Tutorial on the package hmmm

Manuela Cazzaro
Università di Milano-Bicocca

Roberto Colombi
Università di Bergamo

Sabrina Giordano
Università della Calabria

Abstract

In this tutorial we show how complete hierarchical multinomial marginal (HMM) models for categorical variables can be defined, estimated and tested using the **hmmm** package.

Keywords: marginal models, generalized interactions, chi-bar-square distribution.

1. Introduction

Marginal models are defined for categorical variables by imposing restrictions on marginal distributions of contingency tables, (Agresti 2013, Ch 12). A complete hierarchical multinomial marginal (HMM) model is specified by an ordered set of marginal distributions and a set of interactions (contrasts of logarithms of sums of probabilities) defined within different marginal distributions according to the rules of hierarchy and completeness, see Bergsma and Rudas (2002), Bartolucci, Colombi, and Forcina (2007).

By imposing equality and inequality constraints on marginal interactions, interesting hypotheses (i.e., independence in sub-tables where some categories are collapsed, association in marginal tables, conditional independence or additive effects of covariates in marginal tables, marginal homogeneity, monotone dependence, positive association, among others) can be tested in HMM models.

We developed a new package **hmmm** for the R statistical programming environment (R Core Team 2012) for estimating and testing HMM models with equality and inequality constraints on marginal parameters. The R package **hmmm** is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=hmmm`.

The class of models that the **hmmm** package enables us to deal with is wide since the complete hierarchical marginal models are a generalization of several models proposed in the literature of categorical data analysis. For example, *log-linear models* are HMM models where all the interactions are defined within the joint distribution. The Bergsma and Rudas (2002) *marginal models* are HMM models where the interactions of log-linear type are defined in different marginal distributions. Bartolucci *et al.* (2007) proposed an extension of the Bergsma and Rudas HMM models involving more general types of interactions, while Glonek and McCullagh (1995) *multivariate logistic models* are HMM models which use all the marginal distributions and the parameters are the highest order interactions that can be defined within every marginal distribution.

Furthermore, other models that can be treated with **hmmm** are *hidden Markov models* with observed categorical variables whose distributions conditioned by the latent states are defined

as HMM models and Lang (2004) *multinomial Poisson homogeneous* (MPH) *models* which include HMM models as special cases.

Marginal models are introduced in Section 2 and the functions of the package **hmmm** for defining, estimating and testing HMM models with equality constraints on marginal interactions are illustrated in Sections 3, 4, 5, 6. Sections 4, 5 present more general types of interactions while Section 6 deals with models for repeated measures. The interactions are allowed to depend on covariates in Section 7. Section 8 is devoted to inequality constrained HMM models and Section 9 shows how Lang MPH models, subject to inequality constraints, can be estimated using the package. Final remarks complete the work.

2. Basic concepts

Consider q categorical variables denoted by the first q integers. The set of all variables is $\mathcal{Q} = \{1, 2, ..., q\}$ and a subset of variables which defines a given marginal distribution is denoted by the subset $\mathcal{M}$ of the corresponding integers, $\mathcal{M} \subseteq \mathcal{Q}$.

The vector containing the cell probabilities of the joint distribution is denoted by $\boldsymbol{\pi}$. A one-to-one function $\boldsymbol{\eta} = g(\boldsymbol{\pi})$ defines a parameterization of $\boldsymbol{\pi}$ in terms of $\boldsymbol{\eta}$.

In the complete hierarchical multinomial marginal models, the elements of $\boldsymbol{\eta}$ are parameters called *marginal interactions*. The marginal interactions are contrasts of logarithms of sums of probabilities defined within different marginal distributions associated to a non-decreasing sequence of marginal sets $\mathcal{M}_1, \ldots, \mathcal{M}_k$ ($\mathcal{M}_k = \mathcal{Q}$) according to the rules of hierarchy and completeness. More specifically, in complete hierarchical multinomial marginal models, every interaction is defined in one and only one marginal distribution (completeness) and within the first marginal set which contains it (hierarchy). For instance, given three binary variables, and the marginal sets $\mathcal{M}_1 = \{1\}, \mathcal{M}_2 = \{1, 2\}, \mathcal{M}_3 = \{1, 2, 3\}$, the interactions in $\boldsymbol{\eta}$ are three logits, three log-odds ratios and a third-order interaction defined as follows: a logit is defined on the univariate distribution of variable 1, a second logit and a log-odds ratio are defined on the bivariate distribution of the first two variables. More precisely the second logit is defined on the conditional distribution of variable 2 given that the first variable is at the reference category. All the remaining interactions (a logit, two log-odds ratios and the third-order interaction) involve variable 3 and are defined in the set $\mathcal{M}_3$.

The elements of $\boldsymbol{\eta}$, defined on the marginal distribution of the variables in $\mathcal{M}$, are specified by assigning a logit type to each variable $i \in \mathcal{M}$ among the 5 different types: baseline (`b`) $\eta_i(x; b) = \log\{Pr(i = x)\} - \log\{Pr(i = 1)\}$ (the reference category is the first), local (`l`) $\eta_i(x; l) = \log\{Pr(i = x)\} - \log\{Pr(i = x - 1)\}$, global (`g`) $\eta_i(x; g) = \log\{Pr(i > x - 1)\} - \log\{Pr(i \leq x - 1)\}$, continuation (`c`) $\eta_i(x; c) = \log\{Pr(i > x - 1)\} - \log\{Pr(i = x - 1)\}$ and reverse continuation (`rc`) $\eta_i(x; rc) = \log\{Pr(i = x)\} - \log\{Pr(i \leq x - 1)\}$, with $x = 2, ..., c_i$, where c_i is the number of categories of the variable i. Log-odds ratios and higher-order interactions are defined as contrasts of the mentioned logits as shown by Bartolucci *et al.* (2007), Douglas, Fienberg, Lee, Sampson, and Whitaker (1990), among others. For example, if logits of type (`g`) and (`c`) are used for variable i and j respectively, the following log-odds ratios of type global-continuation (`gc`) are defined as $\eta_{ij}(x_1, x_2; g, c) = \eta_j(x_2; c|i > x_1 - 1) - \eta_j(x_2; c|i \leq x_1 - 1) = \eta_i(x_1; g|j > x_2 - 1) - \eta_i(x_1; g|j = x_2 - 1)$ with $x_1 = 2, ..., c_i$ and $x_2 = 2, ..., c_j$. Moreover, if logit baseline (`b`) is assigned to a third variable, third-order interactions are of global-continuation-baseline type (`gcb`) defined as

$\eta_{ijk}(x_1, x_2, x_3; g, c, b) = \eta_i(x_1; g|j > x_2 - 1, k = x_3) - \eta_i(x_1; g|j = x_2 - 1, k = x_3) - \eta_i(x_1; g|j > x_2 - 1, k = 1) + \eta_i(x_1; g|j = x_2 - 1, k = 1)$, with $x_1 = 2, ..., c_i$, $x_2 = 2, ..., c_j$ and $x_3 = 2, ..., c_k$. A similar reasoning holds for higher-order interactions.

In the Bergsma and Rudas models the components of $\boldsymbol{\eta}$ are log-linear parameters defined in marginal distributions (only baseline type `b` logits are used), while in the Bartolucci *et al.* parameterization all the previous logits can be used and $\boldsymbol{\eta}$ is called a vector of *generalized marginal interactions* which are more meaningful when the variables have an ordinal nature. Moreover, Cazzaro and Colombi (2013) proposed another type of parameters, called *recursive* (or *nested*) *marginal interactions* based on a new type of logits (recursive logits, `r`) which will be described in Section 5.

The vector $\boldsymbol{\eta}$ can be written in matrix form as $\boldsymbol{C}log(\boldsymbol{M}\boldsymbol{\pi})$ where the rows of the matrix $\boldsymbol{C}$ are contrasts, $\boldsymbol{M}$ is a zero-one matrix which sums the probabilities of appropriate cells, and the operator $log(.)$ is coordinate-wise. See the appendix of Bartolucci *et al.* (2007) for the construction of the $\boldsymbol{C}, \boldsymbol{M}$ matrices.

Conditional independencies among variables can be considered by imposing simple zero restrictions on the model parameters as $\boldsymbol{E}\boldsymbol{\eta} = \boldsymbol{0}$ (Sections 3, 4, 5, 6), the effect of covariates on responses can be modelled by a *linear predictor* as $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ (Section 7) and hypotheses of stochastic dominance or positive association bear on inequality constraints as $\boldsymbol{D}\boldsymbol{\eta} \geq \boldsymbol{0}$ (Section 8).

