A short guide to CUB 3.0 Program

ARTICLE · JANUARY 2013

A Short Guide to CUB 3.0 Program

Maria Iannario and Domenico Piccolo
Department of Political Sciences, University of Naples Federico II, Italy.
Email: *maria.iannario@unina.it; domenico.piccolo@unina.it*

Version released on `September 2013`

Abstract

The purpose of this short guide is to introduce users to the version 3.0 of a program, written in the R statistical environment, to make effective applications of CUB models by exploiting their many capabilities both from computational and graphical point of views. The basic commands are presented with some examples. Generalizations and extensions of the standard CUB models are also mentioned. For more extensive study readers are suggested to look for updated references.

Key Words: Ordinal data, CUB models, CUBE models,

1 Introduction

In several applied researches, data are collected as categorical ordinal observations. Sometimes they are actually ordered (as in judgements, preferences, degree of adhesion to a sentence, etc.) whereas, in other circumstances, they are categorized for convenience (age of people in classes, measures of objects in block of constant size, blood pressure for classifying heart health status, etc.). It is possible to consider also ranks as ordinal data if we limit ourselves to interpret the ranks of a single object as an ordered evaluation. Caution is necessary in interpreting ranks of related objects since evaluations are not independent.

The program[1] we are going to introduce is a statistical software able to specify, estimate and testing a large class of statistical models, defined as CUB since in the standard options they are a convex *C*ombination of discrete *U*niform and shifted *B*inomial random variables. The software is organized on the basis of some main functions (able to perform the general purpose of building statistical models) which call for several other functions (in charge of limited and specific objectives). The program is presented as a large script; thus, users can apply also a subset of the available functions, if necessary and/or convenient.

We report the notation and the wording. The parameters involved in the models, described in text and comments, are denoted as: `pai` (π), `csi` (ξ), `phi` (ϕ), `delta` (δ), `bet` (β), `gama` (γ), `omega` (ω). The variance-covariance matrix of estimates is denoted as `varmat`. In addition, *only* the two global main functions of the program (that is, `CUB` and `CUBE`) are in upper cases.

[1] The version 3.0 of the CUB program is freely available from the Authors upon request.

Notice that all programs involving the building of CUB models and their extensions return a number of values of interest, as specified in the code with `assign(.)`. More specifically, after running a main function for estimation and testing a CUB (or of a CUBE) model for the observed `ordinal` (=ratings or rankings), the values `pai`, `csi` (or the values `pai`, `csi`, `phi`) and the matrix `varmat` of the estimated models are available in the computer memory. In this way, the main information about estimated models may be saved and maintained for further elaborations.

2 Data input

Hereafter, for convenience, we limit the discussion to sample data available as n ratings $\boldsymbol{r} = (r_1, r_2, \ldots, r_n)'$, where $r_i \in \{1, 2, \ldots, m\}$ for a given $m \geq 3$. Notice that, *to estimate any model*, the constant m must *always* be specified as a global variable. Indeed, m cannot be defined -in any circumstance- as the maximum integer of the vector `ordinal` since this criterion would fail if no respondents choose the highest value of the support.

We assume that ordinal data $\boldsymbol{r}$ are realizations of a random variable R and are available as a vector `ordinal` in the environment R. The same is true for possible covariates which we introduce to explain responses and improve the fitting.

If ordinal data are available in a matrix or data frame, the following code is convenient.

```
> dati=read.table("C:/.../...",header=T)
> ordinal=dati[,j]              ### if ordinal data are in the j-th column
```

Sometimes, ratings are available as aggregated frequencies $(n_1, n_2, ..., n_m)'$ in a vector `frequencies`. Then, we have to expand them to generate a vector of length $n = n_1 + n_2 + ... + n_m$ and then run the `CUB` command.

```
> ordinal=rep(1:m,frequencies)
> CUB(ordinal)
```

Notice that `frequencies` must be a vector of length m even if some observed frequencies are 0.

The current version of the program does not allow missing values in the input; several softwares with different approaches are available in the literature to impute missing values for ordinal data. CUB models are an interesting alternative for the imputation of missing data by substituting the modal value of the estimated CUB model.

3 Building a CUB model

Assume you are in the directory where the main program is resident and that you loaded the vector of ratings `ordinal` you are going to analyze. Let m =`number-of-ordinal-categories`.

