# $\ell_{EM}$: A general program for the analysis of categorical data[1]

Jeroen K. Vermunt

Department of Methodology and Statistics,
Tilburg University
The Netherlands

September 25, 1997

[1]When you report results obtained with $\ell_{EM}$, you should refer to this manual as "Vermunt, J.K. (1997). LEM: A General Program for the Analysis of Categorical Data. Department of Methodology and Statistics, Tilburg University" and/or to my published Ph.D. dissertation "Vermunt, J.K. (1997). Log-linear Models for Event Histories. Thousand Oakes: Sage Publications", which describes the models and algorithms implemented in $\ell_{EM}$.

# Contents

# Chapter 1

# Introduction

The $\ell_{EM}$[1] program is a general system for the analysis of nominal, ordinal, and interval level categorical data. It can be used to obtain parameter estimates for

- log-linear models
- log-multiplicative association models,
- correspondence analysis,
- regression models for categorical dependent variables,
- path models for categorical endogenous variables,
- latent structure models for categorical items,
- lisrel-like models for categorical endogenous variables,
- models for nonresponse in categorical variables,
- log-rate and discrete-time logit models for analyzing event history data.

It should be noted that the purpose of this manual is not to explain in detail all these models, but to demonstrate the use of the $\ell_{EM}$ program by means of many examples of input files. The various models and the technical details on their estimation are described in Vermunt (1996b, 1997) and the references cited below.

Below, first some additional general information will be given (sections 1.1 and 1.2). The different types of models that can be estimated with $\ell_{EM}$ are presented in the next seven chapters: models for cell frequencies (chapter 2), regression models (chapter 3), path models (chapter 4), latent class models (chapter 5), path models with latent variables (chapter 6), models with partially missing data (chapter 7), and event history models (chapter 8). Chapter 9 deals with settings which can be changed by the user to influence the working of the program. Chapter 10 describes the components which appear in the output file. And finally, chapter 11 presents the complete $\ell_{EM}$ command syntax.

## 1.1 Running the program

The DOS protected mode version of $\ell_{EM}$ is compiled with Borland Pascal 7.0. It runs on a DOS computer with a mathematical co-processor. Since it uses extended memory, the size of the problems that can be dealt with depends mainly on the amount of internal memory of the computer that is used. The program is started at the DOS prompt by typing:

[1]The name $\ell_{EM}$ stands for 'log-linear and event history analysis with missing data using the EM algorithm'.

```
LEM <name of the input file> <name of the output file>.
```

If the input and output files are not specified, the program will ask to supply their names.

The Windows 95 version of $\ell_{EM}$ is written in Delphi Pascal 2.0. In this version, which is started by running the executable file LEMWIN.EXE, the user has three text windows at his disposal: Input, Output, and Log. A model has to be specified in and runned from the Input window. The resulting output and log files are automatically loaded when activating the Output and Log window, respectively.

Most of the items in the File, Edit, and Window submenus of the Windows version of $\ell_{EM}$ are standard in Windows 95 applications. Here, we describe only the use of the menu items which are specific for $\ell_{EM}$.

The Run item in the File submenu, as well as the Examples submenu are only visible when the Input window is active. A model that is specified in the Input window is runned by clicking on Run. The Examples submenu can be used to load one of the example input files in the Input window. There are more than 200 examples covering the hole range of models that can be estimated with $\ell_{EM}$.

The GoTo submenu, which is only visible when the Output window is active, can be used to jump to the begin of one of sections of the output file.

The Chi-squared option from the Tools submenu can be used to compute Chi-squared probabilities. The Preferences option in the same submenu makes it possible to change some settings. More precisely, clicking on Font yields the standard Windows dialog for changing the font type and size. The Show Log item makes it possible to turn on/off the appearence of a log window when $\ell_{EM}$ is running. The Confirm item switches off/on the confirmation requests appearing if the content of the Input or Output window is not saved. By clicking on Reset, the default Font, Show Log, and Confirm settings are restored. Finally, the default window sizes can be restored by means of Restore in the Window submenu.

Thus, both in the DOS and the Windows version, one has to prepare an input file using the $\ell_{EM}$ command syntax. The $\ell_{EM}$ syntax consists of commands which have to be typed in lower case and of which only the first three characters are significant. The input file is read in free format, with spaces or commas as separation characters. Comments can be put in the input file using asterisks. When a '`*`' is encountered, the rest of the line is considered to be comment, and therefore skipped.

## 1.2 Some technical details on parameter estimation

Parameter estimation is performed by means of maximum likelihood. Several procedures are implemented for obtaining maximum likelihood estimates of the model parameters. The iterative proportional fitting (IPF) algorithm is used for estimating simple hierarchical log-linear models (Bishop, Fienberg, and Holland, 1975). Non-hierarchical log-linear models, log-multiplicative models, regression models based on cumulative link functions, and models with the more complicated type of equality and fixed-value restrictions on conditional probabilities are estimated with the uni-dimensional Newton algorithm (Goodman, 1979; Vermunt, 1996b, 1997). The same kind of algorithm is used to estimate models with non-parametric ordinal constraints (see Croon, 1990) and with linear restrictions on cell frequencies (see Haber and Brown, 1986; and Bergsma, 1997).

ML estimates of the parameters of models containing latent variables or partially missing data are computed by means of the Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin, 1979; Vermunt 1996b, 1997), where the above-mentioned methods are applied in the M step. Although this EM algorithm can sometimes be a bit slow, it is extraordinary stable.[2]

[2]Actually, $\ell_{EM}$ uses a variant of the EM algorithm which combines features of the GEM (Little and Rubin,

For most models, it is also possible to use one of four other algorithms for parameter estimation, namely, Newton-Raphson, Steepest-descent, Broyden-Fletcher-Goldfarb-Shanno (BFGS), or Levenberg-Maquardt (see Press et all, 1986). Both the information matrix used in Newton-Raphson and Levenberg-Maquardt and the gradient vector used in all four methods are approximated numerically. It should be noted that the information matrix is also used for computing asymptotic standard errors and checking identification.

1987) and the ECM (Meng and Rubin, 1993) algorithms. GEM involves improving the expected complete data likelihood in the M step rather than maximizing it, while ECM implies that a conditional maximization procedure, such as IPF or uni-dimensional Newton, is used in the M step rather than a multidimensional maximization procedure (see Vermunt, 1996b, 1997).

# Chapter 2

# Models for cell frequencies

The main topic of this chapter is the specification of log-linear models for multi-way frequency tables. But also some other types of models for cell frequencies are presented, such as log-multiplicative association models, log-rate models, models with linear restrictions on cell frequencies, and correspondence analysis.[1]

## 2.1 Hierarchical log-linear models

Suppose we want to specify a log-linear model for a three-way frequency table formed by the variables $A$, $B$, and $C$. A saturated log-linear model for this table is given by

$$\log m_{abc} \quad = \quad u + u_a^A + u_b^B + u_c^C + u_{ab}^{AB} + u_{ac}^{AC} + u_{bc}^{BC} + u_{abc}^{ABC} \,, \tag{2.1}$$

where $m_{abc}$ denotes an expected cell count and the $u$ terms log-linear parameters.

A subset of the family of log-linear models are hierarchical log-linear models. When a particular interaction effect is included in a hierarchical log-linear model, all lower-order effects containing a subset of the indices of the effect concerned must be included in the model as well. An attractive feature of hierarchical log-linear models is that they can be estimated with the simple Iterative Proportional Fitting (IPF) algorithm, which fits the set of margins which have to be reproduced according to the specified model.

In $\ell_{EM}$, the specification of a hierarchical log-linear model will at least consist of the following four pieces of information:

- the number of variables,
- the dimensions or number of categories of the variables,
- the model to be estimated,
- the table with observed frequencies.

A $\ell_{EM}$ input file specifying an independence model for a three-way frequency table formed by the variables `A`, `B`, and `C` can be specified as follows:

```
* example 2.1a: hierarchical log-linear model
man 3
dim 2 2 3
mod {A,B,C}
dat [31  77 35
     68  60 65
```

[1] Some textbooks which describe extensively the models discussed in this chapter are Bishop, Fienberg, and Holland (1975), Goodman (1978), Haberman (1978, 1979), Knoke and Burk (1980), Agresti (1990), Hagenaars (1990), and Vermunt (1996b, 1997).

```
44 147 61
68  50 44]
```

The command `man` indicates the number of manifest variables.[2] With `dim` one specifies the dimensions of the table to be analyzed. It should be noted that when reading the frequencies the levels of the last variable in the list is assumed to change first. So, here `C` (which has three categories) changes first, while `A` changes last.

The hierarchical log-linear model of interest is specified with the command `mod` (model), where the fitted margins must be given between curly braces or parentheses.

One way of specifying the data to be analyzed is – as is done in the above example – to type the frequency table between square brackets after the command `dat`. Another option is to specify (again after the command `dat`) the name of the file from which the frequencies have to be read. As is shown in section 2.7, it is also possible to use data in the form of individual records rather than in the form of a frequency table.[3]

In the next example, we will specify a no-three-variable interaction model for the same frequency table. Now we will, however, use our own variable labels, put some comments in the input file, and read the observed frequency table from a file:

```
* example 2.1b: hierarchical log-linear model
* A = age; R = religious membership; P = Political preference
* no-three-variable interaction model
man 3             * number of (manifest) variables
dim 2 2 3         * levels of the variables
lab A R P         * variable labels
mod {AR,AP,RP}    * log-linear model
dat ex21.fre      * file containing frequency table
```

As can be seen, the command `lab` is used to specify the variable labels. The default variable labels for manifest variables are `A`, `B`, `C`, `D`, etc.. It is recommended to use upper case letters for the variable labels to prevent that they are confounded with the $\ell_{EM}$ commands, which have to be typed in lower case.[4]

It will be clear that by replacing the model specification in example 2.1b one can easily specify any other type of hierarchical log-linear model for frequency table `ARP`. For instance,

```
mod {ARP}
```

would yield a saturated model and

```
mod {AR,RP}
```

a model in which `A` and `P` are conditionally independent of one another given a person's score on `R`.

The default coding scheme which is used to identify the parameters of hierarchical log-linear models is effect coding. Dummy coding can requested with the command `dum`. For instance, adding the line

```
dum 2 2 3
```

to the above input file will yield dummy coded parameters, in which the last categories of `A`, `R`, and `P` are used as reference categories.

[2] We use the term manifest variables to be able to distinguish them from latent variables (`lat`, chapter 5), response indicators (`res`, chapter 7), continuous exogenous variables (`con`, chapter 3), and the time and the risk variable (`tim` and `ris`, chapter 8).

[3] In some situations it is even necessary to use individual records, i.e., in models with continuous exogenous variables and in most types of event history models.

[4] It is also possible to use labels which are longer that one character (up to three characters). In that case, one must separate the variables by means of a '.'.

## 2.2 Non-hierarchical log-linear models

Log-linear models can be defined in a much more general way by viewing them as a special case of the family of generalized linear models (McCullagh and Nelder, 1989). In its most general form, a log-linear model can be defined as

$$\log m_i \quad = \quad \sum_j \beta_j x_{ij} \,, \tag{2.2}$$

where $i$ denotes a cell entry, $\beta_j$ is a particular $u$ term, and $x_{ij}$ is an element of the design matrix.

This much more general formulation makes it possible to specify all kinds of non-hierarchical models. Usually, the design matrix is used to impose one of the following three types of restrictions on the log-linear parameters: fixing parameters to zero in a non-hierarchical way, making parameters equal to one another, or specifying parameters to be in a fixed ratio to one another (Haberman, 1978; Agresti, 1990; Rindskopf, 1990; Vermunt, 1996b, 1997).

In $\ell_{EM}$, these non-hierarchical log-linear models can be specified by means of the options for user-defined designs. In addition, there are some predefined designs for the most common situations, namely, for simple log-linear effects, uniform associations, row and columns associations, symmetric associations, diagonal and off-diagonal parameters, total score parameters, difference parameters, and parameters of ranking models.

### 2.2.1 Using design matrices

The $\ell_{EM}$ program has two commands for specifying design matrices: `cov(..)` and `fac(..)`. The functioning of `cov(..)` resembles the interval level covariates in SPSS, while `fac(..)` resembles the nominal factors in GLIM. The complete syntax of these two command is:

```
cov(<margins>,<# of effects>,<group margin>,<a/b/c>,<# of groups>)
fac(<margins>,<# of effects>,<group margin>,<a/b/c>,<# of groups>)
```

The use of these commands can be illustrated with the same data example as was used in the section on hierarchical log-linear models (2.1). Suppose we want to specify hierarchical log-linear model `{AR,RP}` for table `ARP`. Using the command `cov(..)`, this can be accomplished as follows:

```
* example 2.2a: use of cov(..)
* hierarchical model {AR,RP}
man 3
dim 2 2 3
lab A R P
mod {cov(ARP,7)}
dat ex22.fre
des [ 1  1  1  1  1  1 -1 -1 -1 -1 -1 -1   * A 1
      1  1  1 -1 -1 -1  1  1  1 -1 -1 -1   * R 1
      1  0 -1  1  0 -1  1  0 -1  1  0 -1   * P 1
      0  1 -1  0  1 -1  0  1 -1  0  1 -1   * P 2
      1  1  1 -1 -1 -1 -1 -1 -1  1  1  1   * AR 11
      1  0 -1 -1  0  1  1  0 -1 -1  0  1   * RP 11
      0  1 -1  0 -1  1  0  1 -1  0 -1  1]  * RP 12
```

As can be seen, `cov(ARP,7)` is used between the curly braces of the model. The first parameter `ARP` indicates the margin for which a design will be specified. The second parameter `7` indicates the number of log-linear effects.

The command `des` is used to specify the design matrix defining the seven parameters for margin `ARP`. As with the frequencies, one can either specify the design matrix between square

brackets or specify the name of the file from which the design matrix must be read. Note that what is specified is, in fact, a transposed design matrix, in which the first four rows refer to the one-variable parameters and the last three rows to the two-variable interactions. It is not necessary to include a main effect (a row with ones) because the main effect is automatically included in the model to ensure that the sample size is reproduced.[5]

In example 2.2a, we specified a design for the complete table `ARP`. However, the obligation to indicate for which margin a particular design will be given makes it possible to specify user-defined designs in a much more compact way. The same model can also be specified as follows:

```
* example 2.2b: compact use of cov(..)
* hierarchical model {AR,RP}
man 3
dim 2 2 3
lab A R P
mod {cov(A,1),cov(R,1),cov(P,2),cov(AR,1),cov(RP,2)}
dat ex22.fre
des [1 -1           * A 1
     1 -1           * R 1
     1 0 -1         * P 1
     0 1 -1         * P 2
     1 -1 -1 1      * AR 11
     1 0 -1 -1 0 1  * RP 11
     0 1 -1 0 -1 1] * RP 12
```

As can be seen, the model defines that there is one effect for margin `A`, one for `R`, two for `P`, one for margin `AR`, and two for margin `RP`. The design matrix can now be much smaller, namely, for each effect it contains as many number as the number of cells in the margin concerned. For example, `cov(P,2)` results in 2 times 3 numbers in the design matrix. The $\ell_{EM}$ program automatically expands the specified designs to the complete table.[6]

Example 2.2b can easily be modified in such a way that we get dummy coded log-linear parameters rather than effect coded ones. This can be accomplished with the following design matrix:

```
des [1 0          * A 1
     1 0          * R 1
     1 0 0        * P 1
     0 1 0        * P 2
     1 0 0 0      * AR 1
     1 0 0 0 0 0  * RP 11
     0 1 0 0 0 0] * RP 12
```

As can be seen, the last categories of `A`, `R`, and `P` are used as reference categories.

