in the food industry as a tool for marketing decisions’, *Advances in Data Analysis and classification* **6**(4), 303–321.

Iannario, M. & Piccolo, D. (2010), ‘A new statistical model for the analysis of customer satisfaction’, *Quality Technology & Quantitative Management* **7**(2), 149–168.

Iannario, M. & Piccolo, D. (2012), ‘Cub models: Statistical methods and empirical evidence’, *Modern Analysis of Customer Surveys* pp. 231–258.

Iannario, M. & Piccolo, D. (2015), ‘A generalized framework for modelling ordinal data’, *Statistical Methods & Applications* pp. 1–27.

Iannario, M., Piccolo, D. & Simone, R. (2016), *CUB: A Class of Mixture Models for Ordinal Data.* R package version 1.0.
*https://CRAN.R-project.org/package=CUB

Innario, M. (2012), ‘Cube models for interpreting ordered categorical data with overdispersion’, *Quaderni di statistica* **14**(14), 137–140.

Manisera, M. & Zuccolotto, P. (2013), ‘Nonlinear cub models: some stylized facts’, *Quaderni di Statistica* **15**, 111–130.

Manisera, M. & Zuccolotto, P. (2014), ‘Modeling rating data with nonlinear cub models’, *Computational Statistics & Data Analysis* **78**, 100–118.

Manisera, M. & Zuccolotto, P. (2015), ‘Visualizing multiple results from nonlinear cub models with r grid viewports’, *Electronic Journal of Applied Statistical Analysis* **8**(3), 360–373.

Manisera, M. & Zuccolotto, P. (2016), 'Treatment of "don't knowresponses in a mixture model for rating data', *METRON* **74**(1), 99–115.

McLachlan, G. & Krishnan, T. (1997), 'The em algorithm and extensions'.

Mullen, K., Ardia, D., Gil, D., Windover, D. & Cline, J. (2011), 'DEoptim: An R package for global optimization by differential evolution', *Journal of Statistical Software* **40**(6), 1–26.
*http://www.jstatsoft.org/v40/i06/

Oberski, D. & Vermunt, J. (2015), 'The cub model and its variations are restricted loglinear latent class models', *EJASA* .

Piccolo, D. (2003*a*), 'Computational issues in the em algorithm for ranks model estimation with covariates', *Quaderni di Statistica* **5**, 1–22.

Piccolo, D. (2003*b*), 'On the moments of a mixture of uniform and shifted binomial random variables', *Quaderni di Statistica* **5**(1), 85–104.

Piccolo, D. (2015), 'Inferential issues on cube models with covariates', *Communications in Statistics-Theory and Methods* **44**(23), 5023–5036.

Piccolo, D., Capecchi, S., Iannario, M. & Corduas, M. (2013), 'Modelling consumer preferences for extra virgin olive oil: the italian case', *Politica Agricola Internazionale - International Agricultural Policy* p. 25.

R Core Team (2017), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
*https://www.R-project.org/

Rigby, R. A. & Stasinopoulos, D. M. (2005), 'Generalized additive models for location, scale and shape,(with discussion)', *Applied Statistics* **54**, 507–554.

Zeileis, A. & Croissant, Y. (2010), 'Extended model formulas in r: Multiple parts and multiple responses', *Journal of Statistical Software* **34**(1), 1–13.