In the **hmmm** package, HMM models involving equality and inequality constraints are seen as special cases of MPH models (Cazzaro and Colombi 2009) and are estimated by maximizing the log-likelihood function of a *reference log-linear model* under the constraints: $\boldsymbol{E}\boldsymbol{\eta} = \boldsymbol{0}$ or $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ and $\boldsymbol{D}\boldsymbol{\eta} \geq \boldsymbol{0}$ through a modified version of the algorithm proposed by Lang (2004) for equality constraints only. The reference log-linear model is usually the saturated one, though not necessary.

3. How to define and estimate marginal models

The starting point for the marginal modelling of categorical data is a multidimensional table providing the joint distribution of two or more unordered and/or (partially) ordered categorical variables.

In this section, we will describe the main functions of the **hmmm** package to handle marginal models. Three are the key steps: load the vector of counts, define the HMM model through the function `hmmm.model()`, estimate and test the model using `hmmm.mlfit()`. We will go through each step to illustrate the flexibility and potential of the package.

In the **hmmm** package, the input data `y` must be a vectorized contingency table. The following example clarifies how the cell frequencies are arranged in `y`.

To start with, we consider the `accident` data. This data frame regards accidents occurred to 1052 workers of a northern Italian city in 1998 who claimed for a compensation. The data are provided by INAIL, the Italian institute for insurance against factory accidents and concern the variables: `Type` of the injury (with 3 levels: `uncertain`, `avoidable`, `not-avoidable`), `Time` to recover (number of working days lost, an indicator of seriousness of the injury, with 4 levels: `0 |- 7`, `7 |- 21`, `21 |- 60`, `>= 60`), `Age` of the worker (years, with 3 levels: `<= 25`, `26 - 45`, `> 45`) and solar `Hour` (part of the day in which the accident occurred, with 2 levels: `morning`, `afternoon`).

Data are in an aggregated case form where the last column stores the counts for each configuration of the variables. As an example, look at the first 20 rows of the data frame `accident`

```
R> library("hmmm")
R> data("accident", package = "hmmm")
R> accident[1:20,]
```

```
            Type      Time      Age    Hour Freq
1      uncertain   0 |-- 7     <=25 morning   21
2      avoidable   0 |-- 7     <=25 morning    9
3  not-avoidable   0 |-- 7     <=25 morning    0
4      uncertain  7 |-- 21     <=25 morning   10
5      avoidable  7 |-- 21     <=25 morning    9
6  not-avoidable  7 |-- 21     <=25 morning    0
7      uncertain 21 |-- 60     <=25 morning    5
8      avoidable 21 |-- 60     <=25 morning    1
9  not-avoidable 21 |-- 60     <=25 morning    1
10     uncertain     >= 60     <=25 morning    2
11     avoidable     >= 60     <=25 morning    0
12 not-avoidable     >= 60     <=25 morning    1
13     uncertain   0 |-- 7 26 -- 45 morning   78
14     avoidable   0 |-- 7 26 -- 45 morning   51
15 not-avoidable   0 |-- 7 26 -- 45 morning    1
16     uncertain  7 |-- 21 26 -- 45 morning   46
17     avoidable  7 |-- 21 26 -- 45 morning   28
18 not-avoidable  7 |-- 21 26 -- 45 morning    5
19     uncertain 21 |-- 60 26 -- 45 morning   15
20     avoidable 21 |-- 60 26 -- 45 morning   21
```

Note that in **hmmm**, the variables have to be denoted by integers, the lower the number identifying the variable, the faster its categories change in the vectorized contingency table. As an example, in the data frame `accident`, the categories of variable `Type` change faster so in **hmmm** `Type` is var. 1. Variable `Time` changes after `Type` so `Time` is var. 2, `Age` varies afterwards `Type` and `Time` so it is var. 3, and `Hour` is var. 4.

Now we show how to get a vector of labeled frequencies from the data frame `accident`. The length of the row names is controlled by the `st` argument. Row names identify the cells of the contingency table and are used in the outputs displaying estimated cell probabilities. Only the first three rows are printed to give an example

```
R> y <- getnames(accident, st = 9)
```

```
     cell names                      counts
[1,] uncertain 0 |-- 7 <=25 morning 21
[2,] avoidable 0 |-- 7 <=25 morning 9
[3,] not-avoid 0 |-- 7 <=25 morning 0
```

In general, the data can be also organized in a data frame with a separated row for each case or in a contingency table form, but for using the command `getnames` in these cases, the data have to be coerced into the aggregated case form.

We can now define, estimate and test HMM models for these data. Let us start by defining a saturated HMM model, i.e., a model without any restrictions on the interactions.

As mentioned in Section 2, for defining a HMM model, first the sequence of marginal sets and the type of logit assigned to each variable within the sets have to be declared. The command `marg.list()` serves this need. Here, with respect to the `accident` data, the chosen marginal sets are: the bivariate distribution of the variables 3, 4; the two joint distributions of the variables 1, 3, 4 and 2, 3, 4 and the joint distribution of the four variables. For each variable in a marginal set the corresponding logit symbol is inserted (`b` baseline, `g` global, `c` continuation, `rc` reverse continuation, `r` recursive, `l` local), while the variables not included in the marginal set are denoted by `marg`. So, for example, in the statement below, `"marg-marg-b-b"` indicates the first marginal set involving variables 3, 4 both with baseline logits. In this example, all the log-linear interactions in every marginal set are of baseline type (Sections 4 and 5 are devoted to illustrate the use of more general types of interactions)

```
R> margin <- marg.list(c("marg-marg-b-b", "b-marg-b-b",
+  "marg-b-b-b", "b-b-b-b"))
```

The function `hmmm.model()` in the next statement defines the HMM model and creates the list of interactions on the marginal distributions declared by `marg.list()`. In the arguments of `hmmm.model()`, as well as `marg` to which the output of `marg.list()` is assigned, information on the number of categories `lev` and on the `names` of the variables in the stated order are also given

```
R> model <- hmmm.model(marg = margin, lev = c(3, 4, 3, 2),
+  names = c("Type", "Time", "Age", "Hour"))
R> model
```

```
      inter. inter.names        marg. marg.names         type npar start end
 [1,] 3      Age                34    Age,Hour           b    2    1     2
 [2,] 4      Hour               34    Age,Hour           b    1    3     3
 [3,] 34     Age.Hour           34    Age,Hour           bb   2    4     5
 [4,] 1      Type               134   Type,Age,Hour      b    2    6     7
 [5,] 13     Type.Age           134   Type,Age,Hour      bb   4    8     11
 [6,] 14     Type.Hour          134   Type,Age,Hour      bb   2    12    13
 [7,] 134    Type.Age.Hour      134   Type,Age,Hour      bbb  4    14    17
 [8,] 2      Time               234   Time,Age,Hour      b    3    18    20
 [9,] 23     Time.Age           234   Time,Age,Hour      bb   6    21    26
[10,] 24     Time.Hour          234   Time,Age,Hour      bb   3    27    29
[11,] 234    Time.Age.Hour      234   Time,Age,Hour      bbb  6    30    35
[12,] 12     Type.Time          1234  Type,Time,Age,Hour bb   6    36    41
[13,] 123    Type.Time.Age      1234  Type,Time,Age,Hour bbb  12   42    53
[14,] 124    Type.Time.Hour     1234  Type,Time,Age,Hour bbb  6    54    59
[15,] 1234   Type.Time.Age.Hour 1234  Type,Time,Age,Hour bbbb 12   60    71
```

The output lists the parameters of the model (elements of the parameter vector $\boldsymbol{\eta}$ described in Section 2) and illustrates how they are allocated according to the principle of hierarchy and completeness. In particular, the first two columns (`inter.`, `inter.names`) indicate the interactions through integers and the names of the variables they refer to, columns 3 and 4 describe the marginal distributions (`marg.`, `marg.names`) where the interactions are defined, the `type` of logit assigned to the involved variables are specified in column 5, the number (`npar`) of interactions is displayed in column 6 and the first and last positions they occupy in the vector of parameters are indicated in the last two columns `start`, `end`. For example, the first row of the output reveals that interactions `3`, related to var. 3 `Age`, are defined within the marginal distribution `34` of variables `Age,Hour`. They are two (`2` in column `npar`) baseline type (`b`) logits which occupy the first two positions (`1` in column `start`, `2` in column `end`) in the vector of ordered parameters of the model. The two logits are calculated on the conditional distribution of var. 3 given that var. 4 assumes the reference category. Moreover, in the third row, the two interactions `34` in 4th and 5th positions are baseline (`bb`) log-odds ratios. These interactions are defined in the first marginal distribution `34`, so that, for the principle of hierarchy and completeness, they cannot be defined in the successive marginal distributions `134`, `234`, `1234`. The rest of the output is interpreted similarly.