To activate the program in the R environment and build a CUB model (without covariates) for `ordinal`, it is sufficient to run the following commands.

```
> source("CUB.R")
> m=number-of-ordinal-categories
> CUB(ordinal)
```

If a *shelter effect* is present at the category $R =$ `ccc`, then the code for estimating a CUB model with such a *shelter effect* is the following.

```
> source("CUB.R")
> m=number_of_ordinal_categories
> CUB(ordinal,shelter=ccc)
```

In both cases, the output is a long list of inferential results on the parameters and on the fitting of the model. In addition, the default version of the program plots the observed (relative) frequency and the (estimated) CUB probability distributions. To omit the plot, modify the main command as follows:

```
> CUB(ordinal, makeplot=FALSE)
```

When matrices `Y` and `W` contains the columns of subjects' covariates for *uncertainty* (parameters $1 - \pi_i$) and *feeling* (parameters $1 - \xi_i$), respectively, the commands to build a CUB model with covariates are the following.

```
> CUB(ordinal,Y=paicov)           ### if paicov is a covariate for uncertainty

> CUB(ordinal,W=csicov)           ### if csicov is a covariate for feeling

> CUB(ordinal,Y=paicov, W=csicov) ### if paicov and csicov are covariates
                                  ### for uncertainty and feeling, respectively
```

If covariates are obtained by manipulations or derive from different data set it is possible to bind them (if all vectors have the same length as `ordinal`) as in the following examples:

```
> x1=dat1[,1]; x2=log(dat2[,3]); x3=1:n        ### definitions of covariates
> CUB(ordinal,Y=cbind(x1,x2),W=cbind(x3,x1*x2) ### CUB model with covariates
                                               ### for uncertainty and feeling
```

To visualize an estimated CUB model in the parameter space, with an asymptotic confidence ellipse around the estimates, the `library(ellipse)` should be loaded. Then, to draw a 95% confidence ellipse for the parameter (π, ξ), we require the variance-covariance matrix of estimates. Thus, the code is the following.

```
> source("CUB.R")
> m=number_of_ordinal_categories
> CUB(ordinal)
> plot(1-pai,1-csi,main="Estimated CUB model for ordinal",
       xlim=c(0,1),ylim=c(0,1),
       xlab=expression(paste("Uncertainty  ", (1-pi))),
       ylab=expression(paste("Feeling  ", (1-xi))))
> library(ellipse)
> lines(ellipse(varmat,centre=c(1-pai,1-csi)), lwd=2)
```

In presence of covariates, the output does not include plots, except when a single dichotomous covariate `dum` (strictly defined with values $0, 1$) is introduced to explain *uncertainty* or *feeling*. In these circumstances, an automatic plot is produced to show the estimated probability distributions conditioned by `dum=0` (circled) and `dum=1` (dotted), respectively. Typical commands are the following.

```
> dum=ifelse(Gender=="male",0,1)  ### a dichotomous covariate for Gender

> CUB(ordinal,Y=dum)              ### CUB model with Gender for uncertainty

> CUB(ordinal,W=dum)              ### CUB model with Gender for feeling


> CUB(ordinal,Y=dum,W=dum)        ### CUB model if Gender is the "same"
                                  ### covariate for uncertainty and feeling
```

Some preliminary analysis is necessary to build models where both subjects and objects' covariates are present. Assume that ratings are in the vectors `item1, item2, ...` and are vectorized into `ITEM`. In addition, subjects' covariates and objects' covariates for *uncertainty* and *feeling* are expanded into `Ytilde`, `Wtilde`, `Xtilde`, respectively. Then, the following command:

```
> CUB(ITEM,Y=cbind(Ytilde,Xtilde),W=cbind(Wtilde,Xtilde))
```

will generate ML estimates of $\boldsymbol{\theta}$ parameters and related statistics.

4 CUBE models

More options are offered to build CUBE models, as shown in the complete version of the command.

```
> CUBE(ordinal)                         ### essential command for CUBE models

> CUBE(ordinal,starting=rep(0.1,3),maxiter=500,toler=1e-6,
       makeplot=TRUE,expinform=FALSE)
                                        ### complete command for CUBE models
```

Thus, the user may choose to modify initial values `starting` (as discussed in Section 5), to set maximum number of iterations (`maxiter`=500), to define the tolerance for the convergence of log-likelihood functions (`toler`= 1e−6), to visualize the observed and fitted distributions (if `makeplot=FALSE` the plot is omitted) and to choose between the application of observed (default option) or expected information matrix (if `expinform=TRUE`) in the computation of standard errors of the ML estimates.

The same options are also available for building CUBE models with covariates $\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{Z}$ for explaining *uncertainty*, *feeling* and *overdispersion*, respectively. The corresponding commands are the following.

```
> CUBE(ordinal,Y,W,Z)  ### essential command for
                       ### CUBE models with covariates

> CUBE(ordinal,Y,W,Z,starting=rep(0.1,ncol(cbind(Y,W,Z))+3),
       maxiter=500,toler=1e-6)
                       ### complete command for
                       ### CUBE models with covariates
```

With special regard to CUBE models with covariates, it should be emphasized that accurate starting values for a CUBE model are strictly necessary, given the lengthy convergence process of EM algorithm. Thus, as an effective strategy, we suggest to implement the following steps:

- start the estimation procedure with a small random subset of the whole sample;
- re-start the estimation procedure on the whole data set and give as starting values those obtained in the subset experiment with a very high tolerance (`toler=0.1`, say);
- plug these new preliminary estimates in the command to search for the final and more efficient estimates.

