The R package CUB:  
a class of mixture models for ordinal rating data

Maria Iannario  Domenico Piccolo  Rosaria Simone  
University of Naples Federico II  University of Naples Federico II  University of Naples Federico II

Abstract

The R package CUB has been developed for the analysis of ordinal rating data with a class of finite mixture distributions whose main feature is the probabilistic specification of the decision-making process as the combination of uncertainty and feeling components. Extensions include models for overdispersion, inflated categories and large heterogeneity occurrences. The parameter estimation procedure is based on maximum likelihood methods, with the implementation of the Expectation-Maximization (EM) algorithm. Several features of package CUB, including simulation routines, are presented on two data sets loaded within the package.

Keywords: Rating data, CUB models, CUBE models, Shelter effect, CUSH models, IHG models.

1. Introduction

Ordered categorical data are commonly used in scientific fields where respondents express a graduated evaluation on a specific item. Examples include ratings of preference in consumer studies, degree of pain measured on a visual analogue scale, numerical or verbal rating scale, sensory evaluation on food perception and appreciation, self evaluation of well-being, job satisfaction, economic perceived conditions and so on. Cumulative link models are a suitable class of models for such data since the ordered nature of observation is exploited and the flexible regression framework allows good fitting performances. These models with a logit link are widely known as the proportional odds model due to McCullagh (1980). The cumulative models are also known as ordinal regression models although the latter term is used to indicate other models for ordinal responses such as continuation ratio models and adjacent category models (see e.g. Agresti (2010); Tutz (2012)). Other alternatives with the indication of the logit and probit link functions are ordered logit models and ordered probit models (Greene and Hensher 2010).

Several R packages or functions for implementing these statistical models for ordered factor responses have been introduced, mainly addressing methods based on latent variables linked to observed categorical responses via threshold models. Among others, we quote polr() in MASS package (Venables and Ripley 2002), fitting logistic or probit regression models, for instance. In rms the function lrm() is available (Harrell 2009) for binary and proportional odds logistic regression; in VGAM the function vglm() (Yee 2010) to fit vector generalized linear models, whereas in Zelig it is possible to exploit zelig() to estimate several statistical models; in mnet the function multinom() allows to fit multinomial log-linear models (Ripley and Venables
2016) and in **Ordinal** the function `clm()` (Christensen 2015) fits general cumulative link models. For a Bayesian perspective, Markov Chain Monte Carlo for ordered probit regression are implemented in the package `MCMCpack` via calls to `MCMCoprobit()` (Martin et al. 2009).

An alternative approach is the use of models anchored to a structured discrete probability distribution (D’Elia and Piccolo 2005) with a close relationship to the data generating process. More specifically, a two-component mixture is designed to shape ordinal observations as a weighted combination of a preference component and an uncertainty component, each controlled by a given distribution.

The statistical appeal of these models relies on several features, among which: a sharp relationship with the data generating process producing ordinal data; parsimony in the parameter estimation; effective graphical devices useful for the interpretation, and direct inclusion of respondents’ characteristics to specify how they relate with the two components.

This class of models stems from the baseline Combination of a discrete Uniform and a shifted Binomial random variable -CUB models- firstly introduced by Piccolo (2003) and D’Elia and Piccolo (2005), where two latent components, called feeling and uncertainty, are jointly considered to specify the mixture model. General references for the statistical background include Iannario (2012a,b, 2015); Iannario and Piccolo (2010a). A detailed list of alternative models is presented in Iannario and Piccolo (2016a); see also Tutz et al. (2016). Robustness issues for CUB models are discussed in Iannario et al. (2016b).

Further methodological studies have been published about CUB models, among others Corduas et al. (2009); Gambacorta et al. (2014); Grilli et al. (2014) for a latent class version; Iannario (2012c,d, 2014); Manisera and Zuccolotto (2014a) for including don’t know options also with covariates or for assuming non-linear spacing between adjacent categories: Manisera and Zuccolotto (2014b). In addition, applications have been performed in many disciplines, ranging from labour economics (Gambacorta and Iannario 2013) to demography (Iannario and Piccolo 2010b), happiness economics (Capecchi and Piccolo 2014), linguistics (Balirano and Corduas 2008), marketing (Iannario et al. 2012), medicine (D’Elia 2008), sensory science (Piccolo and D’Elia 2008; Capecchi et al. 2015). Proposals for the generalization of the CUB model in a bivariate perspective are discussed by Corduas (2015). Moreover, Andreis and Ferrari (2013), Colombi and Giordano (2016) introduced a general multivariate setting.

This article deals with the package `CUB` written in the R environment and available from CRAN at [http://CRAN.R-project.org](http://CRAN.R-project.org). It is presented in its first consistent update, but the implemented program has been long experienced among scholars and researchers working with CUB models and their extensions.

The paper is organized as follows. Section 2 briefly reviews **GEneralized Mixture Models with uncertainty (GEM)** for ordinal data stemming from CUB models. In Section 3 the main functions and features of package `CUB` are illustrated by means of case studies based on two data-sets loaded within the package. Finally, extra features of GEM models and future extensions are briefly discussed in Section 4.

Notice that package `CUB`, although performs statistical analysis for the whole class of GEM models, is named after the original proposal.

2. **GEM models specification**
The basic idea of GEM framework is to model the ordinal responses \((R_1, \ldots, R_n)\), where \(R_i\) is given by the \(i\)-th subject, \(i = 1, \ldots, n\), with respect to a given item \(I\), on the basis of two main components related to the decisional process. The first component \((C_1)\) related to the feeling (attraction, satisfaction, awareness towards the item): in the baseline CUB model, it is modelled by means of the shifted Binomial random variable. The second one \((C_2)\) concerns the uncertainty (indecision, blurriness, fuzziness) and it is modelled by means of a discrete Uniform distribution over the support \(\{1, \ldots, m\}\). Here and in the sequel, \(m\) will denote the number of ordinal categories, and it is assumed that \(m > 3\) for identifiability purposes (Iannario 2010). All available information on subjects’ characteristics (covariates) are collected in a matrix \(T\), and \(Y\) and \(W\) are submatrices of \(T\) reporting values of respondents’ covariates useful to explain uncertainty and feeling components, respectively.

Let \(\theta = (\beta', \gamma')'\) be the parameter vector characterizing the distribution of \((R_1, \ldots, R_n)\), with \(\beta', \gamma'\) denoting the parameter vector for the uncertainty and feeling components, respectively. Then, the mixture regression model has the following form:

\[
Pr(R_i = r | y_i, w_i, \theta) = \pi_i Pr(C_{1i} = r | w_i) + (1 - \pi_i) Pr(C_{2i} = r),
\]

for \(i = 1, \ldots, n\) and \(j = 1, \ldots, m\), where \(\pi_i = \pi(y_i, \beta) \in (0, 1]\) are introduced to weight the two components and \(w_i \in W, y_i \in Y\) include the selected covariates for the \(i\)-th subject. They are related to the parameters by means of a logit link, a not compulsory but preferred choice for easiness of interpretation and robustness properties (Iannario et al. 2017):

\[
\text{logit}(1 - \pi_i) = -\beta' y_i; \quad \text{logit}(1 - \xi_i) = -\gamma' w_i.
\]

The implementation of GEM models relies on the Formula interface (Zeileis and Croissant 2010), allowing to specify different regressor matrices for the different parameters. The following call will estimate a GEM model for given ordinal observations:

\[
\text{GEM(Formula, family, ...)}
\]

where Formula is an object of class Formula, and family indicates which sub-class of models has to be fitted. Optional inputs for the specification of the models can be passed via the ... argument, which serves also to modify some technical parameters needed for the estimation procedure. For instance, it is possible to change the maximum number of iterations allowed for running the optimization algorithm (maxiter=500 by default) and the error tolerance for final estimates (toler=1e-6 by default). The argument data is optional. The number of categories \(m\) is, in general, internally retrieved but it is advisable to pass it as an argument to the call if some category has zero frequency.