Once the parameters of the model are known, we can specify how to constrain them for satisfying some hypotheses. A non-saturated model can be defined by imposing equality constraints on certain interactions. For example, we can set to zero the interactions that occupy the positions `12:13`, `14:17` (reported in the columns `start`, `end` of the previous output) in the vector of the parameters in order to state that the conditional independence $1 \perp\!\!\!\perp 4 \mid 3$ holds for the variables at hand. This can be achieved by specifying the argument `sel` of the `hmmm.model()` function

```
R> modelB <- hmmm.model(marg = margin, lev = c(3, 4, 3, 2),
+  names = c("Type", "Time", "Age", "Hour"),
+  sel = c(12:13, 14:17))
```

The model is then estimated by the command `hmmm.mlfit()` whose arguments are the vector of data frequencies and the model

```
R> modB <- hmmm.mlfit(y, modelB)
R> modB

SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 6 ):  Gsq =  6.02965 (p =  0.41988 )
```

The model shows a good fit. Further, estimated parameters can be printed by the following statement

```
R> print(modB, aname = "model B", printflag = TRUE)
```

A much more detailed output with estimated standard errors and estimated cell probabilities is given by

```
R> summary(modB)
```

Note that, the command `summary()` shows also the uncostrained estimates of parameters calculated on the sample frequencies, say `OBS LINK`. It may happen that certain sample frenquencies are null thereby implying that some estimates cannot be determined, and in this case, the `summary()` of the model displays `NaN` in the columns `OBS LINK` and `LINK RESID`.

When the constrained interactions are log-linear parameters defined in the joint distribution (Agresti 2013), it is convenient to use the argument `formula` of the `hmmm.model()` function for specifying the log-linear model without the interactions we impose to be zero. For example, if in addition to the previous constraints, we would like to verify also whether the odds ratios of `Type` and `Time`, in the joint distribution, do not depend on the levels of `Age` and `Hour`, we must set to zero the interactions of the third and fourth order arranged in the positions from 42 to 71. These log-linear interactions are defined in the joint distribution and we can use the statements

```
R> modelA <- hmmm.model(marg = margin, lev = c(3, 4, 3, 2),
+  names = c("Type", "Time", "Age", "Hour"), sel = c(12:13, 14:17),
+  formula = ~ Type * Age * Hour + Time * Age * Hour + Type : Time)
R> modA <- hmmm.mlfit(y, modelA)
```

Thus, `modelA` is nested in `modelB`. The likelihood ratio test to compare the two nested models is obtained by the function `anova()`

```
R> anova(modA, modB)

        statistics value df    pvalue
model A        34.589455 36 0.5356700
model B         6.029646  6 0.4198800
LR test        28.559810 30 0.5407972
```

The last row of the `anova()` reports the likelihood ratio test of hypothesis H_0 : `modelA` versus H_1 : `modelB`, and in this case, it reveals that the more parsimonious `modelA` cannot be rejected. First and second rows show the goodness-of-fit of both models tested against the saturated model, already displayed in the output of `hmmm.mlfit()`.

Note that the previous `modelA` is not log-linear because some constrained interactions are defined in marginal distributions. On the contrary, if `modelA` is defined without constraints on the marginal interactions `12:13, 14:17`, it is log-linear and can be also defined by the specific function `loglin.model()` as follows

```
R> modellog <- loglin.model(lev = c(3, 4, 3, 2),
+  formula = ~ Type * Age * Hour + Time * Age * Hour + Type : Time,
+  names = c("Type", "Time", "Age", "Hour"))
R> modlog <- hmmm.mlfit(y, modellog)
```

4. Generalized marginal interactions

In the previous section all the interactions defined within the marginal distributions are of baseline type. Bartolucci *et al.* (2007) have shown that more general types of interactions

can be used to parameterize marginal models. This possibility is particularly useful because, in presence of ordered categorical variables, the univariate marginal distributions are parameterized more appropriately using non standard logits such as the global and continuation ones for example, and bivariate distributions are parameterized by non standard odds ratios such as the global, global-continuation and the continuation ones. This extension is also important since several hypotheses of restrictive association and monotone dependence can be expressed by equality and inequality constraints on these generalized interactions (in Section 8 the usefulness of these interactions for testing hypotheses of stochastic orderings is clarified).

In this section, we will illustrate an example where HMM models with generalized marginal interactions are defined.

Remind that the `marg.list()` command is used to make clear the logit types assigned to the variables in a marginal distribution as any generalized interaction depends on them.

For example, we consider the `madsen` data (Madsen 1976) concerning 1681 rental property residents who are classified according to the following variables: feeling of `Influence` on the apartment management (var. 1 with 3 ordinal levels: `low`, `medium`, `high`), `Satisfaction` with housing conditions (var. 2 with 3 ordinal levels: `low`, `medium`, `high`), degree of `Contact` with other residents (var. 3 with 2 levels: `low`, `high`), type of `Housing` (var. 4 with 4 levels: `tower block`, `apartment`, `atrium house`, `terraced house`).

For the `madsen` data, let us consider the statement

```
R> margin <- marg.list(c("marg-marg-l-l", "g-marg-l-l",
+  "marg-g-l-l", "g-g-l-l"))
```

This means that in the bivariate distribution of variables 3, 4 all the interactions are of local (`l`) type, while in the joint distribution of 1, 3, 4 the interactions `1` are global (`g`) logits, the interactions `13` and `14` are global-local (`gl`) log-odds ratios. In this marginal distribution, the interactions `134` are differences between the logarithms of two global-local odds ratios. A similar comment holds for the joint distribution of the variables 2, 3, 4.

The model is defined as

```
R> model <- hmmm.model(marg = margin, lev = c(3, 3, 2, 4),
+  names = c("In", "Sa", "Co", "Ho"))
R> model
```

```
      inter. inter.names marg. marg.names  type npar start end
 [1,] 3      Co          34    Co,Ho       l    1    1     1
 [2,] 4      Ho          34    Co,Ho       l    3    2     4
 [3,] 34     Co.Ho       34    Co,Ho       ll   3    5     7
 [4,] 1      In          134   In,Co,Ho    g    2    8     9
 [5,] 13     In.Co       134   In,Co,Ho    gl   2    10    11
 [6,] 14     In.Ho       134   In,Co,Ho    gl   6    12    17
 [7,] 134    In.Co.Ho    134   In,Co,Ho    gll  6    18    23
 [8,] 2      Sa          234   Sa,Co,Ho    g    2    24    25
 [9,] 23     Sa.Co       234   Sa,Co,Ho    gl   2    26    27
[10,] 24     Sa.Ho       234   Sa,Co,Ho    gl   6    28    33
[11,] 234    Sa.Co.Ho    234   Sa,Co,Ho    gll  6    34    39
```

```
[12,] 12     In.Sa       1234  In,Sa,Co,Ho gg   4    40    43
[13,] 123    In.Sa.Co    1234  In,Sa,Co,Ho ggl  4    44    47
[14,] 124    In.Sa.Ho    1234  In,Sa,Co,Ho ggl  12   48    59
[15,] 1234   In.Sa.Co.Ho 1234  In,Sa,Co,Ho ggll 12   60    71
```

If there is an additive effect of variables 3 and 4 on the global (g) logits of variable 1 in the marginal distribution 134, the global-local-local (gll) interactions 18:23 in the above output must be zero and if $2 \perp\!\!\!\perp 3 \mid 4$, that is the global (g) logits of var. 2 are not affected by var. 3 in the marginal distribution 234, the global-local (gl) log-odds ratios 26:27 and the global-local-local (gll) interactions 34:39 must be null. To define and fit the model under the mentioned hypotheses we can run the following statements

```
R> model1 <- hmmm.model(marg = margin, lev = c(3, 3, 2, 4),
+  names = c("In", "Sa", "Co", "Ho"), sel = c(18:23, 26:27, 34:39))
R> data("madsen", package = "hmmm")
R> y <- getnames(madsen, st = 6)
R> mod1 <- hmmm.mlfit(y, model1)
R> mod1
```

```
SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 14 ):  Gsq =  24.49143 (p =  0.039933 )
```

We try to improve the fitting of model1 by removing the hypothesis $2 \perp\!\!\!\perp 3 \mid 4$ but retaining the additive effect of variables 3, 4 on the global logits of variables 1 and 2 in the marginal distributions 134 and 234, respectively. So the interactions in positions 26:27 are no longer null. This model is defined and estimated as follows

```
R> model2 <- hmmm.model(marg = margin, lev = c(3, 3, 2, 4),
+  names = c("In", "Sa", "Co", "Ho"), sel = c(18:23, 34:39))
R> mod2 <- hmmm.mlfit(y, model2)
R> mod2
```

```
SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 12 ):  Gsq =  14.76183 (p =  0.25472 )
```

The model fit is definitely improved.