The second command that can be used to specify user-defined designs is `fac(..)`. This command makes it possible to specify nominal, dummy-coded, designs in a very compact way. With `fac(..)`, the same hierarchical log-linear model `{AR,RP}` could be specified as

```
* example 2.2c: use of fac(..)
* hierarchical model {AR,RP}
```

[5] When reading the design matrix, the variables change in the order in which they are specified in `dim` and `lab`, that is, the last variable changes first. The order in which they appear in `cov(..)` is irrelevant. In other words, `cov(ARP,7)` and `cov(PAR,7)` are completely equivalent statements.

[6] When reading the design matrix, it is assumed that the user-defined designs are in the order in which they appear in the model specification.

```
man 3
dim 2 2 3
lab A R P
mod {fac(A,1),fac(R,1),fac(P,2),fac(AR,1),fac(RP,2)}
dat ex22.fre
des [1 0          * A 1
     1 0          * R 1
     1 2 0        * P 1 and P 2
     1 0 0 0      * AR 11
     1 2 0 0 0 0] * RP 11 and RP 12
```

The margins for which effect are specified and the number of effects is the same as in example 2.2b with `cov(..)`. For example, `fac(P,2)` indicates that there are two effects for margin `P`. The design matrix, however, is different. The design for variable `P` consist of three numbers (1, 2 and 0) rather that 2 times 3 numbers. These numbers indicate that the first parameter (1) concerns `P=1` and the second (2) `P=2`. The zero (0) for the third level of `P` indicates that there is no parameter for `P=3`, or in other words, it is the reference category.

So far, the commands `cov(..)` and `fac(..)` were used only to specify hierarchical log-linear models. Of course, in these situations we do not need to use these commands because the model can much more easily be specified using the notation with fitted marginals.

With `cov(..)` and `fac(..)`, it is quite easy to specify non-hierarchical models which contain higher-order effects without including all the lower-order effects concerned. Suppose, for instance, that we want to modify example 2.2b by fixing the first-order effect of `A` to zero. Since the model still includes the interaction effect between `A` and `R`, it is no longer a hierarchical model. Therefore, we can no longer specify the complete model using the fitted marginals method. With `cov(..)`, such a model could be specified as follows:

```
* example 2.2d: including a higher-order effect with cov(..)
man 3
dim 2 2 3
lab A R P
mod {RP,cov(AR,1)}
des [1 -1 -1 1]  * AR 11
dat ex22.fre
```

This model includes the one-variable effects of `R` and `P` and the two-variable interactions `AR` and `RP`. As can be seen, the notation of hierarchical log-linear models and the commands for user-defined designs may be used together.

The command `cov(..)` can also be used to restrict parameters to be in a fixed proportion to one another. These types of restrictions are usually called linear or ordinal restrictions on the log-linear parameters. Suppose, for instance, that we want to make the interaction `RP` linear in `P`, with the scores -1, 0, and 1 for the categories of `P`. This could be specified as

```
* example 2.2e: linear/ordinal constraints with cov(..)
man 3
dim 2 2 3
lab A R P
mod {AR,P,cov(RP,1)}
des [-1 0 1 1 0 -1]  * RP with P linear
dat ex22.fre
```

As will be explained in section 2.3, this model describes the relationship between R and P by means of a row-association structure.

Another type of restrictions that can be imposed by means of cov(..) and fac(..) are equality constraints on the log-linear parameters. For instance,

```
* example 2.2f: equality constraints with cov(..)
man 3
dim 2 2 3
lab A R P
mod {AR,P,cov(AP,RP,2)}
des [1 0 -1 -1 0 1  * AP 11
     0 1 -1 0 1 -1  * AP 12
     1 0 -1 -1 0 1  * RP 11
     0 1 -1 0 1 -1] * RP 12
dat ex22.fre
```

gives a model in which the interaction terms AP and RP are assumed to be equal to one another. With cov(AP,RP,2) we indicate that there are two parameters which concern the margins AP and RP. In the design matrix, first the designs for the two effects for AP are defined and then for the two effects for RP. The $\ell_{EM}$ program will constrain the first effect for margin AP to be equal to the first effect for RP. In addition, the second effect for AP is made equal to the second effect for RP.

The same type of restriction could also be specified with fac(AP,RP,2). In that case, the design matrix would be of the form

```
des [1 2 0 0 0 0   * AP
     1 2 0 0 0 0]  * RP
```

The design generating command fac(..) is much more flexible than was demonstrated so far. To illustrate this, let us give another couple of examples on the use of this command. Suppose we want to specify a symmetry model for a square mobility table or some other type of turnover table. This can easily be accomplished by

```
* example 2.3a: symmetry model with fac(..)
* O=origin; D=destination
man 2
dim 5 5
lab O D
mod {fac(OD,15)}
des [1  2  3  4  5
     2  6  7  8  9
     3  7 10 11 12
     4  8 11 13 14
     5  9 12 14 15]
dat ex23.fre
```

The model defines that, besides the main effect, there are 15 parameters for the origin-destination table OD. The symmetric structure is specified in the design matrix. It should be noted that, in fact, the 15th parameter in the design is redundant. Thus, including the main effect, the model has 15 independent parameters and not 16. An alternative specification of the same symmetry model is

```
* example 2.3b: symmetry model with fac(..)
man 2
dim 5 5
lab O D
mod {fac(O,D,4),fac(OD,10)}
des [1 2 3 4 0  * O
     1 2 3 4 0  * D
     1 2 3 4 0  * OD
     2 5 6 7 0
     3 6 8 9 0
     4 7 9 10 0
     0 0 0 0 0]
dat ex23.fre
```

Here, the symmetry model is specified as a model in which the one-variable effects of `O` and `D` are equal to one another and in which the two-variable interaction term has a symmetric structure. As can be seen, the last categories of `O` and `D` are used as reference categories in the dummy-coded design that is generated.

The symmetry model can easily be transformed into a quasi-symmetry model, namely, by allowing the one-variable effect of `O` and `D` to be different. This can be realized by replacing the model by

```
{O,D,fac(OD,10)}
```

and omitting the first two rows of the design matrix.

The last example on the use of `fac(..)` concerns a quasi-independence model for the same square table. Such a model can be specified by

```
* example 2.3c: quasi-independence model with fac(..)
man 2
dim 5 5
lab O D
mod {O,D,fac(OD,5)}
des [1 0 0 0 0
     0 2 0 0 0
     0 0 3 0 0
     0 0 0 4 0
     0 0 0 0 5]
dat ex23.fre
```

Here, the quasi-independence is specified as an independence model with an additional set of parameters for the elements of the main diagonal of square table `OD`.[7]

### 2.2.2 Some predefined designs

Besides the possibility of specifying non-hierarchical log-linear models by means of user-defined designs, the $\ell_{EM}$ program contains a number predefined designs for the most common types of effects. This implies that in many situations, non-hierarchical models can be estimated without the necessity of specifying a design matrix.

The predefined design can be called with the command `spe(..)` (special designs), which complete syntax is

[7] It should be noted that the quasi-independence model can also be specified using structural zeros (see section 2.4).

```
spe(<margins>,<type of effect>,<group margin>,<a/b/c>,<# of groups>)
```

The crucial parameter is, of course, `<type of effect>`, which indicates which type of design has to generated. The possible types of effects are simple log-linear parameters, sum-score parameters, parameters of symmetric associations, diagonal and off-diagonal parameters, difference parameters, and parameters of ranking models.

Let us see how some of the input examples presented above can be simplified by means of the use of `spe(..)`. For instance, the model in which we included a higher-order effect without including particular lower order effects (example 2.2d), can also be specified as

```
mod {RP,spe(AR,1a)}
```

The `<type of effect>` '`1a`' generates the design for simple log-linear effects. The same type can be used for specifying equality restrictions between simple log-linear effects (example 2.2f). For instance,

```
mod {AR,P,spe(AP,RP,1a)}
```

will make the two-way iterations `AP` and `RP` equal to one another without the necessity of specifying a design matrix. Example 2.2e, which contains a linear constraint on variable `P` with respect to its relationship to `R`, can more easily be defined by

```
mod {AR,P,spe(RP,1b)}
```

where type `1b` indicates a uniform or linear-by-linear association parameter. The symmetry model of example 2.3a may be specified by

```
mod {spe(OD,3a)}
```

since `3a` generates exactly the symmetric structure that is set up there with `fac(..)`. Finally, the quasi-independence model of example 2.3c can also be defined by

```
mod {O,D,spe(OD,5a)}
```

because type `5a` yields a set of parameters for the main diagonal.

### 2.2.3 Using a group margin

So far, no attention was dedicated to the last three (optional) parameters that can be used with the commands `cov(..)`, `fac(..)`, and `spe(..)`. These parameters make it possible to let the specified effects differ across levels of some other variables, which are called grouping variables. In fact, they can be used to specify higher-order interaction terms in a compact way.

Suppose that we are analyzing simultaneously the mobility tables of different birth cohorts (generations), and that we assume just a simple quasi-independence model for the relationship between the occupation of the father and the occupation of the son. A possible specification for the three-way table concerned could be

```
* example 2.4: the use of grouping variables
* C=cohort; F=father; S=son
man 3
dim 7 5 5
lab C F S
mod {CF,CS,fac(FS,5,C,c)}
des [1 0 0 0 0
     0 2 0 0 0
     0 0 3 0 0
     0 0 0 4 0
     0 0 0 0 5]
dat ex24.fre
```

The hierarchical `CF` and `CS` terms indicate that the first-order effects of `F` and `S` differ across cohorts. Moreover, with the last two parameters in `fac(FS,5,C,c)`, it is indicated that the diagonal parameters defined via `fac(..)` differ for the various levels of `C`. The fourth parameter can have values `a` (homogeneous), `b` (simple heterogeneous), or `c` (heterogeneous). Homogeneous means that there is no interaction, which is the same as not using a group margin. Simple heterogeneous gives a log-multiplicative interaction structure, that is, a multiplicative scaling factor for each level of the joint grouping variable (Xie, 1992; Vermunt, 1996b, 1997).[8] And finally, heterogeneous models yield standard log-linear higher-order interaction terms.

The fifth and last parameter (`<# of groups>`) can be used to further restrict the interaction terms which result from using a group margin. In fact, it allows to specify a design for the group margin. A negative number means that this will be an interval design and a positive number that this will be a nominal design. For example, `fac(FS,5,C,c,3)`, in combination with the additional line

```
1 1 1 2 2 3 3
```

in the design matrix, yields a model in which the diagonal parameters are equal across cohorts 1, 2, and 3, across 4 and 5, and across 6 and 7. On the other hand, `fac(FS,5,C,c,-2)` in combination with an interval level design of the form

```
 1  1  1  1  1  1  1
-3 -2 -1  0  1  2  3
```

will produce linearly changing diagonal parameters. The same kinds of restrictions on the group margin can be used in combination with a log-multiplicative interaction structures, that is, with a simple heterogenous model (Vermunt, 1996b, 1997).[9]

## 2.3 Association models

The $\ell_{EM}$ program has a special set of tools for specifying models with uniform (U), row (R), column (C), and row and column (R+C) associations. These log-linear models is often referred to as association models type I (Goodman, 1979, 1991; Haberman, 1978). Moreover, it is possible to specify log-multiplicative row-column (RC) associations, often referred to as association models type II, and their multidimensional variants, the so-called RC(M) models (Goodman, 1979, 1986, and 1991; Clogg, 1982, Clogg and Shihadeh, 1994).

Assume that $A$ is the row variable and $B$ the column variable. Uniform (U), row (R), column (C), and row and column (R+C) associations are obtained by restricting the two-variable interaction term $u_{ab}^{AB}$ by

$$\begin{aligned} u_{ab}^{AB} &= u^{AB}\, x_a\, x_b\,, \\ u_{ab}^{AB} &= u_a^{AB}\, x_b\,, \\ u_{ab}^{AB} &= u_b^{AB}\, x_a\,, \\ u_{ab}^{AB} &= u_a^{AB}\, x_b + u_b^{AB}\, x_a\,, \end{aligned}$$

respectively. Here, $u^{AB}$, $u_a^{AB}$, and $u_b^{AB}$ denote restricted interaction terms and $x_a$ and $x_b$ equidistant row and column scores.[10] The log-multiplicative RC model restricts

$$u_{ab}^{AB} = \mu_a\, \phi\, \nu_b\,,$$

[8] For identification, the scaling factor of the first group is fixed to be equal to one.

[9] As in unrestricted log-multiplicative models, also in restricted log-multiplicative models the first group parameter is fixed to one for identification.

[10] It should be noted that this type of constraints can also be specified via user-defined designs, that is, with the command cov(..).

where $\mu_a$ and $\nu_b$ are row and column scores to be estimated and $\phi$ is an association parameter.[11] In square tables, we will sometimes want to restrict the row and column scores to be equal to one another (Goodman, 1979; Luijkx, 1994). A RC(M) model involves restricting the two-variable interaction $u_{ab}^{AB}$ by

$$u_{ab}^{AB} = \sum_{m=1}^{M} \mu_a^m \phi^m \nu_b^m \,, \tag{2.3}$$

where $M$ is the number of dimensions.[12] An important recommendation has to be made with respect to the use RC and RC(M) models. Since these models may have local maxima, one must always try out different sets of (random) starting values.

The models presented above all have their multiple-group variants in which the parameters are allowed to differ across levels of some other variables (Clogg, 1982; Clogg and Shihadeh, 1994). For the log-linear association models, the multiple-group variants are equivalent to the way we used a grouping variable in subsection 2.2.3. In RC models, however, the use of a grouping variable is less standard because it leads to several variants. In its most general form, the multiple-group RC model is given by

$$u_{ab}^{AB} + u_{abc}^{ABC} = \mu_{ac}\, \phi_c\, \nu_{bc} \,,$$

where $C$ is the grouping variable. Partially heterogenous specifications are obtained by assuming the row scores, the column scores, or both not to depend on $C$. In almost the same way, one can define a multiple-group variant of the RC(M) model (Becker and Clogg, 1989), that is,

$$u_{ab}^{AB} + u_{abc}^{ABC} = \sum_{m=1}^{M} \mu_{ac}^m\, \phi_c^m\, \nu_{bc}^m \,.$$

Although it is possible, in RC(M) models, one will generally not use the above-mentioned partially heterogenous specifications.