As mentioned, the benchmark model is the CUB mixture model, where the preference part is modelled according to a shifted Binomial random variable:

\[
b_r(\xi_i) = \binom{m-1}{r-1} \xi_i^{m-r} (1 - \xi_i)^{r-1}, \quad r = 1, 2, \ldots, m
\]

with feeling parameters given by \(\xi_i, i = 1, \ldots, n\), as in (2).

The probability distribution chosen to model uncertainty is the discrete Uniform, thus it does not imply estimable parameters ensuring model parsimony. An increase of \(\pi_i\) implies a reduced
impact of the uncertainty component; thus, the quantity \((1 - \pi_i)\) is a (normalized) measure of the uncertainty implied by the model. Alternative subjective choices of distributions for this latent component are proposed and motivated in Gottard et al. (2016).

In the simplest version, the GEM model works on the ratings given to a single item and provides a global measure of feeling and uncertainty for the whole population under study by letting \(\pi_i = \pi\) and \(\xi_i = \xi\), that is, by considering no knowledge about subjects’ characteristics. The corresponding probability distribution can be computed via \texttt{probcub00(m, pai, csi)}. Other extensions which are implemented within package CUB concern the ability to take into account the presence of inflated categories and the overdispersion effect.

### 2.1. Inflated CUB models

Inflated CUB models imply the presence of a shelter effect (Iannario 2012a) located at category \(s \in \{1, \ldots, m\}\).

CUB models are thus extended with the introduction of a degenerate random variable \(D_r^{(s)} = I(R = s)\), whose probability mass is concentrated at \(r = s\), as a third component in (1). The function returns the estimation of parameters related both to the specification:

\[
Pr(R = r \mid \theta) = \pi_1 b_r(\xi) + \pi_2 \frac{1}{m} + (1 - \pi_1 - \pi_2) D_r^{(s)},
\]

and to the alternative formulation:

\[
Pr(R = r \mid \theta) = \delta \left[ D_r^{(s)} \right] + (1 - \delta) \left[ \pi^* b_r(\xi) + (1 - \pi^*) \frac{1}{m} \right], \quad r = 1, 2, \ldots, m.
\]

They are equivalent specifications since:

\[
\begin{cases}
\pi_1 = \pi^*(1 - \delta) \\
\pi_2 = (1 - \pi^*)(1 - \delta)
\end{cases} \iff \begin{cases}
\pi^* = \frac{\pi_1}{\pi_1 + \pi_2} \\
\delta = 1 - \pi_1 - \pi_2.
\end{cases}
\]

The corresponding probability distributions can be computed via \texttt{probcubshe1(m, pai1, pai2, csi, shelter)} for (3) and \texttt{probcubshe2(m, pai, csi, delta, shelter)} for (4), respectively.

For the parameterization (4), the estimable parameter vector is \(\theta = (\pi^*, \xi, \delta)'\); thus it is immediate to quantify the shelter effect by means of the parameter \(\delta\). Moreover, the modification of the uncertainty component so induced is evaluated by comparing the \(\pi\) parameter in model (1) with \(\pi_1\) in model (3). A further specification of the shelter effect adheres to the so-called satisficing behaviour, implemented via \texttt{probcubshe3(m, lambda, eta, csi, shelter)}. Covariates in the model can be specified for all parameters by considering generalized CUB models (\(g\cub\)) (Iannario and Piccolo 2016b).

The examination of real phenomena where responses are affected by an uncertainty level close to the maximum and with inflation in just one category suggested the introduction of CUSH models (Combination of a Uniform distribution and a SHelter component) (Capecchi et al. 2015; Capecchi and Piccolo 2016; Capecchi and Iannario 2016), whose distribution is especially useful in dealing with sample surveys affected by large heterogeneity (several examples are observed in the analysis of activities for leisure times, see the discussion below). In other
words, CUSH models are nested into GEcUB ones when the shifted Binomial component tends to have zero weight (Capecchi and Iannario 2016): indeed, the distribution without covariates may be obtained by (4) when \( \pi^* = 0 \). The routine \texttt{probcush(m, delta, shelter)} computes the corresponding probability distribution:

\[
Pr (R = r \mid \theta) = \delta D_r^{(s)} + (1 - \delta) \frac{1}{m}, \quad r = 1, 2, \ldots, m,
\]

where \( \theta = \delta \), the weight of the shelter effect. If \( X \) is a submatrix of \( T \) specifying the observations on selected covariates for explaining the shelter effect, then the link: \( \text{logit}(\delta_i) = \omega' x_i, i = 1, \ldots, n \), will specify parameters for each respondent profile (here, \( \omega' \) includes an intercept term).

### 2.2. Overdispersion

CUB models are not suitable to account for a possible extra-variability among responses, since location and variability of a Binomial random variable are jointly and severely constrained. Thus, a parsimonious extension has been proposed by introducing CUBE models (Iannario 2014, 2015). Here, the shifted Binomial distribution for \( Pr (C_{1i} = r) \) is replaced by a shifted Beta-Binomial distribution over the support \( \{1, \ldots, m\} \):

\[
g_r(\xi, \phi) = \binom{m-1}{r-1} \frac{\prod_{k=1}^{r} (1 - \xi + \phi (k - 1)) \prod_{k=1}^{m-r+1} (\xi + \phi (r - 1))}{(1 - \xi + \phi (r - 1))(\xi + \phi (m - r)) \prod_{k=1}^{m-1} (1 + \phi (k - 1))}, \quad r = 1, 2, \ldots, m.
\]

(see \texttt{betar()}). Thus, the probability distribution of a CUBE model -computed by \texttt{probcube(m, pai, csi, phi)}- is given by:

\[
Pr (R = r \mid \theta) = \pi g_r(\xi, \phi) + (1 - \pi) \frac{1}{m}, \quad r = 1, 2, \ldots, m.
\]

The additional parameter \( \phi \) which characterizes the distribution is a direct measure of overdispersion (Iannario 2016). When \( \phi = 0 \), a CUBE model reduces to a CUB one; thus, the latter is nested into the first one: this implies that the improvement of the fit of CUBE over CUB models for given data can be easily established by likelihood ratio tests.

Covariates may be introduced for explaining all the three parameters or only the feeling component of the CUBE mixture (Piccolo 2015). The logarithmic link is customarily used for the overdispersion parameter: \( \log(\phi_i) = \alpha' z_i \) where \( z_i \in Z \), and \( Z \) is a submatrix of \( T \) (\( \alpha' \) includes an intercept term). Feeling and uncertainty components are related to subjects’ characteristics as in (2).