In some circumstances, it may be also convenient to reiterate the proposed strategy several times to achieve better and quick results.

5 Inferential issues

A full usage of the program implies some knowledge of the selected estimation routines and of the several measures computed to test and validate the estimated model. In addition, the availability of these functions allows researchers to build further tools for their specific needs.

First of all, the Maximum Likelihood (ML) method is performed thanks to the EM procedure which requires accurate initial values to reach convergence in acceptable time. This goal has been effectively obtained by means of specific functions: `inibest`, `inigrid`, `inibestgama` and `inibestcube`.

For CUB models without covariates the default is `inibest`; for CUB models with covariates for *feeling* the program always implements `inibestgama` whereas for CUBE models (without covariates) `inibestcube` is performed. The commands to obtain such initial estimates are the following.

```
> inibest(freq)          ### automatically computed for CUB models

> inigrid(freq)          ### given as a reference for CUB models

> inibestgama(ordinal,W) ### automatically computed for CUB models
                         ### with covariates W for feeling

> inibestcube(ordinal)   ### automatically computed for CUBE models
```

Observe that the functions `inibest` and `inigrid` are applied to `freq` whereas the functions `inibestgama` and `inibestcube` use the vector `ordinal`. In all other cases, initial values are arbitrarily set at 0.1.

Second, for each parameter of CUB and CUBE models, the output of the program shows parameter estimates, asymptotic standard error, Wald-test and p-value. The variance-covariance and the correlation matrices of estimates are also presented.

Third, a list of likelihood-based measures and several (general and specific) fitting indexes are printed.

Finally, for CUB and CUBE models without covariates, the program presents a table where -for each category- observed relative frequencies, estimated probabilities, Pearson and relative residuals are listed.

6 Plotting facilities

A fundamental value of CUB models is the easiness of interpretation when estimated models are plotted as points in the parameter space, that is the unit square. A simple graphical device is `cubvisual` which visualizes a CUB model for the data vector `ordinal` as a single point in the parameter space with some useful options. If necessary, other estimated models (=points) may be added with the standard commands of the R environment (as the function `points`(.), for instance). Notice that a global value of m must be always defined. Thus, the code is the following.

```
> m=number-of-categories
> cubvisual(ordinal)                                ### minimal information
```

```
> cubvisual<-function(ordinal,labordinal=NULL,
                  maintitle="CUB models parameter space",
                  xlim=c(0,1),ylim=c(0,1))        ### complete options
```

A more general opportunity is offered by the functions `multicub` or `multicube` which allow to plot, with several options, many estimated CUB or CUBE models over the same unit square.

Assume that all ordinal data are the columns (greater than 1) of the observed matrix `matord`. Then, the minimal and complete commands are as follows.

```
> multicub(matord,m)                                   ### minimal information

> multicub(matord,m,etich=as.character(1:ncol(matord)),
         titolo="CUB models",colori="black",simboli=19,
         thickness=1.5,xwidth=c(0,1),ywidth=c(0,1))  ### complete options
```

For each point (=estimated CUB model), we may specify label (with `etich`), colour (with `colori`), symbol (with `simboli`), `thickness` point and the size of the plot (the unit square is the default).

Finally, a recent visual tool for CUB models with covariates is the *Scatter of Parameter Estimates* (= SPE) which consists in the plots of estimated $(\hat{\pi}_i, \hat{\xi}_i)$, for $i = 1, 2, \ldots, n$. This scatter plot is able to detect peculiar behaviour in subset of respondents, as we will show in section 8 with the aid of a real case study.

7 Simulation functions

If pseudo-random numbers are to be generated by CUB models and their extensions, some functions are available in the main program. We will list some of them.

```
> xxx=simcub(n,m,pai,csi)             ### generate n observations from
                                      ### a CUB model with parameters
                                      ### (pai, csi), for a given m

> yyy=simcubshe(n,m,pai,csi,delta,ccc) ### generate n observations
                                      ### from a CUB model
                                      ### with a shelter effect at ccc,
                                      ### for a given m,
                                      ### with parameters (pai, csi)

> www=simcube(n,m,pai,csi,phi)        ### generate n observations from
                                      ### a CUBE model with
                                      ### parameters (pai, csi, phi),
                                      ### for a given m
```

When performing simulation experiments, it is useful to apply routines which are not so elaborate with respect to the presentation of results. Thus, the program includes two simplified versions of the estimation and testing of CUB and CUBE codes, respectively, which have been specifically finalized to perform long run of simulation experiments.

To activate these commands, use the following codes.

```
> cub00forsim(ordinal,maxiter=500,toler=1e-6)      ### CUB model

> cubeforsim(ordinal,starting=rep(0.1,3),maxiter=500,toler=1e-6)
                                                     ### CUBE model
```

For a standard CUB model we may modify both `maxiter` (the maximum number of iterations) and `toler` (the criterion for convergence based on the increment of log-likelihood functions). In addition, for CUBE models we may also modify `starting` (the sequence of initial values for the estimates of parameters).