There are three commands which can be used in to specify association models. The command `ass1(..)` can be used to specify log-linear association models, that is, U, R, C, and R+C models. With `ass2(..)`, one can define log-multiplicative RC association models. And finally, `ass3(..)` defines a variant of the R+C models in which the grouping variable enters log-multiplicatively (Xie, 1992). The complete syntax of these commands is

```
ass1/2/3(<row margin>,<column margin>,<group margin>,<type of model>,
         <type of symmetry>,<# of rows>,<# of columns>,<# of groups>)
```

The specification of a `<row margin>` and a `<column margin>` is, of course, obligatory. The optional specification of a `<group margin>` makes it possible to allow parameters to vary among levels of some other variables. The value of the parameter `<type of model>` consists of a number (between 2 and 6) and a letter (between a and e). The number indicates the type of association model and the letter the way that the parameters vary across levels of the (joint) grouping variable.[13]

[11] Identifying restrictions must be imposed on the parameters of RC models. In the default setting, $\ell_{EM}$ identifies the row, column, and association parameters by fixing the unweighted mean of the row and column scores to 0 and their unweighted standard deviation to 1. These identifying restrictions can be changed by means the command `sca`.

[12] Additional identifying restrictions have to be imposed compared to the RC (or RC(1)) model. In $\ell_{EM}$, the unweighted scores of the various dimensions are orthogonalized by means of a singular value decomposition.

[13] The exact meaning of these numbers and letters is described in chapter 11.

Suppose we have a 6 by 4 table for which we want to specify a RC model. This could be specified as follows:

```
* example 2.5: the use of ass2(..)
man 2
dim 6 4
lab R C
mod {R,C,ass2(R,C,5a)}
dat ex25.fre
```

where `ass2(..)` indicates that it is a log-multiplicative model, and `5a` that it is a row and column model (`5`) with equal parameters across groups (`a`).[14] Now let us introduce a grouping variable `G` with 5 categories. In that case, we could, for example, have a model like

```
* example 2.6a: ass2(..) with a grouping variable
man 3
dim 5 6 4
lab G R C
mod {GR,GC,ass2(R,C,G,5e)}
dat ex26.fre
```

Here, the type of association model, `5e`, indicates that it is a row and column model (`5`) with different row, column, and association parameters (`e`) for the levels of the grouping variable `G`.

RC(M) models can simply be specified by calling the `ass2(..)` command several times in combination with models 5a or 5e. For instance, example 2.5, can be transformed into a RC(2) model by replacing the model by

```
mod {R,C,ass2(R,C,5a),ass2(R,C,5a)}
```

With the parameter `<type of symmetry>` it is possible to constrain row and column scores to be equal in different partial associations (Clogg, 1982). An example of the use of this option is

```
* example 2.6b: ass2(..) with symmetry
man 3
dim 5 6 4
lab G R C
mod {G,R,C,ass2(G,C,5a,a),ass2(G,R,5a,a),ass2(R,C,5a,a)}
dat ex26.fre
```

With '`a`', the last parameter in the `ass2(..)` statements, it is indicated that the scores for `G`, `R`, and `C` are equal in all partial associations.

And finally, the last three parameters of `ass1/2/3(..)`, `<# of rows>`, `<# of columns>`, and `<# of groups>`, make it possible to further restrict the row, column, and association parameters. Their use is exactly the same as that of the parameter `<# groups>` in the user-defined designs (see subsection 2.2.3). A zero means no restrictions, a positive number indicates that a nominal design will be given, and a negative number that an interval design will be given. An example of the use of these options is

```
* example 2.6c: designs for row, column, and group variables
man 3
dim 5 6 4
```

[14] Since we did not specify a grouping variable, the `a` in the model specification is, in fact, redundant.

```
lab G R C
mod {GR,GC,ass2(R,C,G,5b,4,3,-2)}
des [1 1 2 2 3 4
     1 2 3 3
     1 1 1 1 1 1
     -2.5 -1.5 -.5 .5 1.5 2.5]
dat ex26.fre
```

Here, `5b` indicates that we have a simple heterogenous model, that is, a model in which only the $\phi$ parameters depend on `G`. Furthermore, it is indicated that there are 4 different row scores and 3 different column scores, and that we want to specify an interval design for $\phi_g$ consisting of two parameters. In `des`, it is specified which rows and columns have equal scores. In addition, the design for the association parameters is given. Applications of RC models with this type of restrictions on the grouping variable can be found in Luijkx (1994) and Wong (1995).

## 2.4 Using a weight vector

The general log-linear model described in equation 2.2 can be extended by one additional component $z_i$, that is, a weight vector:

$$\log(m_i/z_i) \;=\; \sum_j \beta_j x_{ij}\,. \tag{2.4}$$

The weight vector[15] can, among other things, be used to specify log-rate models, models with structural zeros, and models with fixed effects (Haberman, 1978; Laird and Oliver, 1981; Willekens and Shah, 1983; Clogg and Eliason, 1987).

Suppose we want to specify the same quasi-independence model as in example 3d, but now using structural zeros for the elements of the main diagonal instead of diagonal parameters. Such a model can be specified by

```
* example 2.7: quasi-independence model with wei(..)
man 2
dim 5 5
lab O D
mod {O,D,wei(OD)}
sta wei(OD) [0 1 1 1 1
             1 0 1 1 1
             1 1 0 1 1
             1 1 1 0 1
             1 1 1 1 0]
dat ex27.fre
```

With `wei(OD)`,[16] it is indicated that there will be specified a set of weights for the margin `OD`. The weights are specified with the command `sta` (starting value). As can be seen, for each of the main diagonal elements we specified a weight of zero, which means that they are treated as structural zeros.

An example of a log-rate model could concern an analysis of death rates. For instance, assume that we have an age (7 age groups) by period (5 periods) table with number of deaths and a 7 by 5 table with number of persons at risk in each age-period combination. An example of a log-rate model for this data is

[15] In SPSS, an element $z_i$ is called a cell weight. In the GLIM terminology, the vector with elements $\log(z_i)$ is called an offset.

[16] The use of the command `wei` is changed compared to the experimental version 0.11 of the $\ell EM$ program.

```
* example 2.8a: log-rate model
* A=age; P=period
man 2
dim 7 5
lab A P
mod {A,P,wei(AP)}
sta wei(AP) ex28.wei
dat ex28.fre
```

The files `ex28.fre` and `ex28.wei` contain the observed number of deaths and the observed size of the risk population for each combination of `A` and `P`. This example can easily be transformed into an age-period-cohort (APC) model (Fienberg and Mason, 1979; Hagenaars, 1990). With `fac(..)`, we can add a cohort effect to the model as follows:

```
* example 2.8b: APC model for rates
man 2
dim 7 5
lab A P
mod {A,P,fac(AP,11),wei(AP)}
des [7  8  9 10 11     * A=1
     6  7  8  9 10     * A=2
     5  6  7  8  9     * A=3
     4  5  6  7  8     * A=4
     3  4  5  6  7     * A=5
     2  3  4  5  6     * A=6
     1  2  3  4  5]    * A=7
sta wei(AP) ex28.wei
dat ex28.fre
```

It should be noted that to identify the parameters of APC models, one has to impose additional identifying restrictions on the model parameters.[17]

## 2.5 Linear restrictions on cell frequencies

Besides specifying models with log-linear and log-multiplicative terms, it is also possible to impose linear restrictions on the cell frequencies (see Haber and Brown, 1986). Each of the linear restrictions will be of the form

$$\sum_j c_j \sum_i a_{ij} m_i = 0\,. \tag{2.5}$$

The matrix $\mathbf{A}$, which consists of ones and zeros, specifies a number of sums of frequencies (or margins), while the vector $\mathbf{c}$ defines a contrast for these sums of frequencies. This model is, actually, a special case of the general class of marginal models proposed by Lang and Agresti (1994) and further developed by Bergsma (1997).

These linear restrictions can be specified in $\ell_{EM}$ with the command `lin(..)`, which complete syntax is

```
lin(<margins>,<# of constraints>)
```

[17] The cohort parameters can also be specified via a predefined design, in this case, via a set of difference score parameters.

Note that it is not necessary to specify the elements of **A** since the program automatically computes the requested margins. This means that only the vector of contrast has to be specified.

An important application of the linear model for cell frequencies is the marginal homogeneity model. A marginal homogeneity model for a 5 by 5 mobility table can be specified by

```
* example 2.9: marginal homogeneity
man 2
dim 5 5
lab O D
mod {lin(O,D,4)}
dat ex29.fre
des [1 0 0 0 0   * O=1
     0 1 0 0 0   * O=2
     0 0 1 0 0   * O=3
     0 0 0 1 0   * O=4
     -1 0 0 0 0  * D=1
     0 -1 0 0 0  * D=2
     0 0 -1 0 0  * D=3
     0 0 0 -1 0] * D=4
```

With `lin(O,D,4)`, it is specified that there are four linear restrictions in which the margins `O` and `D` are involved.[18] The first four lines in the design matrix are the elements of **c** for margin `O`, the other lines for corcern margin `D`. As can be seen, the first element of the margin `O` minus the first element of the margin `D` is specified to be zero. The same applies to the second, third, and fourth elements of `O` and `D`.

## 2.6 Correspondence analysis

With $\ell_{EM}$, it is also possible to perform correspondence analysis (Greenacre, 1984; Gifi, 1990). Although this technique seems to fall outside the $\ell_{EM}$ modeling framework, it is implemented because it is an important explorative analysis method for categorical data. In addition, there is a strong relationship between correspondence analysis and the RC models discussed in section 2.3 (Goodman, 1986, 1991).

Correspondence analysis can be requested with the command `cor(..)`. An example of correspondence analysis of a two-way table is

```
* example 2.10: correspondence analysis
man 2
dim 7 7
mod cor(1)
fre ex210.fre
```

With `cor(1)` after `mod`, it is indicated that the 'model' is a correspondence analysis type `1`, which is a simple correspondence analysis of a two-way table.[19]

An example of multiple correspondence analysis of a six-way frequency table is

```
* example 2.11: multiple correspondence analysis
man 6
dim 2 5 3 2 5 2
mod cor(2,3,2)
dat ex211.fre
```

[18] Note that the restriction for the fifth margin is omitted because it is redundant.

[19] Correspondence analysis of two-way tables is sometimes also referred to as (canonical) correlation analysis.

The value 2 of the first parameter in `cor(2,3,2)` indicates that it is a multiple correspondence analysis.[20] The second (optional) parameter denotes the number of dimensions for which one wants output on variables and categories. The third parameter, which is optional as well, indicates for how many dimensions one wants object scores.

Besides the two correspondence analysis methods presented in the above examples, it is possible to request an association analysis with marginal or uniform weights. These association analysis models – specified by `cor(3)` and `cor(4)`, respectively – are least squares variants of the log-multiplicative association models presented in section 2.3.

## 2.7 Using record type data

So far, we assumed that the data is specified in the form of a frequency table which is either included in the input file or read from another file. It is, however, also possible to use data in the form of individual records. To indicate that the data are individual records, one only has to specify the number of records with the command `rec`. An example is

```
* example 2.12a: use of record type data
man 3
dim 2 2 3
lab A R P
mod {AR,RP}
rec 221        * the data file contains 221 records
dat ex212.dat  * name of the data file
```

where the first five records of the data file `ex210.dat` could be

```
1 1 1
1 2 1
2 2 3
1 2 3
2 2 2
etc.
```

As can be seen, the data file contains the values of the variables `A`, `R`, and `P` for each person. The values may range from 1 to the number of categories of the variable concerned.

The commands that define the format of the data must be given after the model specification. Two important additional command with regards to the format of the data are the `ski [..]` (skip columns) and `rco` (read count). The command `ski [..]` can be used to indicate that some columns must be skipped when reading the data file. This makes it possible to have more variables in the data file than are actually used in the model to be estimated. With `rco`, one can indicate that the records contain a count or frequency. For example,

```
* example 2.12b: use of ski and rco
man 3
dim 2 2 3
lab A R P
mod {AR,RP}
rec 20         * data file contains 20 records
ski [2 4]      * skip the second and the fourth column
rco            * the records contain a count
dat ex212.dat
```

[20] Multiple correspondence analysis is sometimes also referred to as optimal scaling or homogeneity analysis.

where the first five records of the data file could now be

```
1 3 1 1 1  30
1 4 2 1 1   2
2 1 2 1 3  23
1 4 2 2 3   5
2 3 2 2 2 234
etc.
```

As indicated with `ski`, only the variables in columns 1 (`A`), 3 (`R`), and 5 (`P`) are actually used in the analysis. The last column contains a count or frequency.

# Chapter 3

# Regression models

The previous chapter presented log-linear models in which no distinction is made between dependent and independent variables. This chapter discusses the estimation of regression models for categorical dependent variables with $\ell_{EM}$.

## 3.1 Multinomial logit models

The multinomial logit model is a special case of the log-linear models discussed in the previous section. Assume that we are analyzing a three-way table $ABC$ in which $C$ is the dependent variable. In that case, the saturated logit model equals

$$\log m_{abc} \quad = \quad \alpha_{ab}^{AB} + u_c^C + u_{ac}^{AC} + u_{bc}^{BC} + u_{abc}^{ABC} \,. \tag{3.1}$$

In fact, this model is very similar to the saturated log-linear model described in Equation 2.1. The only difference is the inclusion of the $\alpha_{ab}^{AB}$ parameters which assure that the margin of the joint independent variable, $AB$, is reproduced.[1]

As demonstrated by Haberman (1979), in its most general form the multinomial logit model, or multinomial response model, can be written down as

$$\log m_{ik} \quad = \quad \alpha_k + \sum_j \beta_j x_{ijk} \,, \tag{3.2}$$

where $k$ is used as the index for the joint distribution of the independent variables and $i$ as an index for the levels of the response variable. The $\alpha_k$ parameters make the margin of the joint independent variables fixed.

Another way to specify the multinomial logit model is as a logistic model for the probability of having value $i$ on the response variable given that one has value $k$ on the joint independent variables, that is, for $\pi_{i|k}$. This yields

$$\pi_{i|k} \quad = \quad \frac{\exp\left(\sum_j \beta_j x_{ijk}\right)}{\sum_l \exp\left(\sum_j \beta_j x_{ljk}\right)} \,. \tag{3.3}$$

Let us start with the same frequency table as in examples 2.1 and 2.2 of chapter 2. A multinomial logit model for variable `P` with two independent variables `A` and `R` can be specified as follows:

```
* example 3.1a: multinomial logit model for P
man 3
dim 2 2 3
```

[1] It should be noted that, in fact, $\alpha_{ab}^{AB} = u + u_a^A + u_b^B + u_{ab}^{AB}$.

```
lab A R P
mod {AR,AP,RP}
dat ex31.fre
```

Here, the multinomial logit model is defined as a hierarchical log-linear model. The margin of the independent variables, `AR`, is reproduced by including the term `AR` in the model. In addition, there are interactions between `R` and `P` and between `A` and `P`.