It has been remarked (Iannario 2012b) that CUBE models with:

\[
\pi = 1; \quad \xi = \frac{(m - 1) \theta}{1 + m(\theta - 2)}; \quad \phi = \frac{1 - \theta}{1 + \theta(m - 2)};
\]

correspond to IHG models (D’Elia 2003) with \( \theta \in [0, 1] \), where IHG stands for Inverse Hypergeometric distribution. These models arise from unimodal extreme-feeling distributions with no uncertainty. In this case, for the \( i \)-th respondent, a parameter \( \theta_i \) characterizes the
model and gives the probability of observing a rating corresponding to the first/last category: therefore, $\theta_i$ is a direct measure of preference, attraction, pleasantness towards the investigated item. The routine `probing(m, theta)` computes the IHG probability distribution over $m$ categories:

$$Pr(R = r | \gamma) = \theta(1 - \theta)^{r-1} \frac{m - 1}{m} \prod_{l=1}^{r} \frac{m - l + 1}{m - l + \theta(l - 1)} \quad r = 1, \ldots, m.$$  

Also for this sub-class of models, it is possible to study response patterns among respondents by including subjects’ covariates to explain the preference parameter $\theta_i$, $i = 1, \ldots, n$, via a logit link: $\text{logit}(\theta_i) = \nu' u_i$, where $u_i \in U$ and $U$ is a submatrix of $T$ ($\nu'$ includes an intercept term).

After this self-contained overview of the statistical methodology grounded on GEM models, next section provides guidelines for the main calls to fit GEM models on given data when using package CUB.

### 2.3. Main calls

As already specified, the baseline GEM model is the CUB distribution. To fit a CUB model to given ordinal data arranged in a vector `ordinal`, the most general call to run is:

```r
GEM(Formula(ordinal~Y|W|X), family="cub", ...)
```

where $Y$, $W$, $X$ indicate the model matrices of explanatory variables for uncertainty, feeling and possible shelter components, respectively. If, for all components, no covariate is included in the model, then $Y=0$, $W=0$, $X=0$ need to be specified. For instance

```r
GEM(Formula(ordinal~0|0|0), family="cub")
```

fits a CUB model with no covariates for given `ordinal` data. Then,

```r
GEM(Formula(ordinal~Y|0|0), family="cub")
```

considers only the covariate matrix $Y$ to explain the uncertainty component. Similarly,

```r
GEM(Formula(ordinal~0|W|0), family="cub")
```

includes the covariate matrix $W$ to explain only the feeling component, whereas:

```r
GEM(Formula(ordinal~Y|W|0), family="cub")
```

specifies a CUB model with covariates for both components.

To test for a shelter effect, it is possible to implement either a CUB model without covariates by specifying the value of the shelter $s$ in the main call to `GEM()`:
or to implement the more comprehensive specification concerning GeCUB models:

\[
\text{GEM(} \text{Formula(ordinal} \sim 0|0|0), \text{ family}=\text{"cub"}, \text{ shelter}=s)\]

CUSH models can be fitted by the main function \text{GEM()}, optionally with the specification of covariates, by specifying \text{family} argument to "cush":

\[
\text{GEM(} \text{Formula(ordinal} \sim Y|W|X), \text{ family}=\text{"cush"}, \text{ shelter}=s)\]

\[
\text{GEM(} \text{Formula(ordinal} \sim 0), \text{ family}=\text{"cush"}, \text{ shelter} = s) \quad \# \text{ without covariates}
\]
\[
\text{GEM(} \text{Formula(ordinal} \sim X), \text{ family}=\text{"cush"}, \text{ shelter} = s) \quad \# \text{ with covariates}
\]

If \(Y, W\) and \(Z\) are matrices of values of selected covariates for explaining uncertainty, feeling and overdispersion, respectively, CUBE models can be estimated via:

\[
\text{GEM(} \text{Formula(ordinal} \sim 0|0|0), \text{ family}=\text{"cube"}) \quad \# \text{ without covariates}
\]
\[
\text{GEM(} \text{Formula(ordinal} \sim 0|W|0), \text{ family}=\text{"cube"}) \quad \# \text{ with covariates for feeling}
\]
\[
\text{GEM(} \text{Formula(ordinal} \sim Y|W|Z), \text{ family}=\text{"cube"}) \quad \# \text{ with covariates for all parameters}
\]

For CUBE model with no covariates, the default option \text{expinform} = \text{FALSE} implies that the variance-covariance matrix of estimates is based on the observed information matrix. If \text{expinform} = \text{TRUE} is added as an extra argument, the procedure considers the expected information matrix instead. Note that the fitting procedure for CUBE models with covariates for all components might be particularly slow: hence, in the first runs, we tentatively suggest to lower the tolerance and the maximum number of iterations allowed for the optimization algorithm by specifying \text{maxiter} and \text{toler} arguments in the function call (for instance, \text{maxiter}=50, \text{toler}=1e-3). Alternatively, one might estimate a random sub-sample of data and adopt the resulting estimates as preliminary values for a \text{GEM} fit to the whole data.

Finally, estimation of IHG models is implemented via:

\[
\text{GEM(} \text{Formula(ordinal} \sim 0), \text{family}=\text{"ihg"}) \quad \# \text{ without covariates}
\]
\[
\text{GEM(} \text{Formula(ordinal} \sim U), \text{family}=\text{"ihg"}) \quad \# \text{ with covariates}
\]

Every call to the main \text{GEM} creates an object of the class "\text{GEM}" -to which the subclass specified by the \text{family} argument is appended. Next sections provide details on inferential issues for \text{GEM} models, and the corresponding implemented \text{S3} methods.

### 3. Implementation and inference in \text{CUB}

Package \text{CUB} fits the models presented in Section 2 using Maximum Likelihood (ML) estimation as performed by classical EM procedures (McLachlan and Krishnan 1997). For every unspecified detail about \text{GEM} models, see Iannario and Piccolo (2016a). Further, details about the EM steps for \text{CUB} models can be found in Piccolo (2006), for \text{CUBE} models in Iannario (2014); Piccolo (2015), and for \text{GeCUB} models in Iannario and Piccolo (2016b). For \text{CUSH}
models, the reader is referred to Capecchi and Piccolo (2016) for a full discussion, whereas for IHG see D’Elia (2003).

3.1. Method implementation

For an object of class "GEM", storing the information of the fitting procedure of a GEM model to given data, the standard S3 method logLik() applies to return the maximized log-likelihood at the final ML estimates. Thus, in the baseline version of a GEM model, parameters are estimated by maximizing the log-likelihood:

\[ \ell(\theta) = \sum_{i} \log \left[ Pr \left( R_i = r | y_i, w_i, \theta \right) \right] . \]

The optimization procedure is run via the function optim(), except for CUB models without covariates (for which the EM steps are directly implemented: D’Elia and Piccolo (2005)) and for CUSH models without covariates (for which the ML estimate of the shelter parameter can be straightforwardly computed, see Capecchi and Piccolo (2016)). Whenever bounds on parameter estimates have to be required, the box-constrained option for optim() is selected, and it considers a restriction of the admissible range of the parameter space: in this case, method = "L-BFGS-B" is selected for multidimensional optimization problems (with finite difference approximation selected with the option gr = NULL), while method = "Brent" is selected for the unidimensional optimization problem when IHG models with no covariate are fitted. In all other cases, the default Nelder-Mead method is selected.