8 Some examples

We exemplify previous commands in some real and faked situations with a main emphasis on the graphical outputs.

First of all, given $m = 9$, we generate $n = 500$ ordinal observations from a CUB model with $\pi = 0.3$ and $\xi = 0.8$. Then, we estimate the parameters and plot them in the parameter space with a 95% confidence ellipse. The parameters correlation is also computed and shown: this is possible since `CUB` function calls for `cub00` which `assign`s `varmat` to the variance-covariance of the estimates. In Figure 1, we present the output of CUB model estimation run (upper panel) and the visualization of the estimated CUB model in the parameter space (bottom panel). The code is the following.

```
> m=9; n=500
> pai=0.7; csi=0.2
> ordinal=simcub(n,m,pai,csi)
> source("CUB.R")
> ### Division of the screen in two panels
> par(mfrow=c(2,1))
> par(mar=c(5,4,3,2)+0.1)
> ### First plot
>    CUB(ordinal)
> ### Second plot
>    plot(1-pai,1-csi,main="CUB model for ordinal",cex=1.2,cex.main=1,las=1,
               pch=19, xlim=c(0.2,0.4),ylim=c(0.7,0.9),font.lab=4,cex.lab=1,
               xlab=expression(paste("Uncertainty  ", (1-pi))),
               ylab=expression(paste("Feeling  ", (1-xi))))
```

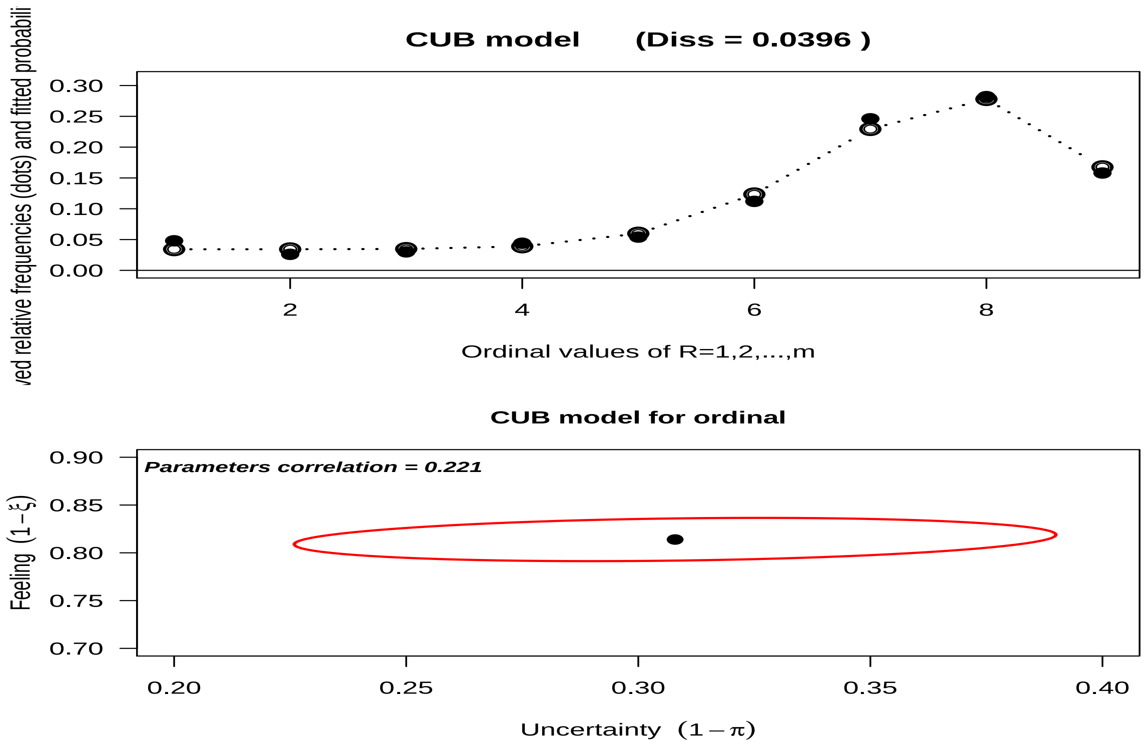

Figure 1: Observed and estimated distribution (upper) and visualization of the CUB model in the parameter space (bottom)

```
> ### Compute parameters correlations
>   corrpar=varmat[1,2]/sqrt(varmat[1,1]*varmat[2,2])
>   labelcorr=paste("Parameters correlation =",round(corrpar,3))
>   text(0.23,0.89,labels=labelcorr,font=4,cex=0.8)
> ### Draw ellipse
>    library(ellipse)
>    lines(ellipse(varmat,centre=c(1-pai,1-csi)),lwd=2,col="red")
>    plot(1-pai,1-csi,main="CUB model for ordinal",
         cex=1.2,cex.main=1, font.lab=4,cex.lab=1,
         pch=19, xlim=c(0.15,0.30),ylim=c(0.75,0.85),
         xlab=expression(paste("Uncertainty  ", (1-pi))),
         ylab=expression(paste("Feeling  ", (1-xi))))
> par(mar=c(5,4,4,2)+0.1)
> par(mfrow=c(1,1))
```