An alternative specification, which is more close to the model description in equation 3.3, is obtained by replacing the model specification by

```
mod P|AR {AP,RP}
```

By the statement `P|AR` before the model specification, it is indicated that a multinomial logit is specified for $\pi_{p|ar}$ rather than a log-linear model for expected cell frequency $m_{arp}$. As will be explained in more detail in chapter 4, this specification decomposes the joint distribution of `A`, `R`, and `P`, $\pi_{arp}$, into $\pi_{ar}\pi_{p|ar}$. No restrictions are imposed on $\pi_{ar}$, while the requested multinomial logit is estimated for $\pi_{p|ar}$.

The commands for specifying non-hierarchical log-linear models can also be used in logit models. Suppose we want to treat the trichotomous dependent variable `P` as ordered by imposing restrictions on the two-variable interactions `PA` and `PR`, for instance, by assuming them to be linear in `P`. This can be accomplished by means of `cov(..)`, that is, by replacing the model specification by

```
mod P|AR {P,cov(PA,1),cov(PR,1)}
```

and including the design matrix

```
des [-1 0 1 1 0 -1
     -1 0 1 1 0 -1]
```

As can be seen from the design matrix, we assigned scores -1, 0, and 1 to the levels of `P`. Another option is to estimate the scores of `P` rather than fixing them. This can be done with a log-multiplicative RC model, that is,

```
mod P|AR {P,ass2(A,P,5a,a),cov(R,P,5a,a)}
```

Note that we assume that the scores for `P` are equal in both two-variable interaction terms.

## 3.2 Cumulative link functions

As was demonstrated above, the multinomial logit model can be restricted to take the order of the categories of the dependent variables into account by assigning or estimating category scores. This is, however, not the only possible method for dealing with ordinal dependent variables. An alternative set of models involves defining a linear model for a transformation of the cumulative response probabilities (McCullagh and Nelder, 1989; Agresti, 1990). The $\ell EM$ program can deal with four types of cumulative response models: logit, probit, complementary log-log, and log-log models.[2] These cumulative models are of the form

$$g[P(Y \leq i|k)] \quad = \quad \gamma_i + \sum_j \beta_j x_{jk}\,, \tag{3.4}$$

[2]These models can, of course, also be used if the dependent variable is dichotomous. In that case, the binary version of the model concerned is obtained.

where $Y$ is the dependent variable, $i$ is a value of $Y$, $g[...]$ is the link function and $\gamma_i$ the threshold parameter belonging to $i$th level of the dependent variable $Y$.[3]

An additional restriction that can be imposed on the model described in equation 3.4 is assuming the threshold parameters to be equidistant, that is,

$$\gamma_i \;=\; \gamma + x_i\,\gamma' . \tag{3.5}$$

Here, $x_i$ is the score assigned to level $i$ of the response variable. In $\ell EM$, these category scores have a mutual distance of 1 and a mean of 0. This leads to a regression model in which the response variable is discrete and assumed to be of interval measurement level.

Suppose we want to transform the multinomial logit model of example 3.1a into a cumulative logit model. This can be done by

```
* example 3.1b: cumulative logit model
man 3
dim 2 2 3
lab A R P
mod P|AR cum(a) {cov(A,1) cov(R,1)}
dat ex31.fre
des [1 -1    * A
     1 -1]   * R
```

The main difference with the multinomial logit model is the inclusion of the statement `cum(a)`. This indicates that one wants to estimate a cumulative model of type '`a`', which is a cumulative logit model. Types '`b`', '`c`', and '`d`' refer to a probit, complementary log-log, and log-log model, respectively, while their restricted variants (see equation 3.5) are denoted by '`e`', '`f`', '`g`', and '`h`'.[4]

Another difference with the specification of multinomial logit models is that it is no longer possible to specify effects using the simple fitted marginals notation. In cumulative models, the effects must be specified via user-defined designs (`cov(..)` or `fac(..)`) or predefined designs (`spe(..)`).

## 3.3 Continuous covariates

The $\ell EM$ program also allows the user to include continuous variables as exogenous variables in a regression model for a categorical response variable. To make that possible, we have to define the regression model concerned on the individual level rather than on the level of a frequency table.

Let the index $k$ in the regression models described in equations 3.2, 3.3, and 3.4 now denote a particular individual observation rather than a cell in the marginal distribution of the independent variables. This means that a particular $x_{ijk}$ or $x_{jk}$ contains the value of observation $k$ on the independent variable $j$ (for response category $i$).

An example of a multinomial logit model for dependent variable `P` with two continuous covariates is:

```
* example 3.2: logit model with continuous covariates
man 1
con 2
```

[3] Since, except for the log-log model, positive values of $\beta_j$ imply a negative relationship between the independent and response variables, the sign of the $\beta_j$ parameters is sometimes reversed in cumulative response models. Here, we just work with the description in formula 3.4.

[4] It is also possible to specify a linear regression model with a normally distributed error term via type '`i`'.

```
dim 3
lab P x
mod P|x {P,cov(x,1,P,c,-2),cov(x,2,P,c,-2)}
rec 100
des [1 0 -1
     0 1 -1
     1 0 -1
     0 1 -1]
dat ex10.dat
```

The statement `con` is used to specify the number of continuous variables. The label for the continuous covariates is '`x`', which is the same as the default label. Continuous variables can be used in the regression model by means of the command `cov(..)`. The first parameter in `cov(..)` must be the label of the continuous covariates, in this case, `x`; the second one indicates the number of the covariate. So, here, we included both the effect of the first and second continuous covariate on `P` in the model. As can be seen, dependent variable `P` is used as a grouping variable to create and interaction between `P` and the covariate concerned. The contrasts in the design matrix indicate that an effect-coding scheme is used for grouping variable `P` to identify the parameters.[5]

Another way to use continuous covariates in the model specification is as grouping variables in `cov(..)`, `fac(..)`, or `spe(..)`. The above model can also be specified as

```
mod P|x {P,spe(P,1a,x,c,1),spe(P,1a,x,c,2)}
```

Here, `spe(..)` is used to specify an interaction term between `P` and the two continuous covariates. The last parameter in `spe(..)` denotes the covariate number. Note that we no longer need to specify a design matrix since predefined design type `1a` generates the necessary contrast for `P`.

If continuous exogenous variables are used, the data file must be in the form of individual records. The first columns must contain the categorical variables. As indicated in the above input file, the data file `ex10.dat` will contain 100 records. The first five records of this file might be

```
1  10   3.5
2   8   4.6
1   3   7.1
3  11   3.0
2   5  -0.2
etc.
```

Of course, the data file could contain more columns, which could be skipped with `ski [..]`. It is also possible to add a count to the records (see section 2.7).

The above example can easily be transformed into, for example, a cumulative probit model for `P`. In that case, the model specification would be

```
mod P|x cum(b) {cov(x,1) cov(x,2)}
```

We no longer need to use `P` as a grouping variable in `cov(..)` because, as in all cumulative response models, the covariate effects do not interact with the dependent variable.

[5]The use of grouping variables is explained in subsection 2.2.3.

# Chapter 4

# Path models

Path models are structural models with more than one endogenous variables. In the same way as ordinary path models consist of a set or ordinary regression equations, log-linear path models consist of a series of logit models for a set of categorical endogenous variables (Goodman, 1973).

## 4.1 Conditional probability structure

Suppose we want to specify a path model with two endogenous variables, political preference ($P$) and voting behavior ($V$), and two exogenous variables, age ($A$) and religion ($R$). The first step in defining a path model for the four-way table $ARPV$ is the decomposition of their joint probability, $\pi_{arpv}$, into a set of marginal and conditional probabilities on the basis of the assumed causal ordering of the variables (Goodman, 1973; Wermuth and Lauritzen, 1983, 1990). In this case, the following decomposition seems to be most appropriate:

$$\pi_{arpv} \quad = \quad \pi_{ar}\, \pi_{p|ar}\, \pi_{v|arp}\,. \tag{4.1}$$

Such a model can be specified with $\ell_{EM}$ by

```
* example 4.1: conditional probability structure
man 4
dim 2 2 3 2
lab A R P V
mod AR
    P|AR
    V|ARP
dat [15 16 50 27 30 5 18 50 31 29 51 14
     35 9 127 20 58 3 21 47 30 20 40 4]
```

As can be seen, with `mod` we can specify the requested probability structure.[1] The model described in equation 4.1 and specified in example 4.1a is, in fact, an unrestricted model. It is, however, possible to modify the probability structure in such a way that one obtains a restricted model. For example, a model in which `P` is assumed not to depend on `A`, and `V` not to depend on `R` is obtained by replacing the model specification by[2]

[1] This specification in terms of conditional probabilities is an important change compared to the experimental version 0.11 of the $\ell_{EM}$ program.

[2] It should be noted that if we are not interested in the parameters of `AR`, we can also omit `AR` from the model specification: Variables which are used as independent but not as dependent are automatically treated as exogenous. Moreover, variables which do not appear in the model specification are assumed to be endogenous but independent of the other variables.

```
mod AR
    P|R
    V|AP
```

The possibility of specifying such restricted probability structures can, among other things, be used for defining discrete-time Markov models (Bishop, Fienberg, and Holland, 1975; Agresti, 1990). Suppose we have panel data which provide information on the same variable at five points in time. A first-order Markov model for the five-way cross-tabulation concerned can be specified by

```
* example 4.2a: first-order Markov model
man 5
dim 2 2 2 2 2
mod A B|A C|B D|C E|D
dat ex42.fre
```

where `A-E` are the labels for the variable of interest at the different point in time. It can easily be seen that the value of the response variable at a particular point in time depends only on the value at the previous point in time.

## 4.2 Logit parameterization

Specification of either a restricted or unrestricted probability structure is usually only the first step in the specification of a path model for categorical data. In most situations, we want to further restrict the probabilities by means of a set of regression equations. In $\ell_{EM}$, it is possible to restrict each of the probabilities by means of a log-linear model, a multinomial logit model, or one of the cumulative response models.[3]

Suppose that we want to specify a path model consisting of a series of logit models for the table `ARPV` (see example 4.1). This model could, for instance, be of the form

```
mod AR
    P|R {RP}
    V|ARP {AV,RV,PV}
```

Here, the margin `AR` is fixed. Furthermore, `P` is assumed not to depend on `A`, and a saturated logit model is specified for $\pi_{p|r}$. Note that exactly the same model is obtained if '`P|R {PR}`' is replaced by '`P|AR {PR}`', that is, by restricted logit model for $\pi_{p|ar}$. Finally, a logit model without three- and four-variable interaction terms is defined for dependent variable `V`.

In fact, every type of specification that is allowed in the models described in chapters 2 and 3 can also be used in path models, that is, hierarchical and non-hierarchical models, association models, cumulative link functions, etc.. If no model is specified for a probability, the program will assume a saturated log-linear model for the probability concerned.

There is one additional feature that is relevant in the context of log-linear path models, namely, the possibility to impose equality constraints on the parameters appearing in different submodels.[4] Suppose, for instance, that we want to specify a first-order Markov model in which the association structure between subsequent time points is equal for all transition probabilities. This can be accomplished as follows:

[3] In his paper on modified path models, Goodman (1973) proposed specifying log-linear or logit models for the unrestricted set of probabilities. As demonstrated by Vermunt (1996a, 1996b, 1997), it is computationally more efficient to specify the regression models for the restricted probabilities. In addition, it may prevent fitted zeros.

[4] In this context, the term submodel is used to denote the model for one of the probabilities in a path model.

```
* example 4.2b: restrictions across submodels via all
man 5
dim 2 2 2 2 2
mod A {A} B|A {B} C|B {C} D|C {D} E|D {E}
all {cov(AB,BC,CD,DE,1)}
des [1 -1 -1 1  * AB
     1 -1 -1 1  * BC
     1 -1 -1 1  * CD
     1 -1 -1 1] * DE
dat ex42.fre
```

In the model specification, we defined only the one-variable effects, that is, the effects which differ across time points. The command `all {<parameters>}` is used to specify an additional set of parameters which may appear in any of the submodels. Between the parentheses one may use the commands `cov(..)`, `fac(..)`, `spe(..)`, `ass1(..)`, `ass2(..)`, and `ass3(..)`. The use of these commands is described in subsections 2.2 and 2.3. Above, `cov(..)` – in combination with the correct design matrix – was used to constrain the two-variable interactions to be equal across time point.[5]

## 4.3 Restricting probabilities

The previous two sections described two different methods for specifying restricted path models: omitting particular variables from a conditional probability and restricting a probability by means of a log-linear or cumulative link parameterization. The $\ell_{EM}$ program contains two additional commands to specify constraints on the probabilities: `eq1` to specify equality constraints and `eq2` to specify both equality and fixed-value restrictions.[6]

The use of the command `eq1` is quite simple. Suppose for instance, that we want to modify the above Markov model into a stationary Markov model, in other words, $\pi_{b|a} = \pi_{c|b} = \pi_{d|c} = \pi_{e|d}$. This can be accomplished by

```
* example 4.2c: the use of eq1
man 5
dim 2 2 2 2 2
mod A {A}
    B|A {AB}
    C|B eq1 B|A
    D|C eq1 B|A
    E|D eq1 B|A
dat ex42.fre
```

With the command `eq1`, one can indicate that a particular set of conditional probabilities equals another set of probabilities.[7] In this case, we made `C|B`, `D|C`, and `E|D` equal to `B|A`. Although here we specified a just simple saturated for `B|A`, any type of model could be specified for the probability appearing after `eq1`.

The use of `eq2` is a bit more complicated because it requires the specification of a kind of design matrix. The same model as in example 4.2c can be specified with `eq2` as

[5] We could replace `cov(..)` by `spe(AB,BC,CD,DE,1a)`, the predefined design for simple effects. In that case, it is no longer necessary to specify a design matrix.

[6] Although not discussed below, the command `lin(...)` described in section 2.5 can also be used to impose any type of linear restriction on the probabilities.

[7] An important requisite to be able to use `eq1` is, of course, that the two sets of probabilities have the same structure: The number and order of the categories of the dependent and independent variables must be equal.

```
* example 4.2d: the use of eq2
man 5
dim 2 2 2 2 2
mod A
    B|A eq2
    C|B eq2
    D|C eq2
    E|D eq2
des [1 0 2 0    * B|A
     1 0 2 0    * C|B
     1 0 2 0    * D|C
     1 0 2 0]   * E|D
dat ex42.fre
```

The numbers in the design matrix have the following meaning: 0 indicates a free parameter, -1 denotes a parameter which has a fixed value, and positive numbers are parameters which are restricted to be equal, where the parameters with the same number are equal. So, here, the (1,1) and the (2,1) combinations are constrained to be equal across time points, while the (1,2) and the (2,2) combinations are unrestricted.[8]

To demonstrate the flexibility of `eq2`, let us present another (fictive) restricted Markov model,

```
* example 4.2e: the use of eq2
man 5
dim 2 2 2 2 2
mod A
    B|A eq2
    C|B eq2
    D|C eq2
    E|D eq2
des [1 0 0 2    * B|A
     1 0 0 3    * C|B
     -1 0 0 2   * D|C
     3 0 -1 0]  * E|D
sta D|C [.9 .1 .5 .5]
sta E|D [.5 .5 .2 .8]
dat ex42.fre
```

Here, some conditional probabilities are made equal to one another, and others are fixed to a particular value. From the design matrix, it can be seen that it is assumed that the probability of $B = 1$ given $A = 1$ equals the probability of $C = 1$ given $B = 1$, the probability of $B = 2$ given $A = 2$ equals the probability of $D = 2$ given $C = 2$, and the probability of $C = 2$ given $B = 2$ equals the probability of $E = 1$ given $D = 1$. In addition, the probability of $D = 1$ for $C = 1$ and the probability of $E = 1$ for $D = 2$ are fixed to specific values, which are is specified with `sta`. The command `sta` (starting value) can be used to specify starting values for the model parameters. The fixed probabilities will, of course, retain their starting values, in this case, .9 and .2, respectively.