In addition, for given ordinal data and model-specific parameters, log-likelihood values can be computed via the following main calls (the default option is that neither covariate nor shelter effect are included):

```r
llCUB <- loglikCUB(ordinal,m,param,Y=Y,W=W,X=X,shelter=s)
llCUBE <- loglikCUBE(ordinal,m,param,Y=Y,W=W,Z=Z)
llIHG <- loglikIHG(ordinal,m,param,U=U)
llCUSH <- loglikCUSH(ordinal,m,param,X=X,shelter=s)
```

The vector param lists the parameters for the specified model, with the uncertainty parameters always specified first, followed by the feeling ones and, last, by the overdispersion or shelter parameters when considering CUBE or CUB models with shelter effect, respectively. For IHG and CUSH models, param is a one-dimensional vector when no covariates are considered, otherwise it lists all the regression coefficients for the preference and the shelter parameter, respectively. When covariates are included in the model, the first component of the corresponding (sub)vector is the intercept term (not to be included in the matrix), followed by as many parameters as the number of covariates used to explain the given component.

The variance-covariance matrix of final ML estimates for a fitted model can be retrieved by applying the standard S3 method vcov() to an object of class GEM. For testing IHG models with or without covariates, for CUSH models with covariates as well as for CUBE models with covariates for all the three parameters, the option hessian = TRUE is selected when calling optim() to return a numerically differentiated Hessian matrix at the final estimates. For the other models, the code implements specific inferential results, as the explicit formulae of
the observed and expected variance-covariance matrices (Piccolo 2006, 2015; Iannario 2012a, 2014).

To compute the (observed) variance-covariance matrix corresponding to given ordinal data and model-specific parameters (for CUB and CUBE models), the following functions are available:

\[
\text{varCUB} \leftarrow \text{varmatCUB}(\text{ordinal}, \ m, \ \text{param}, \ Y = \text{Ycovar}, \ W = \text{Wcovar}, \ X = \text{Xcovar}, \ \text{shelter} = \text{shelter})
\]

\[
\text{varCUBE} \leftarrow \text{varmatCUBE}(\text{ordinal}, \ m, \ \text{param}, \ Y = \text{Ycovar}, \ W = \text{Wcovar}, \ Z = \text{Zcovar}, \ \text{expinform} = \text{FALSE})
\]

No covariate is considered neither for \text{varmatCUB()} nor for \text{varmatCUBE()} by default. In addition, default options for \text{varmatCUB} do not consider \textit{shelter effect}. For \text{varmatCUBE()}, the default option \text{expinform} = \text{FALSE} provides the variance-covariance matrix based on the observed information matrix. Setting \text{expinform} = \text{TRUE}, the variance-covariance matrix will be based on the expected information matrix instead (as advisable when running simulation studies).

Other standard S3 methods implemented for \text{GEM} objects include:

- \text{coef()} to retrieve final parameter estimates;
- \text{logLik()} to obtain the maximized log-likelihood at the final ML estimates;
- \text{BIC()} to compute the BIC index of the fitted model (in order to compare non-nested models (Schwarz 1978));
- \text{fitted()} to display fitted probability distribution.
- \text{summary()} to display summary results of the fitting procedures:
  1. ML parameter estimates of the fitted model, asymptotic standard errors and Wald-tests (based on the asymptotic variance-covariance matrix as returned by \text{vcov()});
  2. For testing purposes, a list of likelihood-based measures (Log-likelihood, Mean Log-likelihood, and for models without covariates also Log-likelihood of the saturated model, Deviance, Log-likelihood of the Uniform and shifted Binomial);
  3. BIC, AIC (Akaike 1974) and ICOMP (Bodzogan 1990) measures for the fitted model, in order to compare non-nested models.

Every call to the main function \text{GEM()} stores the following slots to record some further fitting information:

- \$\text{time}$: the time employed for the estimation procedure;
- \$\text{niter}$: the number of iterations required to the EM algorithm for convergence within the fixed tolerance.
- \$\text{formula}$: to retrieve the Formula object given as input.
Parameter correlation matrix corresponding to an object of class GEM can be computed via `cormat()`, or printed out by specifying `correlation=TRUE` when calling `summary()`. Table 1 summarizes the main calls to function GEM to fit the different models. Possible extra arguments to include in the call will be further illustrated and discussed along with the examples.

3.2. Plot facilities

One of the advantageous features of GEM models is that they allow for easy interpretation of parameters. This is especially convincing for models with no covariates or for (at most) one dichotomous variable, a case for effective graphical devices.

For an object of class GEM, the generic function `makeplot(object)` will return a plot which compares fitted probabilities and observed relative frequencies. Moreover, for CUB models, given the one-to-one correspondence between the probability distribution of `R` and the parameter vector \( \theta = (\pi, \xi)' \), a relevant feature of the approach is the possibility to visualize a CUB model as a point in the parameter space, that is the unit square. To this scope, the user is referred to functions `cubvisual()`, `multicub()` and `cubshevisual()`. Similar features are performed by `cubevisual()` and `multicube()` for CUBE models.

Other instances of graphical analysis will be shown along with discussion of empirical evidence in Section 4.

4. Package CUB in use

Some basics routines that can help in the comprehension of CUB models and their extensions are presented. The following code produces Figure 4.1, in which the plots of CUBE probability mass functions for varying parameters are shown, as in Iannario (2015). Figure 4.1 displays also the expectation and the variance of the given CUBE distributions, computed via `expcube()` and `varcube()`, respectively. Similar functions for CUB models are `expcub00()` and `varcub00()`.

```r
###########################################################################
### Selection of 9 CUBE models with csi = 0.3 over 9 ordinal categories ***
###########################################################################
m<-9; csi<-0.3
########### varying pai and phi parameters
paival<-seq(0.9,0.1,by=-0.1)
phival<-rep(c(0.05,0.1,0.3),times=3)
model<-cbind(paival,phival); nmodels<-nrow(model)
##########################################
par(mfrow = c(3,3))
par(mar = c(2,4,3,1)+0.1)
### Probability distribution plots for jmod=1,2,...,9
for (jmod in 1:nmodels){
paij<-model[jmod, 1]
phi<-model[jmod, 2]
paij<-model[jmod, 1]
phi<-model[jmod, 2]
```
### Table 1: Summary of possible calls to GEM()