Moreover, to add the hypothesis that the global (g) log-odds ratios of the variables 1 and 2 do not depend on the levels of variables 3 and 4, the (ggl) and (ggll) interactions, which occupy the positions 44:71 in the vector of parameters, have to be constrained to zero

```
R> model3 <- hmmm.model(marg = margin, lev = c(3, 3, 2, 4),
+  names = c("In", "Sa", "Co", "Ho"), sel = c(18:23, 34:39, 44:71))
R> mod3 <- hmmm.mlfit(y, model3)
R> mod3
```

```
SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 40 ):  Gsq =  45.61355 (p =  0.25008 )
```

This model has a reasonable fit.

For an alternative way of specifying other similar hypotheses see Section 7 where the effect of covariates on interactions is taken into account.

5. Recursive marginal interactions

Cazzaro and Colombi (2013) extended the class of generalized marginal interactions by introducing a new logit type: the recursive (or nested) logits. In the simplest case, these logits are defined in correspondence of a partition of the categories of a variable.

A first set of logits contains the baseline logits defined on the probabilities of the sets of the partition (the reference set can be chosen arbitrarily). A second set includes the baseline logits which are defined within every set of the partition (the reference category can be chosen arbitrarily in every set).

As an example, we consider the `relpol` data, Bergsma, Croon, and Hagenaars (2009, p. 24), on religion and political orientation of a sample of 911 U.S. citizens extracted from the General Social Survey, 1993, with var. 1 `Religion` with levels PR `Protestant`, CA `Catholic`, NO `None` and var. 2 `Politics` with levels EL `Extremely liberal`, LI `Liberal`, SL `Slightly liberal`, MO `Moderate`, SC `Slightly conservative`, CO `Conservative`, EC `Extremely conservative`. For `Religion` we consider the partition with sets R={PR, CA}, N={NO} in order to distinguish between religious and non-religious citizens and for `Politics` we highlight the partition in the sets L={EL, LI, SL}, M={MO} and C={SC, CO, EC}. Note that we introduce the sets L and C as aggregation of categories following an obvious ideological similarity criterion ('Liberals' and 'Conservatives').

Given the proposed partition, the first recursive logit (with reference categories set R) of the variable `Religion` is

$$log\left[\frac{pr(N)}{pr(R)}\right]$$

and the second recursive logit (reference category PR) is

$$log\left[\frac{pr(\text{CA})}{pr(\text{PR})}\right].$$

Focusing on variable `Politics`, the first and the second recursive logits (with reference set L) defined on the probabilities between the sets of the partition are

$$log\left[\frac{pr(C)}{pr(L)}\right] \qquad \text{and} \qquad log\left[\frac{pr(M)}{pr(L)}\right];$$

the recursive logits defined within the sets of the partition are the following four logits

$$log\left[\frac{pr(\text{EL})}{pr(\text{LI})}\right], \quad log\left[\frac{pr(\text{SL})}{pr(\text{LI})}\right], \quad log\left[\frac{pr(\text{SC})}{pr(\text{CO})}\right] \quad \text{and} \quad log\left[\frac{pr(\text{EC})}{pr(\text{CO})}\right],$$

respectively. Note that the reference category is LI for Liberals and CO for Conservatives.

The number of recursive logits is always equal to the number of categories minus one. The use of interactions based on recursive logits is requested in `marg.list()` by the use of `r` instead of `b`, `l`, `g`, `c` and `rc`, (see Section 2 for details)

```
R> marginals <- marg.list(c("r-marg", "marg-r", "r-r"))
```

The recursive logits are specified by the function `recursive()` that requires an argument for every variable. The argument is `0` for every variable to which a recursive logit is not assigned otherwise it is a matrix. This matrix has as many rows as the number of recursive logits of the variable involved and columns equal to the number of the categories of the variable. In particular, the rows of this matrix specify the categories whose probabilities appear in the numerator and denominator of the recursive logits. In a row, a value `1` (`-1`) corresponds to the categories whose probability is cumulated at the numerator (denominator), `0` if the category is not involved.

With reference to var. 1 `Religion`, the following are the statements defining matrix `rec1`

```
R> rec1 <- matrix(c(-1, -1,  1,
+                   -1,  1,  0), 2, 3, byrow = TRUE)
```

To the first row of matrix `rec1` is associated the first logit $log\,[pr(N)/pr(R)]$ of var. 1 `Religion` as it picks out the probabilities pr(PR) and pr(CA) that are summed at the denominator and the probability pr(NO) at the numerator of the logit; second row identifies the second logit of var. 1 in a similar way.

These are the necessary statements to define matrix `rec2` for var. 2 `Politics`

```
R> rec2 <- matrix(c(-1, -1, -1,  0,  1,  1,  1,
+                   -1, -1, -1,  1,  0,  0,  0,
+                    1, -1,  0,  0,  0,  0,  0,
+                    0, -1,  1,  0,  0,  0,  0,
+                    0,  0,  0,  0,  1, -1,  0,
+                    0,  0,  0,  0,  0, -1,  1), 6, 7, byrow = TRUE)
```

The first row, for example, of matrix `rec2` identifies the categories whose probabilities are cumulated at the denominator (pr(EL), pr(LI) and pr(SL)) and at the numerator (pr(SC), pr(CO) and pr(EC)) of the first recursive logit $log\,[pr(C)/pr(L)]$ of var. 2 `Politics`.

The matrices `rec1` and `rec2` are then the arguments of the function `recursive()`

```
R> rec <- recursive(rec1, rec2)
```

Finally the output of `recursive()` must be assigned to the argument `cocacontr` of `hmmm.model()`. Consider the following statements

```
R> model <- hmmm.model(marg = marginals, lev = c(3, 7),
+  names = c("Rel", "Pol"), cocacontr = rec)
R> model
```

```
     inter. inter.names marg. marg.names type npar start end
[1,] 1      Rel         1     Rel        r    2    1     2
[2,] 2      Pol         2     Pol        r    6    3     8
[3,] 12     Rel.Pol     12    Rel,Pol    rr   12   9     20
```

It is worthwhile to remember that the parameters of vector $\boldsymbol{\eta}$ are defined by assigning a logit type to each variable (recursive type in this case) and higher-order interactions (as recursive log-odds ratios in this case) are defined as contrasts of the mentioned logits (for more details on higher-order recursive interactions see Cazzaro and Colombi (2013)). The output shows that the vector $\boldsymbol{\eta}$ of 20 parameters of the models is structured in the following way: the first two parameters are the recursive logits of var. 1 `Religion`; successively, the 6 recursive logits of var. 2 `Politics` are stated and finally the 12 recursive log-odds ratios of the two involved variables complete the parameterization of the model.

In particular, note that the first 8 elements of the vector $\boldsymbol{\eta}$ are the previously presented logits in the order as we described them. This follows from the order of the rows of the `rec1` and `rec2` matrices.

To exemplify the kind of hypotheses that can be modeled with recursive logits and to show as well how linear constraints on marginal interactions can be tested, let us consider the constraints

$$log\left[\frac{pr(\mathrm{EL})}{pr(\mathrm{LI})}\right] - log\left[\frac{pr(\mathrm{EC})}{pr(\mathrm{CO})}\right] = 0 \tag{1}$$

$$log\left[\frac{pr(\mathrm{SL})}{pr(\mathrm{LI})}\right] - log\left[\frac{pr(\mathrm{SC})}{pr(\mathrm{CO})}\right] = 0 \tag{2}$$

stating that the distribution between extreme and moderate attitudes is the same within conservatives and liberals. The condition in Equation 1 equates the 3th and 6th recursive logits of `Politics` that occupy positions 5 and 8 in the vector $\boldsymbol{\eta}$ of parameters, respectively. The condition in Equation 2 equates the 4th and 5th recursive logits that are in positions 6 and 7 in the vector $\boldsymbol{\eta}$. Remind that in the **hmmm** package, equality constraints on HMM models are in the form $\boldsymbol{E\eta} = \mathbf{0}$.