On the one hand, `eq2` is more flexible than `eq1`, because with `eq2` it is not necessary to specify two complete sets of probabilities to be equal to one another. But, on the other hand, when using `eq2`, it is no longer possible to specify a regression model for the probabilities concerned.

[8] It should be noted that, in this case, the same constraints can also be imposed with a log-linear parameterization of the conditional probabilities, for instance, by means of the command `fac(..)`.

Besides equality and fixed-value restrictions, it is possible to impose a specific type of inequality restrictions on conditional probabilities. More precisely, a non-parametric ordinal model can be specified for probabilities which consist of one ordered independent variable and one ordered dependent variable (Croon, 1990).

Suppose we have a dependent variable $Y$ with levels $i$ and an independent variable $X$ with levels $k$. If here is a strictly positive relationship between $Y$ and $X$, the cumulative response probability, $P(Y \leq i|X = k)$, must satisfy

$$P(Y \leq i|X = k) < P(Y \leq i|X > k)\,. \tag{4.2}$$

On the other hand, a strictly negative relationship implies that

$$P(Y \leq i|X = k) > P(Y \leq i|X > k)\,. \tag{4.3}$$

Putting `or1` behind a probability will produce the restrictions described in equation 4.2, while `or2` will give the ones of equation 4.3. Examples on the use of these commands will be presented in the context of ordinal latent class models (see subsection 5.2.3).

# Chapter 5

# Latent class models

One of the most important features of the $\ell_{EM}$ program is that it cannot only deal with observed (manifest) variables, but also with unobserved (latent) variables. This makes it, among other things, possible to specify factor analytic models for categorical latent variables with categorical indicators. These models are called latent class models (LCM).[1]

## 5.1 Unrestricted latent class models

In the classical formulation, the latent class model is defined as a probability model (Lazarsfeld and Henry, 1968; Goodman, 1974). Suppose we have four manifest variables denoted by $A$, $B$, $C$, and $D$ which serve as indicators for a categorical latent variable $X$. The classical formulation of the unrestricted latent class model is:

$$\pi_{xabcd} \;=\; \pi_x \, \pi_{a|x} \, \pi_{b|x} \, \pi_{c|x} \, \pi_{d|x} \, . \tag{5.1}$$

Usually, $\pi_x$ is called a latent probability, while the conditional probabilities – $\pi_{a|x}$, $\pi_{b|x}$, $\pi_{c|x}$, and $\pi_{d|x}$ – are called conditional response probabilities.

As can be seen from equation 5.1, the manifest variables are assumed to be independent of one another within the levels the latent variables. This is called the assumption of local independence. As demonstrated by Haberman (1979), the same unrestricted LCM model can also be formulated as a log-linear model for the incomplete frequency table $m_{xabcd}$, that is, as

$$\log m_{xabcd} \;=\; u + u^X_x + u^A_a + u^B_b + u^C_c + u^D_d + u^{XA}_{xa} + u^{XB}_{xb} + u^{XC}_{xc} + u^{XD}_{xd} \, . \tag{5.2}$$

The relationship between the two formulation of the LCM can be illustrated by writing the conditional probabilities in equation 5.1 as a function of the log-linear parameters appearing in equation 5.2 (Haberman, 1979; Heinen, 1996). For instance,

$$\pi_{a|x} \;=\; \frac{\exp\left(u^A_a + u^{XA}_{xa}\right)}{\sum_a \exp\left(u^A_a + u^{XA}_{xa}\right)}, \tag{5.3}$$

which is a saturated logit model for the probability on $A$ given $X$.

In $\ell_{EM}$, both formulations of the latent class model can be used. The classical formulation is, however, computational more efficient because it breaks down the problem in a number smaller problems. For instance, in case of the above example, instead of working with a five-way table ($m_{xabcd}$), one works with a one-way table ($\pi_x$) and four two-way tables ($\pi_{a|x}$, $\pi_{b|x}$, $\pi_{c|x}$, and $\pi_{d|x}$). And, if one is interested in the log-linear parameters, one may parameterize the conditional probabilities using a logit model as described in equation 5.3.

A latent class model for a four-way observed table `ABCD` can be specified with $\ell_{EM}$ as

[1] Textbooks which deal with LCMs are Goodman (1978), Haberman (1979), McCutcheon (1987), Hagenaars (1990), and Vermunt (1997).

```
* example 5.1a: unrestricted LCM
lat 1
man 4
dim 2 2 2 2 2
mod X A|X B|X C|X D|X
dat [59 56 14 36 7 15 4 23
     75 161 22 115 8 68 22 123]
```

Here, '`lat 1`' indicates that the model contains one latent variable, which default label is '`X`'. The first number after `dim` specifies the number of latent classes, the last four numbers the dimensions of the four manifest variables. With `mod` we can specify the probability structure which was described in equation 5.1. As can be seen, the latent class model is just a path model in which one of the variables is unobserved. The data specified after `fre` consists of the observed four-way frequency table `ABCD`.

The log-linear latent class model described in equation 5.2 is obtained by replacing the model specification by

```
mod {XA,XB,XC,XD}
```

As can be seen, we simply specify a hierarchical log-linear model for the incomplete frequency table `XABCD`. The two parameterizations can be combined by specifying the model as

```
mod X {X} A|X {XA} B|X {XB} C|X {XC} D|X {XD}
```

Here, the probability structure of the classical LCM is combined with the logit parameterization of described in equation 5.3.[2]. This is the same as in Formann's (1992) linear-logistic LCM.

## 5.2 Restricted latent class models

In fact, any type of specification that can be used in models for completely observed tables can also be used when there are latent variables. We can use `eq1` and `eq2` to impose equality and fixed-value restrictions on the probabilities of latent class models, and `or1` and `or2` to impose ordinal restrictions (see section 4.3). User-defined designs, predefined designs, and association models can be used to further restrict the log-linear parameter of latent class models. And finally, cumulative link functions can be used to specify latent class models for ordinal items. It should be noted that – although in practice it will not often be useful – it is even possible to use different types of restrictions, or parameterizations, for the various conditional response probabilities of a LCM.

### 5.2.1 Equality and fixed-value restrictions on probabilities

The simplest type of equality restriction in the context of latent class analysis is assuming that two sets of conditional response probabilities are equal to one another. For example, we could assume that $\pi_{a|x} = \pi_{b|x}$ and $\pi_{c|x} = \pi_{d|x}$. With the command `eq1`, we can specify such a restricted LCM as follows:

```
* example 5.1b: restricted LCM via eq1
lat 1
man 4
dim 2 2 2 2 2
mod X
```

[2]It should be noted that, in fact, it is not necessary to specify the saturated models after the probabilities. If nothing is specified, $\ell EM$ will assume a saturated log-linear or logit model for the probability concerned.

```
    A|X
    B|X eq1 A|X
    C|X
    D|X eq1 C|X
dat ex51.fre
```

As can be seen, $\pi_{b|x}$ is made equal to $\pi_{a|x}$ by putting the statement `eq1 A|X` after `B|X`. In addition, $\pi_{d|x}$ is made equal to $\pi_{c|x}$ by specifying `eq1 C|X` behind `D|X`.

More complicated equality and fixed-value restrictions can be imposed by means of the very flexible command `eq2`. A fictive example of the use of this command is

```
* example 5.1c: restricted LCM via eq2
lat 1
man 4
dim 2 2 2 2 2
mod X
    A|X eq2
    B|X eq2
    C|X eq2
    D|X eq2
des [1 0 0 2   * A|X
     2 0 0 1   * B|X
     3 0 0 -1  * C|X
     -1 0 0 3] * D|X
dat ex51.fre
sta C|X [.5 .5 .9 .1]
sta D|X [.1 .9 .5 .5]
```

Here, particular probabilities are assumed to be equal to one another while other ones are fixed to a specific value. First, we specify `eq2` after the set of probabilities for which we want to specify equality and fixed-value restrictions. Then, with a type of design matrix which is specified with `des`, it is indicated which probabilities are free (a zero), equal (equal positive numbers), and fixed (a minus one). And finally, the values for the fixed probabilities are specified as starting values, that is, by means of the command `sta`. In the example, the probability of $A = 1$ for class 1 equals the probability of $B = 2$ for class 2, the probability of $A = 2$ for class 2 equals the probability of $B = 1$ for class 1, and the probability of $C = 1$ for class 1 equals the probability of $D = 2$ for class 2. In addition The probability of $C = 2$ for $X = 2$ and the probability of $D = 1$ for $X = 1$ are fixed to .1. [3]

An interesting type of application of equality restrictions on probabilities is the specification of probabilistic Guttman scales (Proctor, 1970; McCutcheon, 1987). Assume that we have 4 items which can be ordered with respect to their difficulty. Item `A` is the easiest item and item `D` is the most difficult one. One of the probabilistic Guttman models is the Proctor model, which can be specified as follows

```
* example 5.1d: Proctor model
lat 1
man 4
dim 5 2 2 2 2
mod X
```

[3] Mooijaart and van der Heijden (1992) gave the likelihood equations to be solved in the M step of the EM algorithm for LCMs with these general types of equality and fix-value restrictions. $\ell EM$ solves these equations by means of the uni-dimensional Newton algorithm (Vermunt, 1997).

```
    A|X eq2
    B|X eq2
    C|X eq2
    D|X eq2
dat ex51.dat
des [1 0 0 1 0 1 0 1 0 1
     1 0 1 0 0 1 0 1 0 1
     1 0 1 0 1 0 0 1 0 1
     1 0 1 0 1 0 1 0 0 1]
```

As can be seen, we have a model with 5 latent classes, which are the five Guttman types. The error probabilities, which denote the probabilities of giving incorrect answers given the scale type to which one belongs, are assumed to be equal across items and categories.

A Proctor model with item-specific errors is obtained by replacing the design matrix with constraints by

```
des [1 0 0 1 0 1 0 1 0 1
     2 0 2 0 0 2 0 2 0 2
     3 0 3 0 3 0 0 3 0 3
     4 0 4 0 4 0 4 0 0 4]
```

Another even less restrictive probabilistic Guttman model is the latent distance model, which is obtained by the following design matrix:

```
des [1 0 0 1 0 1 0 1 0 1
     2 0 2 0 0 3 0 3 0 3
     4 0 4 0 4 0 0 5 0 5
     6 0 6 0 6 0 6 0 0 6]
```

In this model, the error rates are item specific. In addition, they are category specific, except for the easiest and the most difficult item.

### 5.2.2 Restrictions on log-linear parameters

User-defined and predefined designs can, among other things, be used to specify equality restrictions among the two-variable interaction terms of different indicators:

```
* example 5.1e: LCM with equal two-variable effects
lat 1
man 4
dim 2 2 2 2 2
mod X {X} A|X {A} B|X {B} C|X {C} D|X {D}
all {spe(XA,XB,XC,XD,1a)}
dat ex51.fre
```

In section 4.2, it was already explained how to restrict log-linear parameter across submodels via the command `all`. With the predefined design type `1a`, which generates simple log-linear effects, it is indicated that the two-variable interactions are equal across items. As is explained below, this restricted LCM is, in fact, a discretized variant of the well-known Rasch model (Heinen, 1996; Lindsay, Clogg, and Grego, 1991).

The probabilities Guttman models can also be specified using a log-linear parameterization. For example, the latent distance model is obtained as follows:

```
* example 5.1f: Latent distance model
lat 1
man 4
dim 5 2 2 2 2
mod X
    A|X {}
    B|X {}
    C|X {}
    D|X {}
all {fac(AX,BX,CX,DX,6)}
dat ex51.dat
des [1 0 0 1 0 1 0 1 0 1
     2 0 2 0 0 3 0 3 0 3
     4 0 4 0 4 0 0 5 0 5
     6 0 6 0 6 0 6 0 0 6]
```

Thus, rather than directly restricting the error probabilities using `eq2`, we restrict them by means of a set of log-linear restrictions specified via the command `fac(..)`.

### 5.2.3 Ordinal indicators

Suppose we want to specify a restricted latent class model for 5 items, each having 3 ordered categories. With $\ell_{EM}$, various types of restricted LCMs can be formulated for such polytomous ordered items. One may use a priori zeros, non-parametric ordinal restrictions (see equations 4.2 and 4.3), log-linear association structures, log-multiplicative association structures, or cumulative link functions (see section 3.4).

Clogg (1979) proposed a LCM for Likert-type items, which involves contraining some of the conditional response probabilities to zero. The easiest way to specify such a model with $\ell_{EM}$ is

```
* example 5.2a: LCM for Likert-type items
lat 1
man 5
dim 3 3 3 3 3 3
mod X A|X B|X C|X D|X E|X
dat ex52.fre
sta A|X [.7 .3 .0 .2 .6 .2 .0 .3 .7]
sta B|X [.7 .3 .0 .2 .6 .2 .0 .3 .7]
sta C|X [.7 .3 .0 .2 .6 .2 .0 .3 .7]
sta D|X [.7 .3 .0 .2 .6 .2 .0 .3 .7]
sta E|X [.7 .3 .0 .2 .6 .2 .0 .3 .7]
```

This latent class model has as many latent classes as the number of categories of the items. In addition, the conditional response probabilities are restricted in such way that the item responses correspond to value of the latent variable or an adjacent value. In this case, this involves making the (3,1) and (1,3) latent-manifest combinations structurally zero, which is accomplished by specifying zero starting values for these probabilities.[4]

[4]It should be noted that because of the structural zeros, 10 parameter will not be estimable. Therefore, to obtain the correct number of degrees of freedom, one has to add 10 to the reported number of degrees of freedom. The model can, however, also be specified in such a way that one does not have this problem. One option is to use the `eq2` command to indicate which probabilities are fixed to zero. Another possibility is to use weight vectors to fix the wanted probabilities to zero in combination with user-defined designs to specify the log-linear effects for the non-zero probabilities.

The commands `or1` and `or2` can be used to specify non-parametric ordinal latent class models (Croon, 1990). Suppose that all 5 items are scored in the same direction and that, in addition, we want a model with 5 latent classes. Such a model can be specified by

```
* example 5.2b: non-parametric ordinal LCM
lat 1
man 5
dim 5 3 3 3 3 3
mod X
    A|X or1
    B|X or1
    C|X or1
    D|X or1
    E|X or1
dat ex52.fre
```

As can be seen, we just put the statement `or1` after each of the response probabilities. If particular items are coded in a reversed order, that is, in such a way that a negative relationship with the latent variable can be expected, we have to use `or2` instead of `or1` for the items concerned. The number of latent classes can be modified by changing the first number after the command `dim`.