<table>
<thead>
<tr>
<th>Model</th>
<th>Call</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CUB models</strong></td>
<td></td>
</tr>
<tr>
<td>with no covariates</td>
<td>GEM(\text{Formula}(\text{ordinal}~0</td>
</tr>
<tr>
<td>with no covariates and shelter effect</td>
<td>GEM(\text{Formula}(\text{ordinal}~0</td>
</tr>
<tr>
<td>with covariate matrix ( Y ) for uncertainty</td>
<td>GEM(\text{Formula}(\text{ordinal}~Y</td>
</tr>
<tr>
<td>with covariate matrix ( W ) for feeling</td>
<td>GEM(\text{Formula}(\text{ordinal}~0</td>
</tr>
<tr>
<td>with covariate matrices ( Y, W ) for uncertainty and feeling</td>
<td>GEM(\text{Formula}(\text{ordinal}~Y</td>
</tr>
<tr>
<td>with covariate matrices ( Y, W, X ) for uncertainty, feeling and shelter effect</td>
<td>GEM(\text{Formula}(\text{ordinal}~Y</td>
</tr>
<tr>
<td><strong>CUBE models</strong></td>
<td></td>
</tr>
<tr>
<td>with no covariates</td>
<td>GEM(\text{Formula}(\text{ordinal}~0</td>
</tr>
<tr>
<td>with covariate matrix ( W ) for feeling</td>
<td>GEM(\text{Formula}(\text{ordinal}~0</td>
</tr>
<tr>
<td>with covariate matrices ( Y, W, Z ) for uncertainty, feeling and overdispersion</td>
<td>GEM(\text{Formula}(\text{ordinal}~Y</td>
</tr>
<tr>
<td><strong>CUSH models</strong></td>
<td></td>
</tr>
<tr>
<td>with no covariates</td>
<td>GEM(\text{Formula}(\text{ordinal}~0), family=&quot;cush&quot;, shelter=shelter)</td>
</tr>
<tr>
<td>with covariate matrix ( X ) for shelter effect</td>
<td>GEM(\text{Formula}(\text{ordinal}~X), family=&quot;cush&quot;, shelter=shelter)</td>
</tr>
<tr>
<td><strong>IHG models</strong></td>
<td></td>
</tr>
<tr>
<td>with no covariates</td>
<td>GEM(\text{Formula}(\text{ordinal}~0), family=&quot;ihg&quot;)</td>
</tr>
<tr>
<td>with covariate matrix ( U ) for the preference parameter</td>
<td>GEM(\text{Formula}(\text{ordinal}~U), family=&quot;ihg&quot;)</td>
</tr>
</tbody>
</table>
The R package CUB

```r
prob<-probcube(m, paij, csi, phij)
exp<-expcube(m, paij, csi, phij)
var<-round(varcube(m, paij, csi, phij), digits = 2)
plot(1:m, prob, type = "h", lwd = 3, ylim = c(0,0.4), xlab = "", ylab = "Pr(R=r)", las = 1)
text(2.3, 0.38, bquote(pi == .(paij)), cex = 0.7)
text(5, 0.38, bquote(xi == .(ksi)), cex = 0.7)
text(7.8, 0.38, bquote(phi == .(phij)), cex = 0.7)
text(7, 0.285, bquote(sigma^2 == .(var)), cex = 0.7)
text(3, 0.28, bquote(mu == .(exp)), cex = 0.7)
}
par(mar = c(5,4,4,2)+0.1)
### reset standard margins
par(mfrow = c(1,1))
### reset plot screen
```

Figure 4.1: Probability distributions of CUBE models.
In the following subsections we discuss the usage of package CUB by means of two real data sets, univer and relgoods, bundled within the package. Detailed descriptions, related sample experiment and further references can be found at [http://www.labstat.it/home/research/resources/cub-data-sets-2/](http://www.labstat.it/home/research/resources/cub-data-sets-2/).

### 4.1. CUB models

Data set **univer** arises from a sample survey on students’ evaluation of the Orientation services that has been administered across all the Faculties of University of Naples Federico II in five waves. Participants were asked to express their ratings on a 7 point Kikert-type scale (1 = “very unsatisfied”, 7 = “extremely satisfied”) on the following items:

- **informat**: Level of satisfaction about the acquired information
- **willingn**: Level of satisfaction about the willingness of the staff
- **officeho**: Level of satisfaction about the opening hours
- **competen**: Level of satisfaction about the competence of the staff
- **global**: Level of global satisfaction

Data set collected in 2002 consists of 2179 observations: the first 7 columns correspond to subjects covariates (for instance, the dichotomous variable **gender**, equal to 0 for men and to 1 for women, and the continuous measurement **age**).

As a first step, we show how to visualize simultaneously the ordinal variables included in the data set **univer** by means of the function `multicub()`: it fits a CUB model to every ordinal variable. Then each model is represented as a point in the parameter space corresponding to the obtained ML uncertainty $1 - \hat{\pi}$ and feeling $1 - \hat{\xi}$ estimates: indeed, estimable parameters are $\pi$ and $\xi$, but interpretation depends on the orientation of the measurement scale. If the evaluation is positive in the direction of the scale, then the actual measure of feeling is $1 - \xi$; flag arguments `csiplot` and `paiplot` may be switched to `TRUE` if direct parameter visualization is preferred. Notice that `multicub()` allows for a comparative visual analysis of vectors of rating data, possibly with different lengths and also over different numbers of categories: for this reason, they are required to be stored in a list when calling `multicub()` and optionally a vector `mvett` can be set as input argument to specify the corresponding scale lengths. If only one ordinal variable is considered, then the same visual analysis is implemented via `cubvisual()`.

```r
data(univer)
listord<-univer[,8:12]  # only ratings, excluding covariates
labels<-names(univer)[8:12]
multicub(listord, labels = labels,
        caption = "CUB models on Univer data set", pch = 19,
        pos = c(1,rep(3, ncol(listord)-1)), ylim=c(0.75,1), xlim=c(0,0.4))
```

From Figure 4.2, we may conclude that the highest feeling has been expressed for the willingness of the staff, the lowest for the scheduled office hours. However, since the latter item is
affected by the highest uncertainty, it deserves further specification; thus, hereafter we focus on the item officeho.

```r
## CUB model without covariates for "officeho"
cub_00<-GEM(Formula(officeho~0|0|0), family="cub",data=univer)
summary(cub_00,digits=5)
```

---

Figure 4.2: CUB models without covariates for the items of univer \((m = 7)\).
The estimated CUB model is identified with uncertainty and feeling given by:

```
param<-coef(cub_00,digits=3)
param

pai 0.680
csi 0.197

uncertainty<-1-param[1]
uncertainty
[1] 0.32

feeling<-1-param[2]
feeling
[1] 0.803
```

The following call:

```
## CUB model without covariates
makeplot(cub_00)
```

provides a plot showing both the observed relative frequencies and the fitted probabilities, as in Figure 4.3 (left panel).

Accurate initial values for parameters allow to reach convergence to the ML solutions in acceptable time (Iannario 2012c): in this regard, the EM algorithm for CUB models without covariates is initialized within `GEM(Formula,family="cub")` by means of `inibest(m,freq),`
The R package CUB

a routine that returns preliminary parameter estimates of a CUB model for a given frequency distribution:

```r
data(univer)
freq<-tabulate(univer$officeho,nbins = 7)
ini<-inibest(m,freq)  # preliminary estimates for c(pai,csi)
ini

pai 0.9415651
csi 0.2777778
```

Features of `inibest(m,freq)` can be exploited to obtain preliminary parameter estimates for any purpose. Similarly, `inigrid()` runs a grid search across the parameter space and it returns the parameter vectors corresponding to the maximum of log-likelihood on the given grid.

In order to improve the fit and interpretation of data, we introduce covariates in the model. Specifically, to explain the feeling component we consider the dichotomous covariate `freqserv`, indicating the usage frequency of the service with levels 0 and 1 for non-regular and regular users.

```r
cub_csi<-GEM(Formula(officeho~0|freqserv|0), family="cub",data=univer)
summary(cub_csi,digits=3)
```

---

```r
===>>>> CUB model <<<===== ML-estimates via E-M algorithm

m= 7  Sample size: n= 2179  Iterations= 24  Maxiter= 500

Uncertainty
Estimates StdErr  Wald
pai 0.687 0.0192 35.9

Feeling
Estimates StdErr  Wald
constant -1.152 0.0403 28.58
freqserv -0.811 0.0850 9.55

Log-lik = -3704.357
Mean Log-likelihood= -1.7

AIC = 7414.714
BIC = 7431.773
ICOMP = 7410.96

Elapsed time= 0.5 seconds ===>>>> Thu Feb 08 19:13:19 2018
```
For the previous example, the regression coefficients for the logistic link (all of which are significant) can be retrieved by:

```r
gama <- coef(cub_csi)[2:3]
gama
```

```
[1] -1.151942 -0.811228
```

```r
gama0 <- gama[1]  ## intercept term
gama1 <- gama[2]
```

