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 2012, Ch 12). A complete hierarchical multinomial marginal model (HMM) 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). In particular, in HMM models every interaction is defined in one and only one marginal distribution (completeness) and within the first marginal set which contains it (hierarchy). 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.

In this tutorial we show how to define and estimate HMM models with interactions restricted under equality and inequality constraints or influenced by the effects of covariates using the **hmmm** package.

2. 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 the **hmmm** package, the input data must be a vector of cell frequencies arranged in antilexicographic order. To start with, we show how to get a vector of labeled frequencies from the `accident` data frame, regarding accidents occurred to workers of a northern Italian city in 1998. The data are provided by INAIL, the Italian institute for insurance against factory accidents. The data frame columns contain the variables: var. 1 `Type` of the injury (with 3 levels), var. 2 `Time` to recover (with 4 levels), var. 3 `Age` of the worker (with 3 levels) and var. 4 solar Hour (with 2 levels) along with the counts for each configuration of the variables (last column). Note that variables are denoted by integers, the lower the number identifying the variable, the faster its category subscript changes in the vectorized contingency table.

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

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 last twelve rows are printed to give an example.

```
      cell names                         counts
 [1,] uncertain 0 |-- 7 > 45 afternoon   39
 [2,] avoidable 0 |-- 7 > 45 afternoon   23
 [3,] not-avoid 0 |-- 7 > 45 afternoon   1
 [4,] uncertain 7 |-- 21 > 45 afternoon  29
 [5,] avoidable 7 |-- 21 > 45 afternoon  14
 [6,] not-avoid 7 |-- 21 > 45 afternoon  2
 [7,] uncertain 21 |-- 60 > 45 afternoon 17
 [8,] avoidable 21 |-- 60 > 45 afternoon 12
 [9,] not-avoid 21 |-- 60 > 45 afternoon 8
[10,] uncertain >= 60 > 45 afternoon     6
[11,] avoidable >= 60 > 45 afternoon     10
[12,] not-avoid >= 60 > 45 afternoon     16
```

Let us start by defining a saturated HMM model, i.e., a model without any restrictions on the interactions. First, the list of the marginal sets has to be declared and the command marg.list() serves the need. Here, with respect to the accident data, it defines 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. The symbol b states that 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). The second statement uses the function hmmm.model() to define the HMM model. In the input arguments, as well as the marginal sets, information on the number of categories and on the names of the variables involved is specified. The output illustrates how the interactions are allocated according to the principles of hierarchy and completeness.

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

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 last two columns 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 in the `accident` data. 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()`

```
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 )
```

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)
```

When the constrained interactions are log-linear parameters defined in the joint distribution (Agresti 2012), 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 the distribution of `Type` and `Time` do not depend on the levels of `Age` and `Hour`, we must set to zero the interactions of the second and third 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
```

Note that the previous `modelA` is not log-linear because some constrained interactions are defined in marginal distributions. On the contrary, the previous model without constraints on the marginal interactions is log-linear and can be defined and estimated by the following statements

```
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)
```

3. Generalized marginal interactions

In the previous section all the interactions defined within the marginal distributions are of log-linear 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, or 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 inequality constraints on these generalized interactions (in Section 5 the usefulness of these interactions for testing hypotheses of stochastic orderings is clarified). Bartolucci *et al.* (2007), in particular, showed that the generalized marginal interactions within a marginal set can be defined by assigning a logit type to each variable of the marginal distribution. For example, if we use global logits for both variables in a bivariate distribution, then this distribution is parameterized by global logits and global log-odds ratios. If we assign continuation logits to one variable and global logits to the other one, then we have a parameterization in terms of continuation logits, global logits and continuation-global log-odds ratios.

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. The types of logit allowed in **hmmm** are baseline `b`, local `l`, global `g`, continuation `c` and reverse continuation `rc`. A different type of logit is discussed in the next section.

For example, we consider the `madsen` data (Madsen 1976) with variables: `Influence` (var. 1 with 3 ordinal levels), `Satisfaction` (var. 2 with 3 ordinal levels), `Contact` (var. 3 with 2 levels), `Housing` (var. 4 with 4 levels).