Hence, the previous constraints can be defined by assigning the following constraints matrix `Emat` to the argument `E` of the function `hmmm.model()`. Note that the `Emat` matrix has 2 rows and 20 columns, the number of parameters. Rows 1 and 2 are devoted to constraints reported in Equations 1 and 2, respectively: all the elements of `Emat` are zeros apart from a `1` in the 5th position and a `-1` in the 8th position in the first row; the second row has a `1` in the 6th position and a `-1` in the 7th position

```
R> Emat <- cbind(matrix(0, 2, 4), matrix(c(1, 0, 0, 1, 0, -1, -1, 0), 2, 4),
+  matrix(0, 2, 12))
R> modelE <- hmmm.model(marg = marginals, lev = c(3, 7),
+  names = c("Rel", "Pol"), cocacontr = rec, E = Emat)
```

With reference to the `relpol` data, the following statements fit the model

```
R> data("relpol", package = "hmmm")
R> y <- getnames(relpol, st = 4)
```

```
R> modE <- hmmm.mlfit(y, modelE)
R> print(modE)

SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 2 ):  Gsq =  1.58106 (p =  0.45361 )
```

The tested model has a good fit.

6. Repeated measures on the response variables

Studies where the categorical response variable is observed for each subject repeatedly, under various conditions or at several occasions, are very common in applications. In this section, we show how *repeated measures* can be treated using HMM models subject to equality constraints on marginal interactions.

To this aim we consider an example discussed in Section 12.1.1 of Agresti (2013) and we point out how the marginal logistic models there described can be reformulated as HMM models.

Table 12.1 in Agresti (2013, p. 456) refers to a longitudinal study of mental depression (Koch, Landis, Freeman, Freeman, and Lehnen 1977) for 340 subjects suffering depression classified in four groups according to whether the severity of initial diagnosis is mild or severe and whether the treatment gives standard or new drugs. The study observed the depression assessment of the patients at 3 time occasions ($t = 1, 2, 3$) after treatment. So, there are three response variables: `R1: Depression at t=1`, `R2: Depression at t=2`, `R3: Depression at t=3`, with levels: `normal`, `abnormal`, and two covariates: `T: Treatment` (with levels: `standard`, `new drug`) and `D: Diagnosis` (with levels: `mild`, `severe`). The data are available in the data frame `depression`

```
R> data("depression", package="hmmm")
R> y <- getnames(depression, st = 9)
```

Note that, coherently with the table of data reported in the Agresti's book, in the data frame `depression` the counts are arranged in such a way that `R3`, `R2` and `R1` are var. 1, var. 2 and var. 3, respectively, and var. 4, var. 5 are the covariates `Treatment` and `Diagnosis`.

Agresti proposes a first marginal model to fit these data

$$M_1 : l_{tij} = \alpha + \beta_i^T + \beta_j^D + \beta t$$

where l_{tij} is the logit of the response at occasion t, $t = 1, 2, 3$, given the categories i of `Treatment` ($i = 0$ for `standard drug` and $i = 1$ for `new drug`), and j of `Diagnosis` ($j = 0$ for `mild` and $j = 1$ for `severe`); β_i^T and β_j^D are the main effects of the covariates with a reference category coding $\beta_0^T = 0$, $\beta_0^D = 0$.

Model M_1 is a special case of the saturated logit model

$$M : l_{tij} = \alpha_t + \beta_{ti}^T + \beta_{tj}^D + \beta_{tij}^{T,D}$$

obtained under the eight constraints

$$\beta_{t1}^T - \beta_{11}^T = 0, \quad \beta_{t1}^D - \beta_{11}^D = 0, \quad \beta_{t11}^{T,D} = 0, \quad t = 1, 2, 3, \tag{3}$$

$$\alpha_3 - 2\alpha_2 + \alpha_1 = 0. \tag{4}$$

The constraints in Equation 3 state that the effects of the covariates are additive and do not differ by time occasion. Moreover, under the constraints in Equations 3 and 4, the logits l_{tij} are linear function of time.

Agresti provides a second model which permits the treatment effect to differ by time occasion,

$$M_2 : l_{tij} = \alpha + \beta_{0i}^T + \beta_j^D + \beta t + \beta_{1i}^T t, \quad t = 1, 2, 3.$$

Model M_2 is obtained by imposing to M the seven constraints

$$\beta_{31}^T - 2\beta_{21}^T + \beta_{11}^T = 0, \quad \beta_{t1}^D - \beta_{11}^D = 0, \quad \beta_{t11}^{T,D} = 0, \quad t = 1, 2, 3, \tag{5}$$

$$\alpha_3 - 2\alpha_2 + \alpha_1 = 0. \tag{6}$$

Since models M_1 and M_2 are logistic models, specified on the marginal distribution of each response $1, 2, 3$ and two covariates $4, 5$, we need to insert $\{1, 4, 5\}$, $\{2, 4, 5\}$, $\{3, 4, 5\}$ in the sequence of marginal sets defining the corresponding HMM models. Moreover, we complete the ordered list of marginal sets by adding the full set $\{1, 2, 3, 4, 5\}$ which cannot be omitted, and the set of the covariates $\{4, 5\}$. In this way, all the interactions involving only the covariates will be defined in the first marginal distribution and only the interactions related to the association among the responses at the three time occasions will be defined in the joint distribution, by virtue of the hierarchy and completeness assumptions.

Given the marginal sets, the saturated marginal model for the five variables is defined by the codes

```
R> margin <- marg.list(c("marg-marg-marg-b-b","b-marg-marg-b-b",
+  "marg-b-marg-b-b", "marg-marg-b-b-b","b-b-b-b-b"))
R> name <- c("R3","R2","R1","T","D")
R> modelsat<-hmmm.model(marg = margin, lev = c(2,2,2,2,2), names = name)
R> modelsat
```

```
      inter. inter.names  marg. marg.names   type  npar start end
 [1,] 4      T            45    T,D          b     1    1     1
 [2,] 5      D            45    T,D          b     1    2     2
 [3,] 45     T.D          45    T,D          bb    1    3     3
 [4,] 1      R3           145   R3,T,D       b     1    4     4
 [5,] 14     R3.T         145   R3,T,D       bb    1    5     5
 [6,] 15     R3.D         145   R3,T,D       bb    1    6     6
 [7,] 145    R3.T.D       145   R3,T,D       bbb   1    7     7
 [8,] 2      R2           245   R2,T,D       b     1    8     8
 [9,] 24     R2.T         245   R2,T,D       bb    1    9     9
[10,] 25     R2.D         245   R2,T,D       bb    1    10    10
[11,] 245    R2.T.D       245   R2,T,D       bbb   1    11    11
[12,] 3      R1           345   R1,T,D       b     1    12    12
[13,] 34     R1.T         345   R1,T,D       bb    1    13    13
[14,] 35     R1.D         345   R1,T,D       bb    1    14    14
[15,] 345    R1.T.D       345   R1,T,D       bbb   1    15    15
```

```
[16,] 12    R3.R2       12345 R3,R2,R1,T,D bb    1    16    16
[17,] 13    R3.R1       12345 R3,R2,R1,T,D bb    1    17    17
[18,] 23    R2.R1       12345 R3,R2,R1,T,D bb    1    18    18
[19,] 123   R3.R2.R1     12345 R3,R2,R1,T,D bbb   1    19    19
[20,] 124   R3.R2.T      12345 R3,R2,R1,T,D bbb   1    20    20
[21,] 134   R3.R1.T      12345 R3,R2,R1,T,D bbb   1    21    21
[22,] 125   R3.R2.D      12345 R3,R2,R1,T,D bbb   1    22    22
[23,] 234   R2.R1.T      12345 R3,R2,R1,T,D bbb   1    23    23
[24,] 135   R3.R1.D      12345 R3,R2,R1,T,D bbb   1    24    24
[25,] 235   R2.R1.D      12345 R3,R2,R1,T,D bbb   1    25    25
[26,] 1234  R3.R2.R1.T   12345 R3,R2,R1,T,D bbbb  1    26    26
[27,] 1235  R3.R2.R1.D   12345 R3,R2,R1,T,D bbbb  1    27    27
[28,] 1245  R3.R2.T.D    12345 R3,R2,R1,T,D bbbb  1    28    28
[29,] 1345  R3.R1.T.D    12345 R3,R2,R1,T,D bbbb  1    29    29
[30,] 2345  R2.R1.T.D    12345 R3,R2,R1,T,D bbbb  1    30    30
[31,] 12345 R3.R2.R1.T.D 12345 R3,R2,R1,T,D bbbbb 1    31    31
```

Note that, the parameters α_t, β_{t1}^T, β_{t1}^D, $\beta_{t11}^{T,D}$ of `modelsat` related to the response at the last occasion ($t = 3$) occupy the positions `4:7` in the table above, positions `8:11` for the response at $t = 2$ and `12:15` for $t = 1$.