As a second example, we use the simple command `cubvisual` which plots an estimated CUB model (without covariates) for a single vector `ordinal1` as a point in the parameter space. Then, we add -on the same plot- the representation of a further CUB model estimated

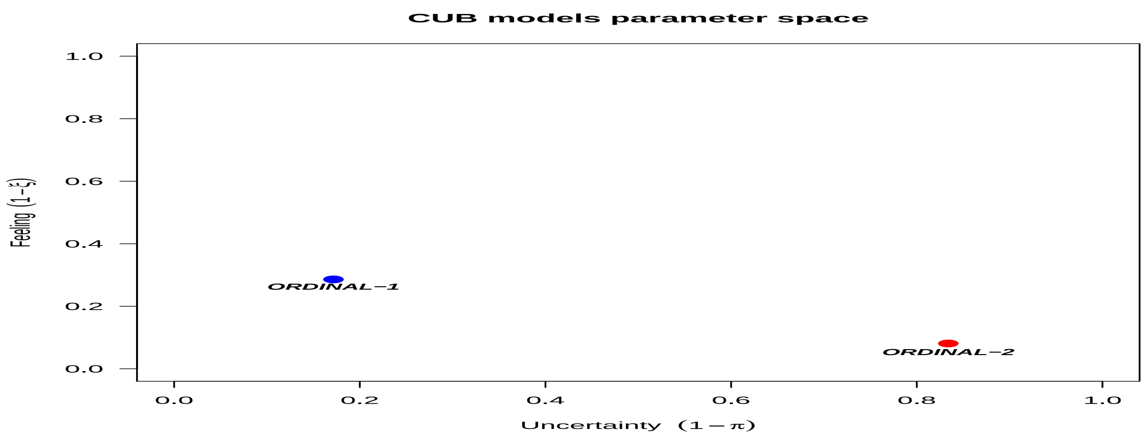

Figure 2: A visual representation of two estimated CUB models in the parameter space

on a different vector `ordinal2`. Code commands follow and the results are shown in Figure 2.

```
> m=7
> source("CUB.R")
> ### First model for ordinal1 ("blue")
> cubvisual(ordinal1,"ORDINAL-1")
> ### Second model for ordinal2 ("red")
> cub00(ordinal2,noplot=TRUE)
> points(1-pai,1-csi,pch=19,cex=1.5,col="red")
> text(1-pai,1-csi,labels="ORDINAL-2",font=4,pos=1,offset=0.5,cex=0.8)
```

As a third example, we use the `multicub` command on a real data set concerning several evaluations expressed on a Likert scale with $m = 10$ by users of a public bus transport to/from a metropolitan area. We assume that all ordinal data have been loaded in a matrix `dati`, consisting of $n = 105$ ratings on $nk = 16$ items. The code is the following.

The first command is the standard one (upper panel of Figure 3) whereas the second one is a more elaborated version of the same command (upper panel of Figure 3).

```
> dati=read.table("C:/.../Transports.R",header=TRUE)
> source("CUB.R")
> m=10
> par(mfrow=c(2,1))
> par(mar=c(5,4,3,2)+0.1)
### First plot (upper)
> multicub(dati,m)
### Second plot (bottom)
```