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

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

This means that in the bivariate distribution of variables 3, 4 all the interactions are of local type, while in the joint distribution of 1, 3, 4 the interactions 1 are global logits, the interactions 13 and 14 are global-local 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.

To test if there is an additive effect of variables 3 and 4 on the global logits of variables 1 and 2 in the marginal distributions 134 and 234, we can run the following statements

```
R> modelad1 <- hmmm.model(marg = margin, lev = c(3, 3, 2, 4),
+  names = c("In", "Sa", "Co", "Ho"), sel = c(18:23, 34:39))
R> data("madsen", package = "hmmm")
R> y <- getnames(madsen, st = 6)
R> modadd1 <- hmmm.mlfit(y, modelad1)
R> modadd1
```

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

Moreover, to add the hypothesis that the global odds ratios of the variables 1 and 2 do not depend on the levels of the other two variables, the ggl and ggll interactions, which occupy the positions 44 : 71 in the vector of parameters, must be constrained to zero

```
R> modelad2 <- 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> modadd2 <- hmmm.mlfit(y, modelad2)
R> modadd2

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

For an alternative way of specifying other similar hypotheses see Section 6 where the effect of covariates on interactions is taken into account.

4. Recursive marginal interactions

Cazzaro and Colombi (2013) extended the class of generalized marginal interactions by introducing a new type of logit: the recursive (or nested) logit. In the simplest case, these logits are defined in correspondence of a partition of the categories of a variable. As an example we consider the `relpol` data, Bergsma, Croon, and Hagenaars (2009, p. 24), 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} and for `Politics` the partition in the sets L={EL, LI, SL}, M={MO} and C={SC, CO, EC}.

A first set of logits contains the baseline logits which are defined within every set of the partition (the reference category can be chosen arbitrarily in every set). For example, this kind of recursive logits for `Religion` and `Politics` are: $log[pr(\text{CA})/pr(\text{PR})]$ and $log[pr(\text{EL})/pr(\text{LI})]$, $log[pr(\text{SL})/pr(\text{LI})]$, $log[p(\text{SC})/p(\text{CO})]$, $log[pr(\text{EC})/pr(\text{CO})]$, respectively. A second set includes the baseline logits defined on the probabilities of the sets of the partition (the reference set can be chosen arbitrarily). Considering the `relpol` data, the recursive logits of the variables `Religion` and `Politics` in this case are: $log[pr(N)/pr(R)]$ and $log[pr(C)/pr(L)]$, $log[pr(M)/pr(L)]$, respectively.

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`.

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. The rows of this matrix specify the categories whose probabilities appear in the numerator and denominator of every recursive logit. In a row a value `1` (`-1`) corresponds to the categories whose probability is cumulated in the numerator (denominator), `0` if the category is not involved. Finally the output of `recursive()` must be assigned to the argument `cocacontr` of `hmmm.model()`.

With reference to the `relpol` data the necessary statements are

```
R> marginals <- marg.list(c("r-marg", "marg-r", "r-r"))
R> R1 <- matrix(c(-1, -1,  1,
+                  -1,  1,  0), 2, 3, byrow = TRUE)
R> R2<-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)
R> rec <- recursive(R1, R2)
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
```

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: $log[pr(\mathrm{EL})/pr(\mathrm{LI})]{=}log[pr(\mathrm{EC})/pr(\mathrm{CO})]$ and $log[pr(\mathrm{SL})/pr(\mathrm{LI})]{=}log[p(\mathrm{SC})/p(\mathrm{CO})]$ stating that the distribution between extreme and moderate attitudes is the same within conservatives and liberals. The first condition equates the third and sixth recursive logit of `Politics` that occupy positions 5 and 8 in the vector of parameters, respectively. The second condition equates the fourth and fifth recursive logit that are in positions 6 and 7 in the vector of parameters. The hypotheses can be tested by assigning the following constraints matrix `Emat` to the argument `E` of the function `hmmm.model()`