Thus, the constraints which specify both models M_1 and M_2 involve only the 12 interactions listed from the 4th to the 15th position. For example $\alpha_3 - 2\alpha_2 + \alpha_1 = 0$ constrains the 4th, 8th and 12th parameters; $\beta_{21}^T - \beta_{11}^T = 0$ and $\beta_{31}^T - \beta_{11}^T = 0$ involve the 5th, 9th and 13th parameters; the constraints $\beta_{21}^D - \beta_{11}^D = 0$ involve the 6th, 10th and 14th parameters; $\beta_{t11}^{T,D} = 0$, $t = 1, 2, 3$, refers to 7th, 11th and 15th parameters.

For specifying M_1 and M_2 in terms of HMM models, we cannot use the argument `sel` in the function `hmmm.model`, because the constraints of M_1 and M_2 are not all zero restrictions on single parameters. This is the reason why we specify the constraints on the interactions in the form $\boldsymbol{E}\boldsymbol{\eta} = \boldsymbol{0}$ (the vector $\boldsymbol{\eta}$ here contains the parameters of `modelsat`). Below we give the details of the construction of the matrix $\boldsymbol{E}$ for both models.

For M_1, the matrix $\boldsymbol{E}_1$ has 31 columns as the number of parameters in `modelsat` and 8 rows as the number of the constraints in Equations 3 and 4. Remind that among those 31 parameters, only the 12 interactions listed from the 4th to 15th position are involved and they are here arranged in the sub-vector $\boldsymbol{\eta}_1$ for simplicity. Thus, we need to define the sub-matrix $\boldsymbol{A}_1$ with 8 rows, one row for each constraint of M_1, and 12 columns, one for each parameter in $\boldsymbol{\eta}_1$ such as the equation $\boldsymbol{A}_1\boldsymbol{\eta}_1 = \boldsymbol{0}$ reproduces the constraints in Equations 3 and 4. At this point, we can define the matrix $\boldsymbol{E}_1$ as a three blocks matrix with a zero-values matrix in the first and third blocks (columns 1-3 and 16-31 of $\boldsymbol{E}_1$) corresponding to unconstrained parameters, and the matrix $\boldsymbol{A}_1$ as a middle block (columns 4-15 of $\boldsymbol{E}_1$). In details, we write

```
R> A1<-matrix(c(
+     0,0,0,1,0,0,0,0,0,0,0,0,
+     0,0,0,0,0,0,0,1,0,0,0,0,
+     0,0,0,0,0,0,0,0,0,0,0,1,
+     0,1,0,0,0,0,0,0,0,-1,0,0,
+     0,0,0,0,0,1,0,0,0,-1,0,0,
```

```
+     0,0,1,0,0,0,0,0,0,0,-1,0,
+     0,0,0,0,0,0,1,0,0,0,-1,0,
+     1,0,0,0,-2,0,0,0,1,0,0,0
+     ),8,12,byrow=TRUE)
R> E1<-cbind(matrix(0,8,3), A1, matrix(0,8,16))
```

Now we can define and fit model M_1 by assigning `E1` to the argument `E` of the function `hmmm.model()`

```
R> model1<-hmmm.model(marg = margin, lev =c(2,2,2,2,2), names = name, E = E1)
R> fitmod1 <- hmmm.mlfit(y, model1)
R> fitmod1

SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 8 ):  Gsq =  34.57154 (p =  3.1996e-05 )
```

The fit is really poor as highlighted by Agresti because the model is based on the assumption that the time effect does not vary by treatment. This hypothesis is removed in model M_2.

The matrix $\boldsymbol{E}_2$ computed below follows the same reasoning just detailed and here defines the constraints in Equations 5 and 6 of M_2

```
R> A2<-matrix(c(
+     0,0,0,1,0,0,0,0,0,0,0,0,
+     0,0,0,0,0,0,0,1,0,0,0,0,
+     0,0,0,0,0,0,0,0,0,0,0,1,
+     0,1,0,0,0,-2,0,0,0,1,0,0,
+     0,0,1,0,0,0,0,0,0,0,-1,0,
+     0,0,0,0,0,0,1,0,0,0,-1,0,
+     1,0,0,0,-2,0,0,0,1,0,0,0),
+     7,12,byrow=TRUE)
R> E2<-cbind(matrix(0,7,3), A2, matrix(0,7,16))
```

We can now define and fit model M_2 using `E2` as argument in the next statement

```
R> model2<-hmmm.model(marg = margin, lev = c(2,2,2,2,2), names = name, E = E2)
R> fitmod2 = hmmm.mlfit(y, model2)
R> fitmod2

SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 7 ):  Gsq =  4.23174 (p =  0.75273 )
```

This model fits much better.

We complete the section by specifying and fitting a further model, not considered by Agresti. It is obtained by assuming that in M_2 it also holds that the depression assessment at the last

time R3 is independent of its severity at the first occasion R1 given the intermediate response R2 and the covariates Treatment and Diagnosis.

The zero restrictions on the interactions of modelsat which occupy the positions 17, 19, 21, 24, 26, 27, 29, 31, that are needed to satisfy the just introduced Markov condition, constrain log-linear parameters defined in the joint distribution, so we can use the argument formula of the hmmm.model() function as explained in Section 3. The statements for this final model are reported below

```
R> model3<-hmmm.model(marg = margin, lev = c(2,2,2,2,2), names = name, E = E2,
+                     formula=~R1*R2*T*D+R3*R2*T*D )
R> fitmod3 = hmmm.mlfit(y, model3)
R> fitmod3

SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 15 ):  Gsq =  11.86665 (p =  0.68909 )
```

The model shows a very good fit.

7. Covariates effects on the response variables

Different models can be estimated by taking into account the effects of covariates on the response variables as in Marchetti and Lupparelli (2011) and Glonek and McCullagh (1995).

Note that the models tested in this section are Glonek and McCullagh *multivariate logistic models* with categorical covariate variables.

We consider the accident data (see Section 3 for details), but note that, now, var. 1 Type of the injury (3 levels), var. 2 Time to recover (4 ordinal levels) are considered as response variables, as they describe the nature of the accidents occurred to workers in terms of prevention and seriousness, and var. 3 Age of the worker (3 levels) and var. 4 solar Hour (2 levels) as covariates since Age can be considered as indicator of experience and Hour as indicator of tiredness of the worker. Remind that the lower the variable number, the faster the variable changes in the vectorized table. Furthermore, the categories of the covariates determine the strata and the data must be arranged in such a way that the categories of the response variables change faster than the categories of the covariates.

In order to estimate different models taking into account the covariate effects on the response variables, the list of the marginal sets of the response variables has to be specified (using marg.list()). The necessary statement is

```
R> marginals <- marg.list(c("b-marg", "marg-g", "b-g"))
```

For the accident data, it is stated that in the marginal distribution of Type the interactions are baseline logits, in the marginal distribution of Time the interactions are global logits and in the bivariate distribution of Type and Time the interactions are baseline-global log-odds ratios.

Successively, a list of model formulas, each for every interaction specified above, defining the effects of the covariates, is needed. The following statements account for additive effect of the

covariates Age and Hour on the marginal logits of the response variables Type and Time and on the association (log-odds ratios) between the responses Type and Time

```
R> al <- list(
+  Type = ~ Type * (Age + Hour),
+  Time = ~ Time * (Age + Hour),
+  Type.Time = ~ Type.Time * (Age + Hour)
+  )
```

It is worthwhile to note that each component of the list has the name of the interaction and contains the model formula of the covariate effects on such interaction.