Other types of ordinal models are obtained by imposing restrictions on the two-variable interaction terms in log-linear LCMs. For instance,

```
* example 5.2c: LCM with uniform association
lat 1
man 5
dim 5 3 3 3 3 3
mod X   {X}
    A|X {A,ass1(A,X,2a)}
    B|X {B,ass1(B,X,2a)}
    C|X {C,ass1(C,X,2a)}
    D|X {D,ass1(D,X,2a)}
    E|X {E,ass1(E,X,2a)}
dat ex52.fre
```

gives a uniform association structure for the relationship between latent and manifest variables.[5] It should be noted that the assumptions which underlie this model are stronger than ordinal. In fact, we treat both the latent variable and the indicators as discrete interval level variables. The above input file can easily be transformed into a row- or column association model. If we replace `2a` by `3a`, we obtain a row-association model, which means that the latent variable will be interval level and the items ordinal.[6] Type of model `4a` (column association) will yield a model in which the items are treated as discrete interval variables and the latent variable as nominal.[7]

Also the log-multiplicative association model can be useful in the context of LCMs for ordinal indicators. For instance,

[5] Rather than using `ass1(..)`, one could also specify the uniform association model with a user-defined design (`cov(..)` and `des`) or with predefined design (`spe(..)`) type `1b`.

[6] In fact, it a nominal model in the items because there is no guarantee that row parameters are ordered. In practice, we can, however, expect them to be ordered.

[7] If assume the column parameters to be equal across items, we get an ordinal model in the latent variable. This can be accomplished by imposing restrictions among submodels with the command `all`.

```
* example 5.2d: LCM with log-multiplicative RC association
lat 1
man 5
dim 5 3 3 3 3 3
mod X {X} A|X {A} B|X {B} C|X {C} D|X {D} E|X {E}
all {ass2(A,X,5a,a),ass2(B,X,5a,a),ass2(C,X,5a,a),
     ass2(D,X,5a,a),ass2(E,X,5a,a))}
dat ex52.fre
```

defines a latent class model in which both the scores of the categories of items and the scores of the latent classes are estimated. By using the `ass2(..)` terms after '`all`', it is possible to restrict the scores of the latent classes to be equal across items. The last parameter in the `ass2(..)` statements is the `<type of symmetry>` parameter. Its value `a` indicates that the row and column scores are equal to the scores of the same variable in other partial associations.[8]

The fourth method for defining LCMs for ordinal items is the use of cumulative link functions for the conditional response probabilities appearing in the basic equation 5.1. An example is

```
* example 5.2e: ordinal LCM with cumulative link functions
lat 1
man 5
dim 5 3 3 3 3 3
mod X   {X}
    A|X cum(a) {cov(X,1)}
    B|X cum(a) {cov(X,1)}
    C|X cum(a) {cov(X,1)}
    D|X cum(a) {cov(X,1)}
    E|X cum(a) {cov(X,1)}
des [-2 -1 0 1 -2
     -2 -1 0 1 -2
     -2 -1 0 1 -2
     -2 -1 0 1 -2
     -2 -1 0 1 -2]
dat ex52.fre
```

Here, the effects of `X` on the items are described by means of a set of cumulative logit models, in which `X` is treated as an interval level variable. Although not demonstrated here, we could modify this input in such a way that `X` is either nominal or ordinal. In addition, the cumulative logit link could be replaced by another link function. For instance, if we change `cum(a)` into `cum(b)`, we get a latent class model in which the conditional response probabilities are restricted by means of probit models.

### 5.2.4 Latent trait models

Latent trait models are measurement models in which a continuous latent variable is assumed to determine the individuals' responses on a set of categorical, dichotomous or polytomous, items. The relationship between the latent variable and the items is described by means of a logistic (logit) or normal ogive (probit) model.

As demonstrated by Heinen (1996), latent trait models are strongly related to LCMs. Actually, they are so strongly related that by means of $\ell_{EM}$ it is possible to obtain marginal

[8] Although most researchers will call this LCM ordinal, there is no guarantee that the scores of the categories of the items are ordered. If this is not the case for one of the items, we can, for instance, restrict the scores of categories which have the wrong order to be equal.

maximum likelihood (MML) estimates of the parameters of latent trait models assuming either a parametric, partially semi-parametric, or fully semi-parametric representation of the distribution of the latent trait variable.[9] Parametric means that the distribution of the latent variable is assumed to be known, for instance, Gaussian or uniform. In partially semi-parametric MML, the latent trait is approximated by means of discrete (latent class) variable. The scores of the categories of this discrete latent variable (the locations of the latent nodes) are fixed, but the latent distribution (the weights of the latent nodes) is treated as unknown. And finally, fully semi-parametric MML differs from semi-parametric MML in that it involves estimation of both the scores and the distribution of the discrete latent variable.

Some of the restricted LCM presented in the previous subsections are, actually, (fully) semi-parametric latent trait models: example 5.1e is a semi-parametric Rasch model, example 5.2c is a partially semi-parametric partial credit model, example 5.2d is a fully semi-parametric polytomous version of the two-parameter logistic model, and example 5.2e is a partially semi-parametric graded response model (Heinen, 1996). Small modifications of these example input files would yield other types of (fully) semi-parametric latent trait models.

Although in parametric models the latent trait is actually a continuous variable, when estimating these models, one has to approximate the assumed distribution function by means of a number of discrete points (latent nodes, quadrature points). The standard practice in programs for estimating latent trait models, such as Bilog (Mislevy and Bock, 1990) and Multilog (Thissen, 1988), is to approximate the latent distribution by means of around 10 discrete points. This means that, in fact, a restricted latent class is specified in which the latent distribution, $\pi_x$, is fixed and in which scores are assigned to the latent classes.

It will now be clear that such continuous latent variable models can also be estimated with $\ell_{EM}$. The latent distribution can be fixed with `eq2` or with a weight vector (`wei(X)`), while the a priori scoring of the categories of `X` can be accomplished by linearly restricting the interaction terms in the submodels for the conditional response probabilities. An example of such a model is

```
* example 5.3a: latent trait model with normally distributed X
lat 1
man 5
dim 9 2 2 2 2 2
mod X   {wei(X)}
    A|X {A,spe(AX,1b)}
    B|X {B,spe(BX,1b)}
    C|X {C,spe(CX,1b)}
    D|X {D,spe(DX,1b)}
    E|X {E,spe(EX,1b)}
dat ex53.fre
sta wei(X) nor(1,8)
```

yields a two-parameter logistic model for 5 dichotomous items. In this example, the latent distribution is approximated by 9 nine discrete points.[10] Since now the latent distribution must not be estimated but treated as fixed, we no longer include the first-order term `X` in the model for $\pi_x$ but indicate that a vector of weights (or a fixed effect) will be specified for $X$. The values of the entries of the weight vector are specified with `sta` (starting value). In this case,

[9] It is well-known that conditional maximum likelihood estimates of the item parameters of Rasch models can be obtained without introducing a latent variable (Mellenberg and Vijn, 1981; Kelderman, 1984). Such models, which contain a set of total score parameters, can be estimated with $\ell_{EM}$ using either `fac(..)` or the predefined design for total score parameters (`2a`).

[10] Of course, if one does not find this accurate enough, one may increase the number of latent nodes just by increasing the number of classes.

we used the special option `nor(..)` to generate starting values which are in agreement with a normal distribution. The first parameter in `nor(..)` determines the method that has to be used to approximate the normal distribution (1=rescaled density; 2=piece of the cumulative distribution) and the second the range of the normal distribution that has to be used (8 means from -4 to +4). The relationship between the latent variable and the items is restricted by means of an uniform association model specified by means of `spe(..)`.

In almost the same way, we could specify Rasch models, polytomous generalizations of the Rasch and the two-parameter logistic models, and latent trait models with cumulative link functions. In addition, we could assume other types of distributional forms for the latent trait variable.

Not only latent trait models, but also factor analytic models for ordinal items can be estimated with $\ell_{EM}$. By specifying a probit model for the conditional response probabilities, one obtains a model that is equivalent to the factor analytic model for dichotomous items (Christoffersen, 1975) or its extension to ordered polytomous items (Muthén, 1984).[11] The only difference is that in $\ell_{EM}$ the parameters are estimated by ML, while these factor analytic models are usually estimated by Generalized Least Squares. Thus,

```
* example 5.3b: factor analysis model for dichotomous items
lat 1
man 5
dim 9 2 2 2 2 2
mod X   {wei(X)}
    A|X cum(b) {spe(X,1b)}
    B|X cum(b) {spe(X,1b)}
    C|X cum(b) {spe(X,1b)}
    D|X cum(b) {spe(X,1b)}
    E|X cum(b) {spe(X,1b)}
dat ex53.fre
sta wei(X) nor(1,8)
```

gives both a factor analysis model for dichotomous items and a normal ogive latent trait model.

[11] The relationship between latent trait models with a probit link and equivalent factor analysis models, including the differences in their parameterization, is explained by Mislevy (1986) and by Takane and De Leeuw (1987).

# Chapter 6

# Path models with latent variables

In chapter 4, we demonstrated how to specify path models for categorical variables, while in chapter 5 it was explained how to define LCMs. The combination of the possibility of specifying a probability structure with the possibility of defining categorical latent variables yields quite a general type of model which can be seen as a categorical variant to the well-known Lisrel model for interval level data (Jöreskog and Sörbom, 1988). That is the reason why Hagenaars (1990, 1993) called this 'a modified Lisrel approach'. An interesting feature of this path model with latent variables is that it contains most of the extensions which are proposed for the standard LCM as special cases (Vermunt, 1996b, 1997). The best-known of these extensions are models with several latent variables, multiple group models, models with external variables, local dependence models, latent Markov models, latent budget models, and several types of finite mixture models, such as mixed logit/probit, mixed Markov, mixed Rasch, and mixed ranking models.

## 6.1 General model

To demonstrate the potentials of the modified Lisrel model which was originally proposed by Hagenaars (1990, 1993), we will first present quite an extended example which combines several features of the model, that is,

```
* example 6.1: modified Lisrel model
lat 2
man 11
dim 2 2 2 2 3 2 2 2 2 2 2 2 2
lab W Y R S T A B C D E F G H
mod RST
    W|R   {RW}
    Y|WST {WY,STY}
    A|W   {A}
    B|W   {B}
    C|W   {C}
    D|W   {D}
    E|Y   eq1 A|W
    F|Y   eq1 B|W
    G|Y   eq1 C|W
    H|Y   eq1 D|W
all {spe(AW,BW,CW,DW,1a)}
dat ex61.fre
```

Here, R, S, and T are exogenous variables, W and Y are latent variables, A-D serve as indicators for W, and E-H serve as indicators for Y. In fact, it is a model for a two-wave panel, in which the exogenous variables are measured at the first point in time and the items at both the first and second point in time. The exogenous variables are used to explain both a person's latent state at the first point in time and the transitions that occur between the latent state at the first point in time and the latent state at the second point in time. The measurement model for W and Y is quite parsimonious, that is, it is assumed to have a Rasch-type structure (equal two-variable interactions) and it is assumed to be time homogeneous.

This was quite a complicated example. The next sections present extensions of the standard LCM which are special cases of the general path model with latent variables.

## 6.2 Models with several latent variables

A latent class model with two latent variables denoted by $W$ and $Y$, each having two indicators, is defined as (Goodman, 1974)

$$\pi_{wyabcd} \quad = \quad \pi_{wy}\ \pi_{a|w}\ \pi_{b|w}\ \pi_{c|y}\ \pi_{d|y}\ .$$

In $\ell_{EM}$, such a model can be specified by

```
* example 6.2: LCM with 2 latent variables
lat 2
man 4
dim 2 2 2 2 2 2
lab W Y A B C D
mod YW A|W B|W C|Y D|Y
dat ex62.fre
```

As can be seen, the probability structure is used to indicate that each of the latent variables has its own set of indicators.

Although in this case it is not relevant because the latent variables are both dichotomous, the possibility to specify a log-linear model for the marginal table WY make it possible to further restrict the relationship between the latent variables. Hagenaars (1986), for instance, proposed symmetry and quasi-symmetry models for the associations between the latent variables. If we now assume that we work with trichotomies rather that dichotomies and that, in addition, the items concern measurements at two point in time, a latent symmetry model could be specified as

```
* example 6.3: LCM symmetry model
lat 2
man 4
dim 3 3 3 3 3 3
lab W Y A B C D
mod YW {spe(YW,3a)}
    A|W
    B|W
    C|Y eq1 A|W
    D|Y eq1 B|W
dat ex63.fre
```

The symmetric structure is defined by the predefined design for symmetry. In addition, the measurement part of the model is assumed to be equal for the two time points.

## 6.3 Multiple group models

In multiple groups analysis, both the latent distribution and the conditional response probabilities are allowed to differ among subgroups. The multiple group LCM for a set of three items is defined as (Clogg and Goodman, 1985; McCutcheon, 1988)

$$\pi_{wabcg} \quad = \quad \pi_g \, \pi_{w|g} \, \pi_{a|wg} \, \pi_{b|wg} \, \pi_{c|wg} \, .$$

As can be seen, the model just involves including an additional variable (indicating group membership) into the model (Hagenaars, 1990), which in this case is denoted by $G$. This multiple-group model can be defined in $\ell_{EM}$ by

```
* example 6.4: multiple-group LCM
lat 1
man 4
dim 2 3 2 2 2
lab W G A B C
mod G W|G A|WG B|WG C|WG
dat ex64.fre
```

Here, both the latent distribution and the conditional response probabilities differ across levels of `G`, which is sometimes called a completely heterogenous model. A model in which the measurement model equals among levels of `G` is obtained by replacing the above model specification by

```
mod G W|G A|W B|W C|W
```

There are, in addition, all types of intermediate or partially heterogenous specifications. One could, for instance, specify a model in which particular conditional response probabilities depend on `G` and others not. Another possibility is to restrict the conditional response probabilities by means of no-three-variable interaction models.

## 6.4 Models with external variables

LCMs with external variables (Goodman, 1974; Clogg, 1981; Hagenaars, 1990) are similar to multiple group models. The main differences between these two models are that in models with external variables there may be more than one variable that influences the latent distribution and that the external variables do not influence the conditional response probabilities.[1] A LCM with three external variables $R$, $S$, and $T$ could be defined as

$$\pi_{wrstabc} \quad = \quad \pi_{rst} \, \pi_{w|rst} \, \pi_{a|w} \, \pi_{b|w} \, \pi_{c|w} \, ,$$

where $\pi_{w|rst}$ may be further restricted by a logit model. With $\ell_{EM}$, such as model could be specified as

```
* example 6.5: LCM with 3 external variables
lat 1
man 6
dim 2 2 2 3 2 2 2
lab W R S T A B C
mod RST
```

[1] Another extension of the standard LCM, the latent budget model (Van der Heijden, Mooijaart, and De Leeuw, 1992), is very similar to the LCM model with external variables as it is defined here.

```
    W|RST {WRS,WT}
    A|W
    B|W
    C|W
dat ex65.fre
```

Besides a measurement model for W, this model contains a logit model for the relationship between the external variables and the latent variable W. Note that the specification of the fixed margin RST could also be omitted from the model specification.