```
R> Emat <- cbind(matrix(0, 2, 4), matrix(c(0, 1, 1, 0, -1, 0, 0, -1), 2, 4),
+  matrix(0, 2, 12))
R> modelE <- hmmm.model(marg = marginals, lev = c(3, 7),
+  names = c("Rel", "Pol"), cocacontr = rec, E = Emat)
R> data("relpol", package = "hmmm")
R> y <- getnames(relpol, st = 4)
R> modE <- hmmm.mlfit(y, modelE)
R> print(modE, printflag = TRUE)

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

```
       inter.  marg.   type STRATA_1
link1  Rel     Rel     r    2.188366
link2  Rel     Rel     r    1.080277
link3  Pol     Pol     r    -0.368723
link4  Pol     Pol     r    -0.404021
link5  Pol     Pol     r    1.708872
link6  Pol     Pol     r    -0.13456
link7  Pol     Pol     r    -0.13456
link8  Pol     Pol     r    1.708872
link9  Rel.Pol Rel,Pol rr   -1.670636
link10 Rel.Pol Rel,Pol rr   -0.256603
link11 Rel.Pol Rel,Pol rr   -0.980403
link12 Rel.Pol Rel,Pol rr   0.26737
link13 Rel.Pol Rel,Pol rr   -0.065654
link14 Rel.Pol Rel,Pol rr   -0.85745
link15 Rel.Pol Rel,Pol rr   -0.567533
link16 Rel.Pol Rel,Pol rr   -0.386656
link17 Rel.Pol Rel,Pol rr   0.797325
link18 Rel.Pol Rel,Pol rr   0.348307
link19 Rel.Pol Rel,Pol rr   1.071584
link20 Rel.Pol Rel,Pol rr   -0.95906
```

5. 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).

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

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 interactions subject to inequality constraints, the marginal set where they are defined 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 input `noineq` is declared `FALSE`

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

If the previous 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 model estimated without any inequality constraints on parameters is, in this case, the saturated model

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

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()` tests problems of type A and B, Silvapulle and Sen (2005, p. 61): the test of type A compares the `nullfit` model under H_0 against the `disfit` model under H_1; 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.

The null distribution of the likelihood ratio statistic G^2 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.

The output of `hmmm.chibar()` provides the values of the likelihood ratio statistics G^2 and their simulated p values for both tests of type A and B.

```
R> test

 chibar simulated pvalues

           test       pvalue
testA 64.457490 3.895161e-09
testB  2.033941 9.344106e-01
```

A much detailed output is printed by `summary`.

6. 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).

We consider the `accident` data, 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 and var. 3 `Age` of the worker (3 levels) and var. 4 solar `Hour` (2 levels) as covariates. Remind that the lower the variable number, the faster the variable sub-script 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 subscripts of the response variables change faster than the subscripts 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()`). With respect to the vector of counts from the `accident` data the necessary statement is

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

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 components, each for every interaction specified above, defining the effects of the covariates on such interactions, 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, sep = ";")
R> mod1 <- hmmm.mlfit(y, model, y.eps = 0.1)
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)
```

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

7. Further remarks

The complete hierarchical marginal models, that can be analyzed with the **hmmm** package, 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 (Section 2). The Bergsma and Rudas (2002) *marginal models* are HMM models where the interactions of log-linear type are defined in different marginal distributions (Section 2). The models described in the other sections are extensions of the Bergsma-Rudas models involving more general type of interactions. Finally, Glonek and McCullagh (1995) *multivariate logistic models* (see the examples of Section 6) 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.

Note that the **hmmm** package can estimate the parameters of all the previous models under inequality constraints.

Furthermore, that are other topics that this tutorial do not cover: (i) *hidden Markov models* where the conditional distribution of several observed variables and the transition probabilities of the latent chain can be specified by HMM models, see Colombi and Giordano (2011); (ii) Lang (2004) *multinomial Poisson homogeneous models* that can be estimated with the **hmmm** package also under inequality constraints.

References

Agresti A (2012). *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 WP, Rudas T (2002). "Marginal Models for Categorical Data." *The Annals of Statistics*, (30), 140–159.

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.

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

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

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