The model that takes into account the covariate effects on the response variables is then specified through the function hmmm.model.X(). Several arguments are included in hmmm.model.X(): the marginal sets (marg), the names of the response variables (names), their number of categories (lev), the names of the covariate variables (fnames) and the number of their categories (strata) but, in particular, the main argument is Formula to which a list as al must be assigned

```
R> model <- hmmm.model.X(marg = marginals, lev = c(3, 4),
+  names = c("Type", "Time"), Formula = al, strata = c(3, 2),
+  fnames = c("Age", "Hour"))
```

The model is then estimated by the command hmmm.mlfit()

```
R> data("accident", package = "hmmm")
R> y <- getnames(accident, st = 9)
R> mod1 <- hmmm.mlfit(y, model)
R> mod1

SUMMARY of MODEL:
OVERALL GOODNESS OF FIT:
    Likelihood Ratio Stat (df= 22 ):  Gsq =  16.47375 (p =  0.7917 )
```

More detailed output (the estimated effects and the estimated standard errors, among others) is given by

```
R> summary(mod1)
```

Note that the covariate effects preceded by the main general effect (Intercept) are listed for every interaction.

The necessary list of model formulas to test another interesting hypothesis, where there is the covariates Age, Hour additive effect on the marginal logits of the responses and the stochastic independence between Type and Time in each sub-table identified by the levels of Age and Hour, is

```
R> alind <- list(
+  Type = ~ Type * Age + Type * Hour,
+  Time = ~ Time * Age + Time * Hour,
+  Type.Time = "zero"
+  )
```

We use `"zero"` to constrain to zero all the interactions of a given type, in this case the log-odds ratios between `Type` and `Time`.

To test the so-called 'Parallel log-odds model', that is if the effect of the covariates `Age` and `Hour` is identical for each of the logits and the log-odds ratios of the responses `Type` and `Time`, we need the following statement

```
R> alpar <- list(
+  Type = ~ Type + Age + Hour,
+  Time = ~ Time + Age + Hour,
+  Type.Time = ~ Type.Time + Age + Hour
+  )
```

8. Inequality constraints on interactions

Hypotheses of monotone dependence and positive/negative association between ordered categorical variables can be ascertained by testing marginal models with inequality constraints on certain interactions. We illustrate how to define, fit and test models with parameters constrained by inequalities using the dataset `polbirth`, Bergsma *et al.* (2009, p. 30), based on the U.S. General Social Survey, 1993.

In the dataset `polbirth` involving data on political orientation and opinion on teenage birth control of a sample of 911 U.S. citizens, var. 1 is `Politics` with categories: `Extremely liberal`, `Liberal`, `Slightly liberal`, `Moderate`, `Slightly conservative`, `Conservative`, `Extremely conservative` and var. 2 is `Birth` with `Strongly agree`, `Agree`, `Disagree`, `Strongly disagree` categories.

With these variables, for example, we can test the hypothesis that the distributions of `Politics`, given the levels of `Birth`, are ordered according to the simple dominance criterion coherently with the strength of the opinion on `Birth` control. This hypothesis is equivalent to require that all the global-local log-odds ratios are non-negative. Continuation-local or local log-odds ratios can be constrained to consider successively stronger notions of monotone dependence (uniform and likelihood ratio stochastic orderings), see Dardanoni and Forcina (1998) and Shaked and Shanthikumar (1994).

Let us test the simple monotone dependence of `Politics` on `Birth`.

The marginal sets, the logit types and the labels of the variables are declared below

```
R> data("polbirth", package = "hmmm")
R> y <- getnames(polbirth)
R> marginals <- marg.list(c("g-marg", "marg-l", "g-l"))
R> names <- c("Politics", "Birth")
```

The marginal set `marg` where the interactions are defined, the interactions `int` subject to inequality constraints, and the `types` of logit used for each variable are listed as follows, so that the log-odds ratios of global-local types are the interactions to be constrained

```
R> ineq <- list(marg = c(1, 2), int = list(c(1, 2)), types = c("g", "l"))
```

The marginal model with inequalities on global-local interactions is defined using the function `hmmm.model()` where `ineq` is assigned to the argument `dismarg`

```
R> model <- hmmm.model(marg = marginals, dismarg = ineq, lev = c(7, 4),
+  names = names)
```

More than one list, like that specified in `ineq`, can compose `dismarg` if interactions defined in different marginal distributions have to be constrained (see details in the help of the `hmmm.model()` function).

The model with non-negative global-local log-odds ratios (simple monotone dependence model) is estimated with the function `hmmm.mlfit()` where the argument `noineq` is declared `FALSE`

```
R> mlr <- hmmm.mlfit(y, model, noineq = FALSE)
```

Note that if `noineq = TRUE` (the default) inequality constraints are ignored. The model estimated without any inequality constraints on parameters is, in this case, the saturated model

```
R> msat <- hmmm.mlfit(y, model)
```

If the inequality constraints are turned into equality, all the global-local log-odds ratios are null and the corresponding model is the stochastic independence model

```
R> model0 <- hmmm.model(marg = marginals, lev = c(7, 4), sel = c(10:27),
+  names = names)
R> mnull <- hmmm.mlfit(y, model0)
```

The fitted models are compared through the function `hmmm.chibar()`. The arguments of `hmmm.chibar()` are the estimated models with inequality constraints turned into equalities (`nullfit`), with inequality constraints (`disfit`) and without inequality constraints on parameters (`satfit`)

```
R> test <- hmmm.chibar(nullfit = mnull, disfit = mlr, satfit = msat)
```

Function `hmmm.chibar()` can be only used to test problems of type A and B, Silvapulle and Sen (2005, p. 61): the test of type A compares the H_0 : `nullfit` model against the H_1 : `disfit` model; while the type B problem means testing H_0 : `disfit` model against H_1 : `satfit` model. The main difference between type A and type B problems is that inequalities are present in the alternative hypothesis of type A and in the null hypothesis of type B

problems. To compare nested models without inequality constraints the function `anova()`, introduced in Section 3, has to be used.

The null distribution of the likelihood ratio statistic for or against inequality constraints turns out to be chi-bar-square, that is a mixture of chi-square distributions. Its tail probabilities are computed by simulation, the method *Simulation 2* described in Silvapulle and Sen (2005, p. 79) is implemented in `hmmm.chibar()` as default. Alternatively, if the number of inequality constraints is not large, (<= 15), the tail probabilities can be exactly computed by the Kudo's method, (Silvapulle and Sen 2005, p. 83), by setting the argument `kudo` of `hmmm.chibar()` equal to `TRUE`.

The output of `hmmm.chibar()` provides the values of the likelihood ratio statistics and their p values for both tests of type A and B

```
R> test

 CHIBAR P VALUES

           test       pvalue
testA 64.457490 1.629837e-09
testB  2.033941 9.657715e-01
```

In this case, `testA` rejects the `nullfit` model in favour of `disfit` model, and `testB` does not reject `disfit` model versus the saturated model. Therefore, the model under inequalities seems to suit the data. A more detailed output is printed by `summary`.

9. MPH models under inequality restrictions

The **hmmm** package can handle Lang *multinomial Poisson homogeneous models* subject to inequality constraints (see Cazzaro and Colombi 2009) through the function `mphineq.fit`.

The MPH models are characterized by an independent sampling plan and a system of equality and/or inequality homogeneous constraints on the vector of expected table counts. The sampling scheme can be a product of Poisson and/or multinomial random variables.

To give an example, we refer to Lang (2004) that investigated citation patterns in three journals of statistics and probability (Journal of the American Statistical Association JASA, Biometrics BMCS, The Annals of Statistics ANNS). Let (i, j) be a cross-citation, where i is the journal of the citing article (1=JASA, 2=BMCS, 3=ANNS) and j is the journal of the cited article (1=JASA, 2=BMCS, 3=ANNS). In particular, we consider the observed counts of Table 2 - "1999 statistics journals citation pattern counts," Lang (2004, p. 349)

```
R> y <- matrix(c(104, 24, 65, 76, 146, 30, 50, 9, 166), 9, 1)
```

Note that the frequency of cross-citation (i, j) of Lang's table enters the vector `y` in position $3(i-1)+j$. The sampling scheme can be described by two matrices, `Z` and `ZF`. The number of strata are established by the number of columns of the population matrix `Z` that is a zero-one matrix having as many rows as the number of counts. A `1` in row c and column s means that the cth count comes from the sth stratum. The columns of `ZF`, the sample constraints matrix,

are the subset of the columns in `Z` corresponding to the multinomial strata with fixed sample size (see details in the help of the `mphineq.fit()` function).

With respect to the previous example, the following statements

```
R> Zmat <- kronecker(diag(3), matrix(1, 3, 1))
R> ZFmat <- kronecker(diag(3), matrix(1, 3, 1))[,3]
```

mean that the population matrix `Z` involves 3 strata with 3 observations each and that the third stratum sample size is considered as fixed.