Then, to compute the feeling parameter corresponding to non-regular and regular users according to (2), respectively, it is possible to run:

```r
csi_nru <- logis(0, gama)  ## csi parameter for non regular user (freqserv=0)
csi_nru
```

```
## [1] 0.2401346
```

```r
csi_ru <- logis(1, gama)  ## csi parameter for regular user (freqserv=1)
csi_ru
```

```
## [1] 0.1231244
```

where the routine `logis()` returns the logistic transform componentwise of a standard matrix (binding the array $Y$ of covariates with a vector of ones to include an intercept term, placed in the first column) multiplied with the parameter vector `param`. In conclusion, the feeling component is specified via:

$$
\text{logit}(1 - \xi_i) = -\gamma_0 - \gamma_1 \text{freqserv}_i, \quad i = 1, \ldots, n.
$$

In addition to a likelihood ratio test obtained by comparing the log-likelihoods of the two models, we remark that the reduction in BIC index from $\text{BIC(cub}_00) = 7535.208$ to $\text{BIC(cub}_csi) = 7431.773$ strongly supports the inclusion of the covariate `freqserv` in the model. For fitting purposes it is also possible to compute the $X^2$ statistic of Pearson by implementing:

```r
pai <- coef(cub_csi)[1]
gama <- coef(cub_csi)[2:3]
data(univer)
pearson <- chi2cub(m=7, univer$officeho, W = univer$freqserv, pai, gama)
```

```
Degrees of freedom => df = 9
Pearson Fitting measure => X^2 = 97.49491 (p-val.= 0 )
Deviance => Dev = 101.0313 (p-val.= 0 )
```

```
str(pearson)
```
The R package CUB

List of 3
$ chi2: num 97.5
$ df : num 9
$ dev : num 101

Furthermore, the call to `makeplot(cub_csi)` produces a plot comparing the two fitted probability distributions conditional on the values of the dichotomous covariate, as shown by Figure 4.3 (right panel).

Note that the (normalized) dissimilarity index between observed and fitted probability distribution for a given fitted model without covariate is reported in the main title of the corresponding plot (see Section 4.5 for more details).

When covariates are specified for the feeling component, optimal preliminary estimates of the corresponding coefficients (Iannario 2008) are implemented via `inibestgama(m, ordinal, W)`, which is the option set to initialize the EM algorithm when calling `GEM(Formula~0|W|0,family="cub")`:

```r
data(univer)
inicsicov<-inibestgama(m,univer$officeho,W=univer$freqserv)
inicsicov

gamma_0 = -0.001596273
gamma_1 = -0.247482482
```

Figure 4.3: CUB without and with dichotomous covariate `freqserv` for feeling on `officeho`. 
We conclude this subsection by briefly presenting a case study in which a continuous covariate and a dichotomous one are jointly considered: in particular, we include the deviation from the mean of the logarithmic transform of \textit{Age}, that is the covariate \textit{lage}, and the covariate \textit{gender} to explain both feeling and uncertainty.

```r
# Deviation from mean of logged Age
lage <- log(age) - mean(log(age))
# Cub-Pais-Csiszar model
library(parallel)
cub_pai_csi <- GEM(Formula(officeho ~ lage + gender | lage + freqserv | 0), family = "cub", data = univer)
summary(cub_pai_csi, correlation = TRUE, digits = 3)
```

<table>
<thead>
<tr>
<th>Uncertainty</th>
<th>Estimates StdErr Wald</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>0.563 0.118 4.78</td>
</tr>
<tr>
<td>lage</td>
<td>1.240 0.613 2.02</td>
</tr>
<tr>
<td>gender</td>
<td>0.495 0.169 2.94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feeling</th>
<th>Estimates StdErr Wald</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>-1.147 0.0404 -28.41</td>
</tr>
<tr>
<td>lage</td>
<td>-0.590 0.2410 -2.45</td>
</tr>
<tr>
<td>freqserv</td>
<td>-0.824 0.0851 -9.68</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters Correlation matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant lage gender constant lage freqserv</td>
</tr>
<tr>
<td>constant</td>
</tr>
<tr>
<td>lage</td>
</tr>
<tr>
<td>gender</td>
</tr>
<tr>
<td>constant</td>
</tr>
<tr>
<td>lage</td>
</tr>
<tr>
<td>freqserv</td>
</tr>
</tbody>
</table>

Log-lik = -3693.888
Mean Log-likelihood= -1.695

AIC = 7399.776
BIC = 7433.895
ICOMP = 7396.68

Elapsed time= 1.49 seconds =====>>> Thu Feb 08 19:13:21 2018
When covariates are included for both parameters as in the previous example, the estimates will be the corresponding coefficients of the logistic regression for the uncertainty and the feeling components:

```r
cbind(ageseq, 0), param[1:3])
csicov0<-logis(cbind(ageseq, 0), param[4:6])
csicov1<-logis(cbind(ageseq, 1), param[4:6])
```

Then, the resulting `cub` model can be summarized by:

\[
\begin{align*}
\text{logit}(1 - \pi_i) &= -\beta_0 - \beta_1 \text{lage}_i - \beta_2 \text{gender}_i \\
\text{logit}(1 - \xi_i) &= -\gamma_0 - \gamma_1 \text{lage}_i - \gamma_2 \text{freqserv}_i.
\end{align*}
\]

Since no plot is directly provided as output, the performance of the `cub` model with significant covariates on feeling and uncertainty parameters may be summarized as in Figure 4.4, obtained by the following code:
Figure 4.4: CUB models with covariates: logged *age* and *gender* for uncertainty and logged *age* and *freqserv* for feeling.

Summarizing, the sketch of analysis so pursued indicates that satisfaction increase with age whereas indecision decreases, and that men are more satisfied than women across all profiles.

### 4.2. Comparing IHG and CUBE models

As shown in Figure 4.1, CUBE models offer a wide flexibility in fitting data having different shapes and features. We now discuss how to perform an analysis on ordinal data with a CUBE model by considering the variable *willingn* within *data(univer)*.

```r
starting<-c(0.5, 0.5, 0.1)
cubefit<-GEM(Formula(willingn~0|0|0),family="cube", starting = starting, maxiter = 100, toler = 1e-4, data=univer)
summary(cubefit,digits=7)
```

=======================================================================
As shown with the current example, the user can set some extra options for the estimation procedure: the maximum number of iterations allowed for the optimization procedure (the default value for CUBE model is maxiter = 1000), the error tolerance to stop iterations (by changing the default option toler = 1e-6), and most importantly the argument starting, that is the vector of starting values to initialize the EM algorithm. The default option is based on the routine inibestcube(), a convenient choice of preliminary estimates for parameters in CUBE models. For CUBE models with covariates, only for feeling or for all the three parameters, initial estimates of parameters are implemented via inibestcubecsi() and inibestcubecov(), which are also set as default options when calling

```r
GEM(Formula(ordinal~0|W|0),family="cube")
```

and

```r
GEM(Formula(ordinal~Y|W|Z),family="cube")
```

respectively.
Given the shape of the distribution of the selected ordinal data (see Figure 4.5, left panel), the low level of uncertainty \((1 - \text{coef}(\text{cubefit})[1] = 0.1174)\) and in particular, the extreme feeling \((1 - \text{coef}(\text{cubefit})[2] = 0.8752)\) corresponding to the modal value at the last category \(m = 7\), we check if the more parsimonious IHG model provides comparable goodness of fit. For such a model, the ML estimation procedure is initialized by considering as starting value the moment estimate of the preference parameter \(\theta\), computed via the function \text{iniihg}(m, \text{freq}).

```r
ihgfit<-GEM(\text{Formula}(\text{willingn} \sim 0), \text{family}="\text{ihg}", \text{data}=\text{univer})
summary(ihgfit, \text{digits}=7)
```