It is also possible to use continuous external variables in a LCM (Dayton and Macready, 1988; Van der Heijden and Dessens, 1994). The use of continuous covariates in $\ell_{EM}$ was already explained in subsection 3.3. In almost the same way as they were used there in a multinomial logit model, they can also be used in a LCM. A LCM with two continuous covariates and three indicators can be specify as follows:

```
* example 6.6: LCM with 2 continuous covariates
lat 1
man 3
con 2
dim 2 2 2 2
lab W A B C
mod W|x {W,cov(x,1,W,c,-1),cov(x,2,W,c,-1)}
    A|W
    B|W
    C|W
rec 221
des [1 -1
     1 -1]
dat ex66.dat
```

The data must now be in record format, where the first three columns contain the individuals' scores on the indicators and the last two columns on the continuous external variables.

It is also possible to specify latent trait models with external variables. There are two different ways of incorporating external variables in latent trait models. The first method involves, as in the above LCMs, specifying a regression model for the latent variable. The second method substitutes the effects of the latent variable on the items by effects of the external variables and an error term (Zwinderman, 1991). Using the latter method, we could specify a Rasch model with two external variables as follows:

```
* example 6.7a: Rasch model with external variables
lat 1
man 5
dim 9 2 2 2 2 2
lab W R S A B C
mod RS
    W     {wei(W)}
    A|WRS {A}
    B|WRS {B}
    C|WRS {C}
all {spe(AW,BW,CW,1b),spe(AR,BR,CR,1a),spe(AS,BS,CS,1a)}
dat ex67.fre
sta wei(W) nor(1,8)
```

Here, W no longer serves as a latent variable, but as a normally distributed error term in the linear regression of the latent variable on the external variables. The effects of R and S on the items have to be interpreted as effects of these variables on the latent trait variable.

The other way of including external variables in latent trait models or factor analysis models for ordered items is

```
* example 6.7b: factor analysis model with external variables
lat 1
man 5
dim 9 2 2 2 2 2
lab W R S A B C
mod RS
    W|RS cum(f) {spe(R,1a),spe(S,1a)}
    A|W  cum(b) {spe(W,1b)}
    B|W  cum(b) {spe(W,1b)}
    C|W  cum(b) {spe(W,1b)}
dat ex67.fre
```

Here, the latent variable W is related to the external variables by means of a restricted probit model (cum(f)), that is, a probit model with equidistant thresholds (see equation 3.5. This is a method to specify the that latent variable W follows a conditional normal distribution given the external variables.

## 6.5 Local dependence models

While in the standard LCM the items are assumed to be conditionally independent of one another, it not a problem to relax this assumption. This leads to what is called a local dependence model (Hagenaars, 1988). Suppose that we want to modify the latent class model described in equation 5.1 by allowing for a direct relationship between $C$ and $D$. One option, in which we make no decision about the causal order between $C$ and $D$, is to specify a model of the form

$$\pi_{xabcd} \quad = \quad \pi_x \; \pi_{a|x} \; \pi_{b|x} \; \pi_{cd|x} \; .$$

On the other hand, if $D$ can be assumed to be posterior to $C$, a specification of the form

$$\pi_{xabcd} \quad = \quad \pi_x \; \pi_{a|x} \; \pi_{b|x} \; \pi_{c|x} \; \pi_{d|cx} \; .$$

is more appropriate. The former specification is obtained as follows:

```
* example 6.8: local dependence models
lat 1
man 4
dim 2 2 2 2 2
mod X A|X B|X CD|X
dat ex68.fre
```

while in the latter specification, the model specification is of the form

```
mod X A|X B|X C|X D|CX
```

Of course, it is also possible to further restrict the probabilities $\pi_{cd|x}$ and $\pi_{d|cx}$, for instance, by a no-three-variable interaction model.

## 6.6 Latent Markov models

Another special case of the path model with latent variables is the latent Markov model (Wiggins, 1973; Poulsen, 1982; Van de Pol and De Leeuw, 1986; and Van de Pol and Langeheine, 1990). Suppose we have observations of the same variable at four occasions. A latent Markov model for such a situation could be defined as

$$\pi_{wxyzabcd} \;=\; \pi_w \, \pi_{x|w} \, \pi_{y|x} \, \pi_{z|y} \, \pi_{a|w} \, \pi_{b|x} \, \pi_{c|y} \, \pi_{d|z} \, .$$

Here, $W$-$Z$ are latent (state) variables at the four points in time; $A$-$D$ are manifest variables which serve as indicators for these latent variables. With $\ell_{EM}$, such a model can be specified by

```
* example 6.9: latent Markov model
lat 4
man 4
dim 2 2 2 2 2 2 2 2
lab W X Y Z A B C D
mod W X|W Y|X Z|Y
    A|W B|X eq1 A|W C|Y eq1 A|W D|Z eq1 A|W
dat ex69.fre
```

As can be seen, we imposed the additional restriction that the measurement model is time-homogeneous, which is one of the possible identifying restrictions.

There are many extension of the simple latent Markov model described above, such as models with more than one indicator per occasion and multiple group models (Van de Pol and Langeheine, 1990), models with external variables (Vermunt, Langeheine, and Böckenholt, 1995), and models with more complicated measurement parts (Vermunt and Georg, 1995). All these extensions can be dealt with within the general framework of path modeling with latent variables.

## 6.7 Mixture models

The last rather broad class of models that can be seen as special cases of the path model with latent variables are finite mixtures of multinomial distributions. Suppose that there are four observed variables $A$, $B$, $C$, and $D$ and a mixture variable $X$. In its most general form, a finite mixture of multinomial distributions is given by

$$\pi_{abcd} \;=\; \sum_x \pi_x \, \pi_{abcd|x} \, .$$

The type of mixture model that is obtained depends, of course, on the restrictions that are imposed on $\pi_{abcd|x}$. For instance, assuming independence between $A$, $B$, $C$, and $D$ yields the LCM. On the other hand, assuming that

$$\pi_{abcd|x} = \pi_{abc} \, \pi_{d|xabc}$$

in combination with a logit or probit parameterization of $\pi_{d|xabc}$ gives a mixed logit or probit model (Kamakura, Wedel, and Agrawal, 1992; Formann, 1992). An example of a mixed logit model for dependent variable `D` is

```
* example 6.10: mixed logit model
lat 1
man 4
```

```
dim 2 2 2 2 2
lab X A B C D
mod X ABC
    D|XABC {DX,DA,DB,DC}
dat ex610.fre
```

As can be seen, `X` is assumed not to be related with the observed covariates `A`, `B`, and `C`. Although this is not necessary within the context of path models with latent variables, it is an assumption that is generally made in mixed discrete choice models. Replacing the model by

```
mod X ABC
    D|XABC cum(b) {spe(X,1a),spe(A,1a),spe(B,1a),spe(C,1a)}
```

yields a mixed probit model.

Mixed Markov models (Poulsen, 1982, Langeheine and Van de Pol, 1990, 1994) are obtained by setting

$$\pi_{abcd|x} = \pi_{a|x}\,\pi_{b|ax}\,\pi_{c|bx}\,\pi_{d|cx}\,.$$

Such a model can be specified with $\ell_{EM}$ by

```
* example 6.11: mixed Markov model
lat 1
man 4
dim 2 2 2 2 2
lab X A B C D
mod X A|X B|AX C|BX D|CX
dat ex611.fre
```

By means of the command `eq1` it is easy to transform this model into a stationary mixed Markov model.

Rost (1990) proposed a mixed Rasch model. This model involves estimating the Rasch model by conditional maximum likelihood, where both the total-score parameters and item difficulties are allowed to vary among latent classes. With $\ell_{EM}$ it can be specified as

```
* example 6.12: mixed Rasch model
lat 1
man 4
dim 2 2 2 2 2
lab X A B C D
mod X ABCD|X {AX,BX,CX,DX,spe(ABCD,2a)}
dat ex612.fre
```

Besides the two-variable interactions between the items and the latent variable, this model includes a set total-score parameters. [2]

With $\ell_{EM}$, it is also possible to specify the mixed ranking models proposed by Croon (1989) and Croon and Luijkx (1993). Assume that we have information on the ranking of three objects, where variable $A$ indicates the object of first choice, $B$ the second, and $C$ the third.[3] A mixed Bradley-Terry-Luce (BTL) model can be specified as follows

[2] It should be noted that one additional restrictions have to be imposed to identify all model parameters. For instance, we could set the difficulty of `D` equal to zero for one latent class.

[3] Note that here the rankings are the variables and the objects are the levels of these variables. So, $B = 3$ means that second choice is object number three.

```
* example 6.13a: mixed ranking model (BTL model)
lat 1
man 3
dim 2 4 4 4
lab X A B C
mod X
    A|X   {}
    B|AX  {wei(AB)}
    C|ABX {wei(AC),wei(BC)}
all {spe(A,B,C,1a,X,c)}
sta wei(AB) [0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0]
sta wei(AC) [0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0]
sta wei(BC) [0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0]
dat ex613.fre
```

As can be seen, we estimate parameters for the categories of `A`, `B`, and `C` for each level of `X`. These are the class-specific utilities of the objects. The structural zeros which are inherent to ranking data – each particular object can be chosen just ones – is dealt with by a set of weight vectors. Note that the weight vectors contain zeros on the main diagonal.

Another somewhat different model for ranking data is the Pendergrass-Bradley (PB) model.[4] A mixed variant of this model can be specified as follows:

```
* example 6.13b: mixed ranking model (PB model)
lat 1
man 3
dim 2 4 4 4
lab X A B C
mod X
    ABC|X {spe(ABC,7a,X,c),wei(AB),wei(AC),wei(BC)}
sta wei(AB) [0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0]
sta wei(AC) [0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0]
sta wei(BC) [0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0]
dat ex613.fre
```

Pre-designed type '`7a`' will generate the correct design for this situation. The parameters are again the utilities of the objects for each latent class.

[4]The PB model is based on the assumption that a ranking stems from all possible paired comparisons, while the BTL model is based on the assumption that ranking is a sequential choice process.

# Chapter 7

# Dealing with partially missing data

Quite often it happens that information is missing on one or more of variables for some individuals. The $\ell_{EM}$ program allows the user to include such cases with partially missing information in the analysis. Moreover, the definition of response indicators makes it possible to specify models for the mechanism causing the missing data, sometimes also called models for nonresponse (Fay, 1986; Baker and Laird, 1988).

## 7.1 Using partially missing data

Suppose we want specify a model for three-way table $ABC$. The variables, $B$ and $C$ are, however, not observed for all individuals. More precisely, one may have missing information on $B$, $C$, or both $B$ and $C$. As a result, we can construct four different types of observed frequency tables belonging to the four different subgroups of individuals for whom we have the same type of information, that is, $ABC$, $AB$, $AC$, and $A$.

Estimation of a log-linear model for the three-way table $ABC$ using the information of the above-mentioned four subgroups can be accomplished by the following input file:

```
* example 7.1: Using partially missing data
man 3
res 1                        * one response indicator
dim 4 2 2 2                  * with four levels
lab R A B C                  * and label R
sub ABC AB AC A              * defines these four subgroups
mod ABC {AB,BC}
    R   {R}
dat [50 90 31 9 31 10 3 4    * subgroup ABC
     12 19 45 5              * subgroup AB
     26 40 11 23             * subgroup AC
     10 4]                   * subgroup A
```

Here, the option to define response indicators is used to specify that there are four different observed frequency tables. More precisely, we declared one response indicator (`res 1`) with four levels (with `dim`). With the command `sub` (subgroups) it is specified which variables are observed for each of the four subgroups. The data specified with `fre` consist of the four different observed frequency tables.

In the model specification, we used the response indicator `R` as one of the variables. In fact, we specified the simplest model for nonresponse, that is, a missing completely at random (MCAR) response mechanism (see section 7.3). In addition, we specified log-linear model `{AB,BC}` for table `ABC`. It should be noted that the likelihood-ratio statistic yields a simultaneous test for the

model of interest and the MCAR assumption. A correct test for the model that is postulated for table `ABC` – in this case `{AB,BC}` – can, however, be obtained by means of a conditional likelihood-ratio test between this model and the saturated model `{ABC}` (Hagenaars, 1990; Vermunt, 1996a, 1996b, 1997).[1]

The second example of a model which is estimated using partially missing information concerns a latent class model[2] with four items:

```
* example 7.2: LCM with missing information
lat 1
man 4
res 1
dim 5 2 2 2 2 2
lab R X A B C D
sub ABCD ABC ABD ACD BCD
mod X A|X B|X C|X D|X R
dat ex72.fre
```

As can be seen, besides persons with completely observed data, there are also persons for which information on one of the four items is missing. Note that in list of variables after `dim` and `lab`, the response indicators precede the latent variables.

## 7.2 Record format data

The computation of the different observed frequency tables which are requested as data may be tedious. It is, however, also possible to circumvent this by using record format data. In that case, $\ell_{EM}$ will compute the various frequency tables from the individual records. With a missing value code, which default value is 0, it is indicated that a particular variable is missing:

```
* example 7.3: partially missing record format data
man 3
res 1
dim 4 2 2 2
lab R A B C
sub ABC AB AC A
mod ABC {AB,BC}
    R   {R}
rec 1230
mis 9
dat ex73.dat
```

The only thing we have changed compared to example 7.1 is that we specified the number of records (`rec 1230`) and a missing-data value (`mis 9`).[3]

An additional feature that can be used when the data is in the form of individual records is the possibility to omit the specification of the subgroups. In that case, the program will find out which subgroups there are in the data. This can save a lot of work in situations in which there are many different subgroups. If there are more subgroups in the data file than the specified number of levels the response indicator, the program will give an error message.

[1] Note that the likelihood-ratio statistic for the saturated model tests only the MCAR assumption.

[2] Technical details on the estimation of models with both latent variables and partially observed data can be found in Vermunt (1996b, 1997).

[3] Note that the missing-data value must be the same for all variables.

## 7.3 Ignorable and nonignorable models for nonresponse

So far, we assumed that we just want to use the missing data without worrying about the precise mechanism causing the missing data. The possibility to define response indicators and to use them in the log-linear path model can, however, also be used to specify models for nonresponse (Fay, 1986; Hagenaars, 1990; Vermunt, 1996a, 1996b, 1997).

In the above examples on partially missing data we assumed that the missing data is missing completely at random (MCAR), which is the strongest possible assumption about the response mechanism. For the estimation of the structural parameters, this may, however, be no problem since each ignorable response mechanism will yield the same parameter estimates. A response mechanism is called ignorable if for each individual the probability of not observing the variables which are currently missing is independent of the value of the variables that are missing. For a more precise definition of ignorable response mechanisms, see, for instance, Little and Rubin (1987) or Vermunt (1996b, 1997).