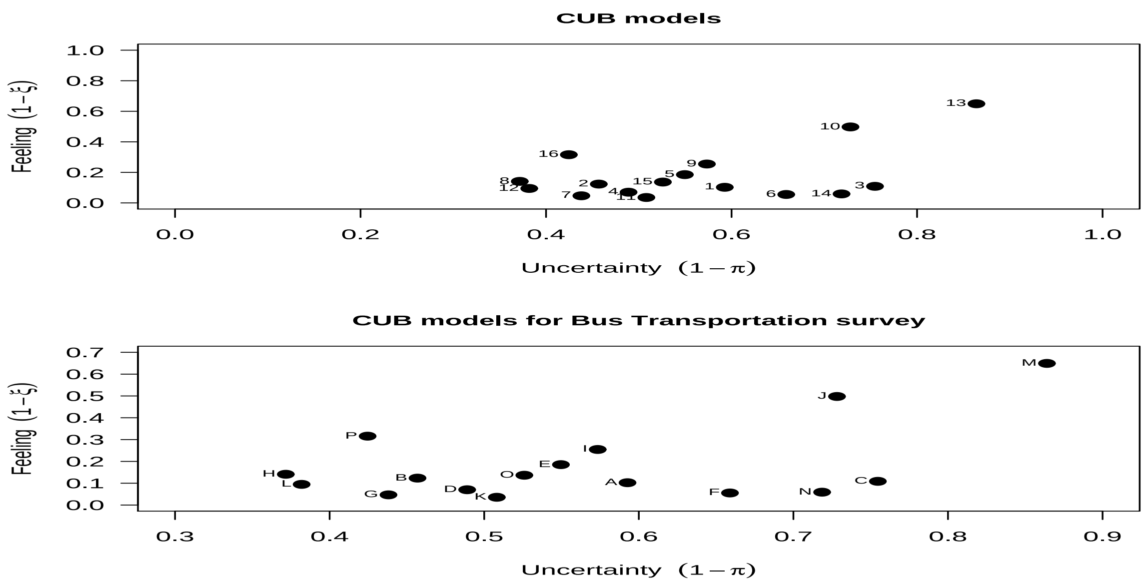

Figure 3: Usage of `multicub` command: standard version (upper) and more elaborated version (bottom)

```
> multicub(dati,m,etich=LETTERS[1:ncol(dati)],
           titolo="CUB models for Bus Transportation survey",
           colori="black",simboli=19,thickness=1.5,
           xwidth=c(0.3,0.9),ywidth=c(0,0.7))
> par(mar=c(5,4,4,2)+0.1)
> par(mfrow=c(1,1))
```

As a fourth example, we consider a large data set called `nes96` which consists in the evaluation of 944 respondents with respect to the political Left-Right orientation of Bill Clinton (=`ClinLR`). This evaluation is examined as a function of the party identification (=`PID`), the age in years (=Age) and the education level (=`Educ`) of the respondents. The ordinal variables `ClinLR` and `PID` are expressed on a Likert scale ranging from *Surely Left*= 1 up to *Surely Right*= 7. We limit ourselves to report the commands for the best CUB model obtained to explain the responses `ClinLR`.

Then, we plot the *SPE* diagram for all respondents, by introducing a new variable to re-define `PID` with a simplified recoding scheme. More specifically, we let `RePID`$= -1, 0, 1$ if `PID`$= 1, 2$, `PID`$= 3, 4, 5$ `PID`$= 6, 7$, respectively. In this way, we consider `RePID` a rough classification of Democrat (`RePID`$= -1$), Intermediate (`RePID`$= 0$ and Republican (`RePID`$=$ $+1$), respectively.

```
### Read data and define variables
> dati=read.table("C:/.../nes_96.txt",header=T)
```