<table>
<thead>
<tr>
<th>Estimates</th>
<th>StdErr</th>
<th>Wald</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta)</td>
<td>0.04756359</td>
<td>0.001414214</td>
</tr>
</tbody>
</table>

Log-lik = -3244.08
Mean Log-likelihood = -1.488793
Log-lik(UNIFORM) = -4240.138
Log-lik(saturated) = -2988.635
Deviance = 510.8901

AIC = 6490.16
BIC = 6495.847
ICOMP = 6488.16

Elapsed time= 0 seconds =====>>> Thu Feb 08 19:13:22 2018

In order to check if the fit improvement obtained with CUBE models justifies the inclusion of two additional parameters, that is, if the more parsimonious IHG model should be discarded in favor of CUBE model, we perform a likelihood ratio test:

```r
llcube<-logLik(cubefit)
llihg<-logLik(ihgfit)
lrt<- -2*(llihg - llcube) ### 495.9135
pv<- 1-pchisq(lrt, 2) ### 0
```

It confirms that a consistently better fit is ensured by the CUBE model. Figure 4.5 displays the observed and fitted probability distributions for both models as returned by \text{makeplot}(\text{cubefit}) and \text{makeplot}(\text{ihgfit}), respectively.

### 4.3. Models for shelter effect

As already mentioned in Section 2, the class of CUB mixture models includes a specific extension to fit the so-called shelter effect, arising in presence of an inflated category. We show how
to perform the analysis of a possible shelter effect on the data set `data(relgoods)`, a survey designed to assess importance of relational goods and involvement in leisure time activities: data were collected in December 2014 in the metropolitan area of Naples, Italy. Every participant was asked to measure his/her personal score for relational goods (for instance, time dedicated to friends and family) and his/her preferences towards leisure activities on a 10 point ordinal scale, ranging from 1 = “Not at all, nothing, never”, to 10 = “Totally, extremely important, always”. On the whole, respondents gave rating scores to 34 items. The sample is composed by 2459 interviews and 16 subjects’ covariates (including gender, education, residence, marital status and age, among others). For these data some missing values are present.

Here, the implementation of the so-called shelter effect within CUB models is discussed. Let us consider the ordinal variable `Writing`, indicating the respondents’ degree of engagement in writing as a preferred activity for leisure time.

One observes an excess in frequency corresponding to the first category, so we first fit an extended CUB model with shelter effect by running the following code (notice that both the available parameterizations (3) and (4) are reported on screen):

```r
cub_she<-GEM(Formula(Writing~0|0|0), family="cub", shelter = 1, maxiter=500,toler=1e-3,data=relgoods)
summary(cub_she)
```

---

![Figure 4.5: CUBE (left panel) and IHG (right panel) fit to the data.](image)

---

```r
cub_she<-GEM(Formula(Writing~0|0|0), family="cub", shelter = 1, maxiter=500,toler=1e-3,data=relgoods)
summary(cub_she)
```

---

```
=======================================================================
=====>>> CUB model <<<===== ML-estimates via E-M algorithm
=======================================================================

```

---

```
m= 10 Sample size: n= 2449 Iterations= 77 Maxiter= 500
```

---

```
Estimates    StdErr    Wald
pai1 0.1922306 0.01652574 11.63219
pai2 0.4902216 0.01907354 25.70166
csi 0.7518936 0.01433876 52.43784
```
As shown by the printed output, the observed distribution is affected by a low weight for the shifted Binomial component \( \text{pai1} \leftarrow \text{coef}(\text{cub_she})[1] \), and a moderately high weight for the Uniform component \( \text{pai2} \leftarrow \text{coef}(\text{cub_she})[2] \): Figure 4.6 (left panel) shows the plot returned by running \text{makeplot}(\text{cub_she}) \), which compares the observed frequencies and the fitted probabilities.

As a matter of fact the data set under consideration includes several instances of distributions characterized by a huge heterogeneity and an inflated category. As already explained in Section 2, such occurrence motivated the specification of CUSH models. For the variable Writing here considered, the fit of a CUSH model is implemented by the code:

\[
\text{cush} \leftarrow \text{GEM(} \text{Formula(Writing}\sim0)\text{,family="cush",shelter = 1, data=relgoods)}
\]

\[
\text{summary(cush,digits=3)}
\]
Summarizing, CUB model with shelter effect provides a better fit in terms of log-likelihood and BIC indexes. As for other models within the GEM family, `makeplot(cush)` compares fitted and observed frequency distributions as displayed in Figure 4.6 (right panel).

Figure 4.6: CUB with shelter (left panel) and CUSH (right panel) fit to the data.

The possibility of including covariates to explain the shelter effect in a CUSH model is presented in the next section along with the simulation routines offered by package CUB.

### 4.4. Simulation routines and experiments

Package CUB offers several facilities to perform simulation experiments involving GEM models. In order to generate \( n \) pseudo-random numbers from CUB, CUBE, CUB with shelter, CUSH and IHG distributions, respectively, the following functions are available:

```r
simCUB <- simcub(n, m, pai, csi)
simCUBE <- simcube(n, m, pai, csi, phi)
simCUBshe <- simcubshe(n, m, pai, csi, delta, shelter)
simCUSH <- simcush(n, m, delta, shelter)
simIHG <- simihg(n, m, theta)
```

For instance, consider the following example, where the theoretical and fitted probability distributions of a CUB model are displayed for simulated data in Figure 4.7:

```r
m<-9; n<-500
pai<-0.7; csi<-0.2
```
pr<-probcub00(m, pai, csi)
set.seed(123)
ordinal<-simcub(n, m, pai, csi)
cub<-GEM(Formula(ordinal~0|0|0),family="cub", maxiter = 50, toler = 1e-4)
pr_est<-fitted(cub)[,1]
plot(1:m, pr, type = "h", xlab = "Ordinal categories", ylab = "Probability", lwd = 3, ylim = c(0, 0.3))
vett<-1:m + 0.2
lines(vett, pr_est, type = "h", col = "blue", lwd = 3, lty = 2)
legend(1, 0.3, legend = c("Theoretical", "Fitted"), col = c("black", "blue"), lty = c(1, 2), lwd = 3, text.col = c("black", "blue"), bty = "n")

pai 0.66719
csi 0.18732

![Graph showing fitted and theoretical CUB distributions on simulated data.](image)

**Figure 4.7:** Fitted and theoretical CUB distributions on simulated data.

In order to simulate observations from a model with covariates, a two-step approach has to be implemented: one should first obtain the parameters ($\pi_i, \xi_i, \ldots$) corresponding to the chosen respondents’ profiles and then generate the sample data by using corresponding simulation routines with the given parameters. We outline how to implement such a procedure for a CUSH model with a dichotomous covariate for the *shelter effect*. Figure 4.8 displays the two CUSH distributions conditional to the value of the covariate.
\begin{verbatim}
omega0<- -1.5
omega1<- -2
delta0<-as.numeric(logis(0, c(omega0, omega1))) ## 0.1824255
delta1<-as.numeric(logis(1, c(omega0, omega1))) ## 0.0293122
m<-9
n0<-700
n1<-1300
set.seed(1234)
ord0<-simcush(n0, m, delta0, shelter = s)
ord1<-simcush(n1, m, delta1, shelter = s)
ordinal<-c(ord0, ord1)
X<-c(rep(0, n0), rep(1, n1))
cushcov<-GEM(Formula(ordinal~X), family="cush", shelter = m)
coef(cushcov)
makeplot(cushcov)
\end{verbatim}

constant -1.492551
X -1.724813

![CUSH distributions, given delta-covariate=0, 1](image)