Lang makes inference on some functions of the m_{ij}s, the expected counts of cross-citations. He considers the Gini concentrations of citations for each of the journals: $G_i = \sum_{j=1}^{3}(m_{ij}/m_{i+})^2$, $i = 1, 2, 3$. The hypothesis tested by Lang considers equal Gini concentrations in the strata, $G_1 - G_2 = 0$ and $G_3 - G_1 = 0$. The following statement builds the Gini indices for each of the journals and calculates the constraints on them

```
R> Gini <- function(m) {
+  A<-matrix(m,3,3,byrow=TRUE)
+   GNum<-rowSums(A^2)
+   GDen<-rowSums(A)^2
+   G<-GNum/GDen
+   c(G[1], G[3]) - c(G[2], G[1])
+   }
```

The model can then be estimated by the command `mphineq.fit()` assigning the function `Gini()` to the argument `h.fct`. Note that the command `mphineq.fit()` includes several arguments: the observed counts (`y`), the population matrix (`Z`), the sample constraints matrix (`ZF`), the function for the equality (`h.fct`) or inequality (`d.fct`) constraints, among others

```
R> mod_eq <- mphineq.fit(y, Z = Zmat, ZF = ZFmat, h.fct = Gini)
```

One hypothesis of interest, that implies the one proposed (but not tested) by Lang himself, is to consider that there is more concentration in ANNS than in the other two journals and more concentration in JASA than in BMCS. This is an example of MPH model subject to inequality constraints: $G3 > G1 > G2$ or equivalently $G_1 - G_2 > 0, G_3 - G_1 > 0$. The model is again defined through the function `mphineq.fit()` where the function `Gini()` is now assigned to the argument `d.fct`

```
R> mod_ineq <- mphineq.fit(y, Z = Zmat, ZF = ZFmat, d.fct = Gini)
```

The fitted models can be, finally, compared through the function `hmmm.chibar()`, already illustrated in Section 8. In this context, it is worthwhile to note that the reference model without inequality constraints corresponds to the saturated model

```
R> mod_sat <- mphineq.fit(y, Z = Zmat, ZF = ZFmat)
R> hmmm.chibar(nullfit = mod_eq, disfit = mod_ineq, satfit = mod_sat)
```

```
 CHIBAR P VALUES

            test       pvalue
testA 24.1438779 1.583694e-06
testB  0.4896523 4.782429e-01
```

Evidently, the model of common Gini concentrations (`mod_eq`) is untenable as we deduced from `testA`, coherently with Lang results. The outcome of `testB` is in favour of `mod_ineq`, it then appears that there is more concentration in ANNS than in the other two journals and more concentration in JASA than in BMCS.

10. Discussion

The main contribution of the R package **hmmm** is to give user-friendly tools to define and fit complete hierarchical multinomial marginal models under equality and inequality constraints on a wide variety of marginal interactions. Classical and more recent marginal models, proposed in the categorical data literature, are special cases of the HMM models. Thus, the potential applicability of the package is wide as several contributions on marginal models are implemented.

In this paper, only the main features of this package have been illustrated, whereas other aspects have not been here analyzed.

First, such a package permits to estimate non-hierarchical or non-complete marginal models. In particular, a marginal model is non-hierarchical when an interaction is not defined in the first marginal distribution containing it and non-complete when an interaction is defined in more than one distribution. Most of these models are not smooth, so the standard MLE asymptotic theory does not apply. Anyway, under some conditions they are smooth and therefore they can be fitted by **hmmm**. Forcina (2012) shows examples of smooth non-hierarchical marginal models. In the case of non-hierarchical non-complete smooth marginal models for every marginal distribution, the list of interactions must be specified by the syntax used for inequalities.

In addition to this, the package can fit hidden Markov models where the conditional distribution of several observable variables and the transition probabilities of the latent chain can be specified by HMM models, see Colombi and Giordano (2011). The `hidden.emfit()` function computes the ML estimates of the parameters via an EM algorithm, but the current version of the package does not provide standard errors.

Moreover, consider that the package is designed to deal with multiway tables and cannot handle individual data, commonly used in presence of non-categorical covariates.

Finally, we are aware that some HMM models are Markov with respect to chain or mixed graphs that can be easily defined in R (see for example the R package **ggm** by Marchetti, Drton, and Sadeghi 2012). Therefore, it could be an useful improvement to enable the package to define a HMM model starting from a graphical representation.

Updated versions of the package will be oriented to overcome the mentioned limits.

We highlighted that in the R environment, other authors dealt with contents connected with our package but in a more restrictive purpose: the R package **cmm** by Bergsma and van der Ark

(2013), accompanying the book by Bergsma *et al.* (2009), and the Lang's `mph.fit` function handling multinomial Poisson homogeneous models (available from the author).

References

Agresti A (2013). *Categorical Data Analysis - 3rd Edition*. John Wiley & Sons, New Jersey.

Bartolucci F, Colombi R, Forcina A (2007). "An Extended Class of Marginal Link Functions for Modelling Contingency Tables by Equality and Inequality Constraints." *Statistica Sinica*, (17), 691–711.

Bergsma W, Croon M, Hagenaars J (2009). *Marginal Models for Dependent, Clustered, and Longitudinal Categorical Data*. Springer-Verlag, New York.

Bergsma W, van der Ark A (2013). ***cmm**: Categorical Marginal Models*. R package version 0.7.

Bergsma WP, Rudas T (2002). "Marginal Models for Categorical Data." *The Annals of Statistics*, (30), 140–159.

Cazzaro M, Colombi R (2009). "Multinomial-Poisson Models Subject to Inequality Constraints." *Statistical Modelling*, (9(3)), 215–233.

Cazzaro M, Colombi R (2013). "Marginal Nested Interactions for Contingency Tables." *Communications in Statistics - Theory and Methods*. To appear.

Colombi R, Giordano S (2011). "Lumpability for Discrete Hidden Markov Models." *Advances in Statistical Analysis*, (95), 293–311.

Dardanoni V, Forcina A (1998). "A Unified Approach to Likelihood Inference on Stochastic Orderings in a Nonparametric Context." *Journal of the American Statistical Association*, (93), 1112–1123.

Douglas R, Fienberg S, Lee M, Sampson A, Whitaker L (1990). *Positive Dependence Concepts for Ordinal Contingency Tables. In: Topics in Statistical Dependence*. 16, p. 189-202. Block, H.W., Sampson, A.R., Savits, T.H. (Eds.), Hayward: Institute of Mathematical Statistics.

Forcina A (2012). "Smoothness of Conditional Independence Models for Discrete Data." *Journal of Multivariate Analysis*, (106), 49–56.

Glonek GFV, McCullagh P (1995). "Multivariate Logistic Models." *Journal of the Royal Statistical Society B*, (57), 533–546.

Koch GG, Landis JR, Freeman JL, Freeman DH, Lehnen RG (1977). "A General Metodology for the Analysis of Experiments with Repeated Measurement of Categorical Data." *Biometrics*, (38), 563–595.

Lang JB (2004). "Multinomial-Poisson Homogeneous Models for Contingency Tables." *The Annals of Statistics*, (32), 340–383.

Madsen M (1976). "Statistical Analysis of Multiple Contingency Tables. Two Examples." *Scandinavian Journal of Statistics*, (3), 97–106.

Marchetti GM, Drton M, Sadeghi K (2012). ***ggm**: A Package for Graphical Markov Models*. R package version 1.995-3, URL http://CRAN.R-project.org/package=ggm.

Marchetti GM, Lupparelli M (2011). "Chain Graph Models of Multivariate Regression Type for Categorical Data." *Bernoulli*, (17), 827–844.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Shaked M, Shanthikumar JG (1994). *Stochastic Orders and their Applications*. Academic, San Diego.

Silvapulle MJ, Sen PK (2005). *Constrained Statistical Inference*. John Wiley & Sons, New Jersey.

Affiliation:

Manuela Cazzaro
Dipartimento di Statistica e Metodi Quantitativi
Università di Milano-Bicocca
20126 Milano, Italia
E-mail: manuela.cazzaro@unimib.it

Roberto Colombi
Dipartimento di Ingegneria
Università di Bergamo
24044 Dalmine (Bergamo), Italia
E-mail: roberto.colombi@unibg.it

Sabrina Giordano
Dipartimento di Scienze Economiche, Statistiche e Finanziarie
Università della Calabria
87036 Arcavacata di Rende (Cosenza), Italia
E-mail: sabrina.giordano@unical.it