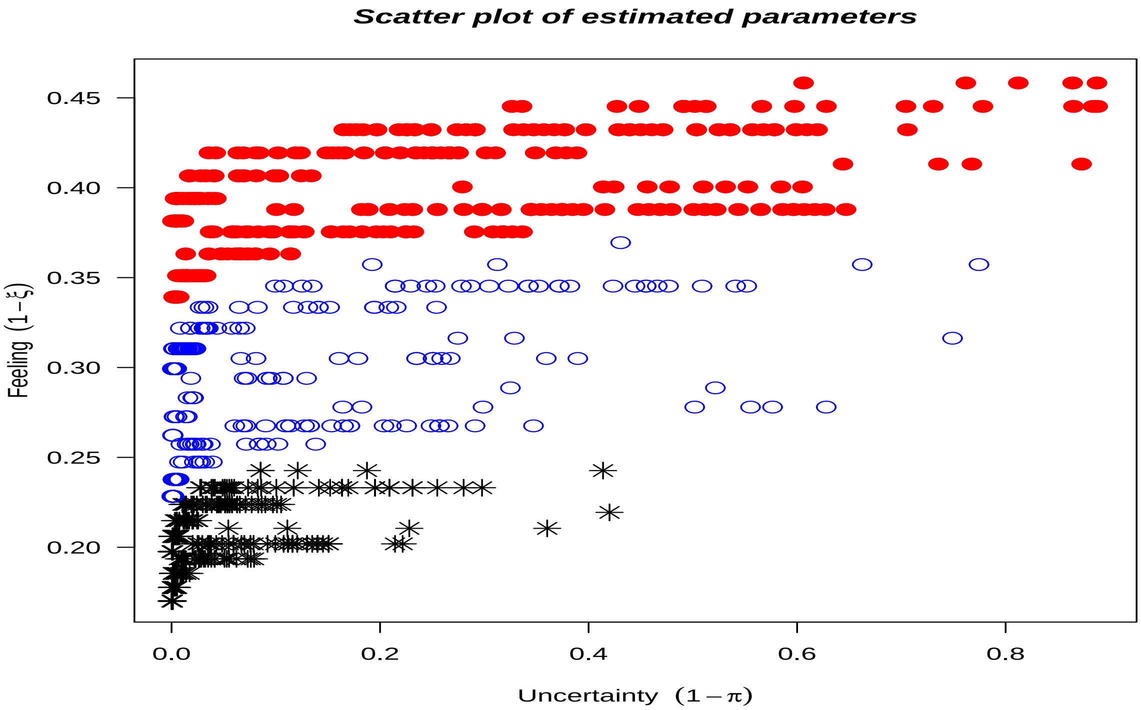

Figure 4: A SPE plot

```
> ClinLR=dati$ClinLR                                      ### 1...7
> PID=dati$PID                                            ### 1...7
> Age=dati$age                                            ### 19...91
> Educ=dati$educ                                          ### 1...7
> n=length(ClinLR)                                        ### n=944
### Estimate CUB model with covariates
> source("CUB.R")
> m=7
> Y=cbind(PID,Age,Educ); W=cbind(PID,Educ)
> CUB(ClinLR,Y,W)
### Numerical estimates of the parameters
#####################
### bet=c(-5.985787, 0.355111, 0.043023, 1.284132)
### gama=c(-0.068924, 0.183626, 0.052654)
#####################
### Recode PID by means of RePID
> RePID=rep(NA,n)
> RePID[PID==1 | PID==2]=-1
```

```
> RePID[PID==3 | PID==4 | PID==5]=0
> RePID[PID==6 | PID==7]=1
#################
### make the SPE plot
> paivet=logis(Y,bet)
> csivet=logis(W,gama)
### Figure 3
> titolo="Scatter plot of estimated parameters"
> simboli=rep(19,n); simboli[repid==0]=1; simboli[repid==1]=8;
> colori=rep("red",n); colori[repid==0]="blue"; colori[repid==1]="black"
> plot(1-paivet,1-csivet,xlim=c(0,0.89),ylim=c(0.17,0.46),
      cex=1.5,pch=simboli,col=colori,
      xlab=expression(1-pi),ylab=expression(1-xi),
      main=titolo,font.main=4)
```

With adequate codes for symbols and colours of the points, Figure 4 clearly visualizes a different behaviour of the respondents as function of `RePID` and make easier the interpretation of the results. Notice that the estimated points $(\hat{\pi}_i, \hat{\xi}_i)$, for $i = 1, 2, \ldots, n$ are functions of the values of the subjects' covariates via a logit link computed thanks to the program function `logis()` included in the main program.

9 Empirical evidence of CUB models

Ordinal data arise in several applied and scientific fields; thus, the applications of related models are pervasive in the statistical literature.

With regard to the application of CUB models we list the topics where they have been successfully applied at the best of our knowledge (ranking* and rating analysis are reported).

- Applications mainly related to *Preferences*:
  - Colors (young people, children, air force cadets)*
  - Cities where to live*
  - Professions for students of Political Sciences graduates*
  - Olive oils preference
  - Coffee preference
  - Sensometric analysis and consumers' behaviours
  - Typical agri-products of South of Italy
  - Italian newspapers*
  - Political affairs: Left/Right self-placement
- Applications mainly related to *Evaluations*:
  - Orientation services
  - University teaching and structures
  - Services for E-bay users

  - Repeatability and reproducibility in MSA
  - Characteristics of bus transports towards a metropolitan area*
  - Degree of preference for buying equo-solidal agricultural products
  - Quality of services in a protected area
  - Customers' satisfaction of European consumers towards salmon
  - Judgment of a city administration
  - Final degree of University graduates
  - Questionnaire validation for patient satisfaction

- Applications mainly related to *Perceptions*:
  - Urban audit surveys about city emergencies*
  - Perceived risk in a printing factory
  - Chronic pain threshold in TMD
  - Synonymy and semantic space of words*
  - Ethnical identity of immigrants by cohorts*
  - European Union objectives and policies*
  - Perception of Economic Security in SHIW
  - Measure of Happiness
  - Job satisfaction in SHIW
  - Job satisfaction of Italian graduates
  - Subjective survival probability to 75 and 90 years
  - Importance-Performance analysis in marketing research
  - Coffee tasting
  - Consumer perception of wine attributes
  - Level of teachers' stress
  - Intention to HPV vaccination
  - Intention to seasonal influenza vaccination

10 Bibliographic notes

The CUB model framework started with Piccolo (2003) and it has been mainly popularized for rank data by D'Elia and Piccolo (2005). Inferential issues for estimation and testing purposes have been established in Piccolo (2006). The identifiability of CUB models has been proved by Iannario (2010).

The modern approach is presented in Corduas *et al.* (2009); Iannario (2012a). Updated references are Iannario and Piccolo (2012a) and Iannario (2012a). Preliminary estimates have been repeatedly tested (Iannario, 2008, 2009b, 2012c) and specific fitting measures for ordinal data models have been proposed (Iannario, 2009a).

The extension of CUB models with subjects' covariates has been obtained by Piccolo (2006); Iannario and Piccolo (2010). The analysis with both subjects' and objects' covariates has been firstly performed in Piccolo and D'Elia (2008).

The consideration of a *shelter effect* has been studied by Iannario (2012a) and successfully applied in Corduas *et al.* (2009). The introduction of covariates in a CUB model with a *shelter effect* has been introduced by Iannario and Piccolo (2012b) with the definition of G*e*CUB models: for these models a program written in the *GAUSS* language is currently available.