**Figure 4.8:** CUSH distributions with a dichotomous covariate.

4.5. Extra features of CUB package
The R package **CUB** provides a flexible framework for the analysis of ordinal data that covers some of the most common distributions related to GEM models. For a given discrete probability distribution, some additional features concern, for instance, the functions `gini()` and `laakso()`, for the normalized Gini (1912) and Laakso and Taagepera (1989) heterogeneity indexes, respectively, and the function `deltaprob(prob)` for the Mean Difference index (according to the de Finetti and Paciello (1930) formulation): Figure 4.9 shows an example of usage.

```r
m<-7
pai<-0.4
csi<-0.2
prob<-probcub00(m, pai, csi)
Giniindex<-round(gini(prob), digits = 3)
Laaksoindex<-round(laakso(prob), digits = 3)
Delta<-round(deltaprob(prob), digits = 3)
plot(1:m, prob, type = "n", ylab = "", xlab = "", ylim = c(0, 0.4), las = 1)
lines(1:m, prob, type = "h", lwd = 3)
legend("topleft", xjust = 1, legend = c(paste("Gini =", Giniindex),
paste("Laakso =", Laaksoindex),
paste("Delta =", Delta)), cex = 0.8)
```

![Figure 4.9: Heterogeneity and mutual variability indexes.](image)

Another example concerns the simulation of the critical values for the normalized dissimilarity index between observed relative frequencies \((f_1, \ldots, f_m)\) and estimated (theoretical)
probabilities \( (p_1(\hat{\theta}), \ldots, p_m(\hat{\theta})) \):

\[
Diss = \frac{1}{2} \sum_{r=1}^{m} |f_r - p_r(\hat{\theta})|,
\]

which is computed by the routine \texttt{dissim}(). This index evaluates the distance of the estimated model to a perfect fit. Due to its importance in model performance, this dissimilarity measure is computed when fitting CUB models and extensions within \texttt{CUB} package and displayed also in the plot title when it is returned. For this index we show how to run a simulation experiment for detecting the critical value for an assigned CUB model. The code concerns the distribution of a CUB model with \( \pi = 0.3 \) and \( \xi = 0.8 \). Figure 4.10 displays the results after \texttt{nsimul} = 10000 simulations of samples of sizes \( n = 300 \) over \( m = 7 \) categories.

```r
pai<-0.3; csi<-0.8
nsimul<-10000
n<-300; m<-7
vectdiss<-rep(NA, nsimul)
for (j in 1:nsimul){
  ordinal<-simcub(n, m, pai, csi)
  mod<-GEM(Formula(ordinal~0|0|0),family="cub")
  theor<-fitted(mod)[,1]
  freq<-tabulate(ordinal, nbins = m)/n
  vectdiss[j]<-dissim(freq, theor)
}

sortvect<-sort(vectdiss)
alpha<-0.05
signif<-sortvect[(1-alpha)*nsimul]  # empirical percentile 0.05
cr<-vectdiss[vectdiss>signif]       # critical region

effe<-density(vectdiss)
band<-effe$bw  # band width of the kernel plot
f<-function(x){  # compute kernel density (the default is Gaussian kernel)
  (1/(band*length(vectdiss)))*sum(dnorm((x-vectdiss)/band))
}

sortcr<-sort(cr)
sup<-numeric(length(sortcr))
for (j in 1:length(sortcr)){
  sup[j]=f(sortcr[j])  # compute kernel density values for critical values
}
title <- paste("Normalized Dissimilarity distribution, Critical level(0.05) = ",
                round(signif, 3))
plot(density(vectdiss), main = title, cex.main = 0.7, lwd = 3, xlab = "",
     cex.main = 0.7, las = 1)
polygon(c(signif, sortcr, sortcr[length(sortcr)], signif), c(0, sup, 0, 0),
        col = "gray")
abline(h=0)
```
Figure 4.10: Distribution of the (normalized) dissimilarity index for simulated CUB model with $\pi = 0.3$ and $\xi = 0.8$.

We conclude the presentation of package CUB by underlying the effective graphical interpretation of parameters allowed by CUB models. When covariates are included to explain feeling and uncertainty, a Scatter of Parameter Estimates (SPE, for short) (Iannario and Piccolo 2014) reveals to be a convincing plotting device to identify different response patterns across subsets of respondents. A SPE simply performs the multiple representation of a CUB model as a point in the parameter space for each parameter vector $(\pi_i, \xi_i)$ associated with each respondent. We show an example based on the univer dataset, by showing the CUB model for the global satisfaction conditioning both feeling and uncertainty on the evaluation expressed for officeho, and further specifying both components for varying age.

```r
data(univer)
ordinal<-univer$global
lage<-log(univer$age)-mean(log(univer$age))  #Deviation from mean of logged Age
Y<-W<-cbind(univer$officeho,lage)
cub_pai_csi<-GEM(Formula(global~Y|W|0),family="cub",data=univer)
bet<-coef(cub_pai_csi)[1:3]
```
The R package CUB

```r
gama <- coef(cub_pai_csi)[4:6]
paivett <- logis(Y, bet)
csivett <- logis(W, gama)
n <- length(ordinal)
main <- "Scatter plot of estimated parameters"
vettcol <- symb <- rep(NA, n)
vettcol[univer$officeho <= 3] <- "red"
vettcol[univer$officeho == 4] <- "black"
vettcol[univer$officeho >= 5] <- "blue"
symb[univer$officeho <= 3] <- 0
symb[univer$officeho == 4] <- 1
symb[univer$officeho >= 5] <- 2
plot(1 - paivett, 1 - csivett, xlim = c(0, 1), ylim = c(0, 1),
     cex = 0.8, pch = symb, col = vettcol,
     xlab = expression(1 - pi), ylab = expression(1 - xi),
     main = main, cex.main = 1, font.main = 4)
legend("topright", legend = c("Unsatisfied", "Indifferent", "Satisfied"),
       cex = 0.7, text.col = c("red", "black", "blue"), pch = c(0, 1, 2),
       col = c("red", "black", "blue"))
```

5. Conclusions

Package CUB is under active development. Future plans include extra functions to fit CUBE models with further covariate specification, as well as higher flexibility for GeCUB models to possibly include covariates only for shelter effect or any pair of components. Implementation of Hierarchical CUB models and other extensions, including an adjusted version of CUB models to account for response styles and a multivariate model for repeated measurements are under scrutiny.

Package CUB has been implemented under R Version 3.2.5.

References


Scatter plot of estimated parameters


**Affiliation:**

Maria Iannario, Domenico Piccolo, Rosaria Simone
Department of Political Sciences
University of Naples Federico II
80138 Naples, Italy
Telephone: +39/081-2537465
Fax: +39/081-2537466
E-mail: maria.iannario@unina.it, domenico.piccolo@unina.it, rosaria.simone@unina.it
URL: http://www.docenti.unina.it/MARIA.IANNARIO
URL: http://www.docenti.unina.it/DOMENICO.PICCOLO
URL: https://www.docenti.unina.it/ROSARIA.SIMONE