CUBE models have been proposed and estimated by Iannario (2012b, 2013); the development of CUBE models with covariates is due to Piccolo (2014).

Specialized extensions of the CUB family of statistical models are the following:

- Hierarchical CUB models (HCUB): Iannario (2012d)
- CUB models in case of complex designs: Gambacorta *et al.* (2013)
- Latent class of CUB models (LC-CUB): Grilli *et al.* (2013)
- CUB models with a varying uncertainty (VCUB): Gottard *et al.* (2013)
- Nonlinear CUB models (NL-CUB): Manisera and Zuccolotto (2013)
- Generalized mixtures model with uncertainty: Iannario and Piccolo (2013b)

A comprehensive comparison among CUB and classical models for ordinal data has been prepared by Iannario and Piccolo (2013a).

These bibliographic notes are not exhaustive since they have been limited to the main methodological papers which originated the framework of CUB models and their extensions. More specific contributions have been added by several researchers.

A complete list of all papers related to CUB models and their applications is available by Authors on request and will be the object of a future revised version of this publication.

Acknowledgement: This paper is a completely new and updated version of Iannario and Piccolo (2009) which refers to CUB program version 2.0. This work has been partly supported by FIRB2012 project at University of Perugia.

References

Corduas M., Iannario M. and Piccolo D. (2009). A class of statistical models for evaluating services and performances, in: M.Bini *et al.* (eds.): *Statistical methods for the evaluation of educational services and quality of products*, Contribution to Statistics, Berlin Heidelberg: Physica-Verlag, Springer, pp.99–117.

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

Gambacorta R., Iannario M. and Vaillant R. (2013). Design-based inference in a mixture model for ordinal variables for a two stage stratified design. *Submitted.*

Gottard A., Iannario M. and Piccolo D. (2013). Varying uncertainty in CUB models. *Submitted.*

Grilli L., Iannario M., Piccolo D. and Rampichini C. (2013). Latent Class CUB Models, *Advances in Data Analysis and Classification*, **7**, DOI10.1007/s11634-013-0143-5.

Iannario M. (2008). Selecting feeling covariates in rating surveys, *Rivista di Statistica Applicata*, **20**, 103–116.

Iannario M. (2009a). Fitting measures for ordinal data models, *Quaderni di Statistica*, **11**, 39–72.

Iannario M. (2009b). A comparison of preliminary estimators in a class of ordinal data models, *Statistica & Applicazioni*, **VII**, 25–44.

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

Iannario M. (2012a). Modelling *shelter* choices in a class of mixture models for ordinal responses, *Statistical Methods and Applications*, **21**, 1–22.

Iannario M. (2012b). CUBE models for interpreting ordered categorical data with overdispersion, *Quaderni di Statistica*, **14**, 137–140.

Iannario M. (2012c). Preliminary estimators for a mixture model of ordinal data, *Advances in Data Analysis and Classification*, **6**, 163–184.

Iannario M. (2012d). Hierarchical CUB models for ordinal variables, *Communications in Statistics. Theory and Methods*, **41**, 3110–3125.

Iannario M. (2013). Modelling uncertainty and overdispersion in ordinal data, *Communications of Statistics. Theory and Methods*, **42**, forthcoming.

Iannario M. and Piccolo D. (2009). A program in R for CUB models inference, Version 2.0, available at *http://www.dipstat.unina.it/CUBmodels1/*

Iannario M. and Piccolo D. (2010). A New Statistical Model for the Analysis of Customer Satisfaction, *Quality Technology & Quantitative Management*, **7**, 149–168.

Iannario M. and Piccolo D. (2012a). CUB models: Statistical methods and empirical evidence, in: Kenett R. S. and Salini S. (eds.). *Modern Analysis of Customer Surveys: with applications using R*, Chichester: J. Wiley & Sons, pp. 231–258.

Iannario M. and Piccolo D. (2012b). A Framework for Modelling Ordinal Data in Rating Surveys, *Proceedings of the Joint Statistical Meetings*, Section on Statistics in Marketing. San Diego, California, pp.3308-3322.

Iannario M. and Piccolo D. (2013a). A comparative analysis of alternative frameworks for modelling ordinal data, *Preliminary report*.

Iannario M. and Piccolo D. (2013b). A generalized framework for modelling ordinal data, *Preliminary report*.

Manisera M. and Zuccolotto M. (2013). Nonlinear CUB models, *Proceedings of the* 9^{th} *Scientific Meeting of CLADAG*, University of Modena and Reggio Emilia, pp.1–4.

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

Piccolo D. (2006). Observed information matrix for MUB models, *Quaderni di Statistica*, **8**, 33–78.

Piccolo D. (2014). Inferential issues in CUBE models with covariates, *Communications in Statistics. Theory and Methods*, forthcoming.

Piccolo D. and D'Elia A. (2008). A new approach for modelling consumers' preferences, *Food Quality and Preference*, **19**, 247–259.