Suppose we would like to modify example 7.1 by adding a model for the probability of observing $B$ and $C$. In that case, it is more appropriate to work with two response indicators: one indicating whether $B$ is missing and one indicating whether $C$ is missing. An ignorable response model could be of the form

```
* example 7.4: Modeling the response mechanisms
man 3
res 2
dim 2 2 2 2 2
lab R S A B C
sub ABC AB AC A
mod ABC    {AB,BC}
    R|ABC  {AR}
    S|RABC {RS,AS}
dat ex74.fre
```

Like in example 7.1, we have four (in this case, 2 times 2) subgroups. Contrary to example 7.1, however, we used 2 response indicators, one indicating whether `B` is missing and another whether `C` is missing. Now the order in which the subgroups are specified is crucial because that defines the meaning of the response indicators. The order of the subgroups is such that missingness on `C` changes before missingness on `B`. Since `S` changes its values before `R`, `S` will indicate whether `C` is observed and `R` whether `B` is observed. More precisely, `RS=11` is subgroup `ABC`, `RS=12` is subgroup `AB`, `RS=21` is subgroup `AC`, and `RS=22` is subgroup `A`;

The response model in example 7.4 assumes an ignorable response mechanism: both `R` and `S` depend only on `A`, which is a variable which is observed for all persons. Two examples of nonignorable response mechanism are:

```
mod ABC    {AB,BC}
    R|ABC  {AR,CR}
    S|RABC {RS,AS,BS}
```

and

```
mod ABC    {AB,BC}
    R|ABC  {AR,BR}
    S|RABC {RS,AS,CS}
```

In the former specification, the response indicators do not depend on the variable which missingness they indicate, while in the latter specification they do. Both models specify, however,

a nonignorable response mechanism because for some persons the values of response indicators depend on variables which are missing (Vermunt, 1996b, 1997).

An ignorable response mechanism which uses all additional degrees of freedom obtained from including observed tables $AB$, $AC$ and $A$ in the analysis can be specified by means of eq2. Such a model, which may also be called a saturated missing at random model (Vermunt 1996b, 1997), is obtained with a model of the form

```
mod ABC    {AB,BC}
    RS|ABC  eq2
```

in combination with design matrix

```
des [0 0 0 0 0 0 0 0
     1 1 2 2 3 3 4 4
     4 5 4 5 6 7 6 7
     8 8 8 8 9 9 9 9]
```

Here, it is specified that the probability of observing both $B$ and $C$ depends on all three variables, of observing $B$ and not observing $C$ on $A$ and $B$, of not observing $B$ and observing $C$ on $A$ and $C$, and of neither observing $B$ nor $C$ on $A$.

# Chapter 8

# Event history analysis

Besides for the estimation of log-linear models with latent variables and other types of missing data, the $\ell_{EM}$ program can be used for specifying event history models. Two types of event history models are implemented, namely, piecewise exponential survival models for continuous-time data (Laird and Olivier, 1981), also known as log-rate models, and logit models for discrete-time data (Allison, 1982). The event history model may be a model for a single nonrepeatable event, for competing risks, for repeatable events or other types of clustered observations, or for multiple-state processes. In addition, it is possible to specify models with nonparametric unobserved heterogeneity, latent covariates, and partially missing covariates, and, in a discrete-time framework, with latent and partially missing states (see Vermunt, 1996b, 1997).[1]

## 8.1 Log-rate models

Suppose we have a model for a single nonrepeatable event with two categorical covariates $A$ and $B$. Let $T$ denote the time variable and $\delta$ be a censoring indicator taking the value 0 if an individual was censored at the recorded time and 1 if an individual experienced the event of interest at the recorded time. In log-rate models, the time axis is divided into a limited number a time intervals. Within a time interval, the hazard rate is assumed to be constant, or survival to be exponential. The time variable $T$ will be used to denote the time interval in which the event or censoring occurred. A saturated log-rate model for the current situation would be of the form

$$\log h_{abt} \;=\; u + u^A_a + u^B_b + u^T_t + u^{AB}_{ab} + u^{AT}_{at} + u^{BT}_{bt} + u^{ABT}_{abt} \,,$$

where $h_{abt}$ denotes the constant hazard rate in the $t$th time interval for $A = a$ and $B = b$. The $u$ terms are the (log-linear) parameters of the hazard model. An example of non-saturated log-rate model, in which all the higher-order interactions are omitted, is

$$\log h_{abt} \;=\; u + u^A_a + u^B_b + u^T_t \,.$$

Note that this is a proportional hazard model since the covariate effects are assumed to be time independent. Besides by omitting particular terms, we can further simply this model by imposing constraints on the time dependence of the hazard rate.

The next subsections explain the specification of log-rate models with the $\ell_{EM}$ program, including all types of extension of the above model for a single nonrepeatable event.

[1] Textbooks on event history analysis are Tuma and Hannan (1984), Yamaguchi (1991), Blossfeld and Rohwer (1995), and Vermunt (1996b, 1997).

### 8.1.1 As a log-linear model with a weight vector

One of the two methods for specifying a log-rate model with $\ell_{EM}$ is as a log-linear with a weight vector (see section 2.4). In this method, which can be used with most standard programs for log-linear analysis, the observed frequency table consists of the number of events in each of the $(a, b, t)$ combinations and the weight vector of the total exposure times (Laird and Olivier, 1981; Clogg and Eliason, 1987; Vermunt, 1996b, 1997). In the case of the above example of a non-saturated log-rate model, the input file could be of the form

```
* example 8.1: log-rate model as a log-linear model with a
*               weight vector
man 3
dim 2 2 5
lab A B T
mod {A,B,T,wei(ABT)}
sta wei(ABT) ex81.wei
dat ex81.fre
```

As can be seen, it is assumed that the time variable has 5 levels and that the covariates have two levels.

It should be noted that although this input file seems to be quite simple, it is not so easy to specify a log-rate model in this way. The reason for this is that computation of the weight matrix with exposure times can be quite complicated.

### 8.1.2 As an event history model

The second method for defining log-rate models involves using the special $\ell_{EM}$ event history modeling options. When using this method, it is it is no longer necessary to supply the matrices with the observed number of occurrences and exposure times as data. The program will compute this information on the basis of the raw data. Using the commands for specifying hazard models, the same model as above could now be of the form

```
* example 8.2: log-rate model as an event history model
man 2
dim 2 2
lab A B
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {A,B,T}
rec 500
dat ex82.dat
```

With `man`, `dim`, and `lab`, we provide the necessary information on the covariates. It is also possible to specify a model for the covariates, which is an option that will be used later on when discussing models with latent and partially missing covariates.

The command `tim` is used to define the number and the begin and end points of the time intervals. In this case, we have 5 time intervals with the begin and end points which are specified between the square brackets. With `ris`, one indicates the number of states and the transitions or risks to be analyzed. Here, we have two states (0 and 1) and one type of risk, namely, the transition from state 0 to state 1.[2] Since two states with transition `[0,1]` is the default setting, this line may also be omitted from the model specification. And finally, the command `haz` is used to specify the hazard model. Within the parentheses, one can specify the model parameters

[2]It should be noted that the states are numbered from 0 to the number of states minus 1.

in the same way as in the log-linear models discussed in the previous chapters, that is, using the hierarchical log-linear notation, user-defined designs, predefined designs, and log-linear and log-multiplicative association structures (Vermunt 1996b, 1997). The label for the time variable is always `T`.

With `rec`, it is indicated that the data are now in the form of individual records. The first five lines of the data file could, for instance, be

```
1 2 10 0
2 2  5 1
2 1  4 1
1 1 20 0
2 2 15 1
etc.
```

The first two columns contain the values of the covariates `A` and `B`, the third of the time variable, and the last of the end state or the censoring indicator. As can be seen, the first and fourth case are censored, the other cases experienced the event. The program will use this information to make the occurrence and exposure matrix. It is important to note that in the calculation of the exposure times, events and censorings are assumed to occur in the middle of the indicated time unit. So, a time of 10 is changed by the program into 9.5, 5 into 4.5, etc. This default setting in the calculation of exposure times can be changed with the command `exp` (exposure time). In the default setting, it is assumed that the time and state of entry into the risk set are both 0. This can be changed by means of the command `rt0` (read time and state at entry into the risk set), which can, among other things, be used for dealing with left censored cases and time-varying covariates. The above example can easily be transformed to yield other types of specifications of the time and covariate dependence of the hazard rate. Using only one time interval yields an exponential survival model. Specifying as many time interval as times at which events occur yields a Cox's proportional hazard model. By including interactions between `T` and the covariates, one obtains nonproportional hazard models. And finally, by restricting the time-dependence, it is possible to approximate the results of parametric hazard models. For instance, a linear effect of `T` gives a Gompertz-type model, while a linear effect of log(`T`) yields a Weibull-type model (Yamaguchi, 1991).

We could approximate a Gompertz model by replacing the `tim` statement in the above example by

```
time 20
```

which means that there are twenty time intervals of length 1, and replacing the hazard model by

```
haz {A,B,spe(T,1b)}
```

Note that predefined design type `1b` yields a linearly restricted log-linear effect.

### 8.1.3 Competing risks

The above example can easily be changed into a competing-risk model. The only thing we need to do is to include an additional variable which indicates the type of event (Larson, 1984; Vermunt, 1996b, 1997). Suppose that individuals may experience one of two types of events. An example of the specification of such a model is

```
* example 8.3: competing-risk model
man 2
dim 2 2
```

```
lab A B
tim 5 [0,4,8,12,16,20]
ris 3 [0,1] [0,2]
haz {TR,AR,B}
rec 500
dat ex83.dat
```

The difference with the model for a single type of event is that now we have 3 different states (0, 1, and 2) and two possible transitions or risks, namely, the transition from 0 to 1 and from 0 to 2. The types of transitions form the levels of the so-called risk variable with label `R`. This risk variable can be used in the model specification in the same way as the other variables. The specified hazard model indicates that the time dependence and the effect of `A` differs for the two risks, while the effect of `B` is the same for both types of events.

The data set will again contain information on the value of `A`, `B`, and the time that an event or censoring occurred. The only difference is that the end state can now take on three different values rather than two: 0 (censored), 1 (event type one), or 2 (event type two).

### 8.1.4 Repeatable events

The definition of models for repeatable events involves, as in competing-risk models, specifying the events of interest using the command `ris`. In addition, the data file must contain information on the times that each of the events occurred. An example of a model for an event that may occur at most three times is

```
* example 8.4: repeatable events
man 2
dim 2 2
lab A B
tim 5 [0,4,8,12,16,20]
ris 4 [0,1] [1,2] [2,3]
haz {T,AR,B}
zer
rec 500
epi 3
dat ex84.dat
```

As can be seen, it is specified that individuals can occupy 4 different states, and that there are three possible transitions, that is, from 0 to 1, 1 to 2, and 2 to 3. These transitions denote the first, second, and third occurrence of the event of interest. Again, we have a risk variable with label `R` which can be used in the specification hazard model. In the example, it is assumed that the time dependence and the effect of `B` are the same for the first, second, and third occurrence of the event under study, while the effect of `A` differs for the three events.

The above input file contains two commands which were not yet explained: `zer` and `epi`. With `zer`, it is indicated that the time variable must be set back to zero after each event. This implies using waiting time rather than process time as the relevant time dimension. If we omit the `zer` statement, the time variable will be process time. The command `epi 3` after the specification of the number of records means that the data set contains information on the end times and states for three episodes or spells, in this case, the times and states belonging to the first, second, and third occurrence of the event of interest. The first five records of the data file could, for instance, be

```
1 2 10 1 25 2 40 3
2 2  5 1 14 2 17 2
```

```
2 1  4 1  4 1  4 1
1 1 20 0 20 0 20 0
2 2 15 1 30 2 30 2
etc.
```

The first record belongs to someone who experiences all three events. The second person experiences event 1 and 2 and is censored 3 time units after experiencing the second event (17-14). It should be noted that if the state does not change between two subsequent episodes or spells, a record is assumed to be censored. Record three is censored immediately after the first occurrence of the event of interest. The fourth case was censored after 20 time units without experiencing an event. And finally, the last record belongs to someone who is censored after the second occurrence of the event of interest.

### 8.1.5 Multiple states

Combining the competing-risk and the repeatable-event situations gives rise to what is called a multiple-state process. This is an event history which may contain different types of events which, in addition, may occur several times. Suppose that we are interested in the transitions between three states. An input file for such a situation could be

```
* example 8.5: multiple-state process
man 2
dim 2 2
lab A B
tim 4 [0,12,24,36,48]
ris 3 [0,1] [0,2] [1,0] [1,2] [2,0] [2,1]
haz {TR,AR,BR}
rec 500
epi 3
rt0
dat ex85.dat
```

As can be seen, there are six possible transition between the three states. This means that the risk variable `R` has six levels. Because the command `zer` is not included in the input file, the model uses process time as the relevant time dimension, which means that it is a (non-stationary) Markov model (Tuma and Hannan, 1984). Besides the number of records (`rec 500`) and the number of episodes per record (`epi 3`), it is specified that the records contain a starting time and state (`rt0`). The first five records of the data file could be

```
1 2 0 2 10 1 25 2 40 0
2 2 0 1  5 2 14 3 27 2
2 1 0 0  4 1  4 1  4 1
1 1 0 0 20 0 20 0 20 0
2 2 0 2 15 1 30 2 48 2
etc.
```

Columns three and four contain the starting time and state. The begin time for each of the five records is 0, while the starting state differs per record.

### 8.1.6 Multivariate hazard model

Besides by means of the `ris`, `zer`, and `epi` commands, there is another method for specifying a general class multivariate hazard models in a more compact way, that is, by the command `mult`. This method is especially useful if each of the types events can occur several times, if a simple

exponential model is assumed for the time dependence, and if the hazard rate does not depend on the number of previous occurrences of the events of interest. This method can, for instance, be used for specifying (multivariate) Poisson regression models (Böckenholt and Langeheine, 1996; Wedel et all, 1993) and stationary Markov models with covariates. An example of the use of the command `mul` is

```
* example 8.6: multivariate hazard model
man 2
dim 2 2
lab A B
tim 1 [0,1000]
mul 3
haz {AR,BR}
exp 1
rec 500
dat ex86.dat
```

By specifying that there is only one time interval, it is assumed that the hazard rate is time independent.[3] The statement `mul 3` indicates that there are three types of events. It should be noted that each of the three types of events may occur as many times as one wants. The hazard model specifies the covariate effects to be event specific. The command `exp 1` indicates that the exposure time in the reported time unit is 1 rather than .5, which means that we do not want to subtract .5 from the reported times in the data file.

The data file will contain a slightly different type of information compared to the hazard models specified so far. For each person, we have to supply, besides the covariate values, the total exposure time to and the number of occurrences of the three events of interest. The first records in the data file could, for instance, be

```
1 2 10 2  5  2  3 0
2 2 10 5  6  2  7 2
2 1 10 4  6  4  4 1
1 1  5 0 10  1 10 3
2 2  3 2 10  1  8 2
etc.
```

The first individual is exposed 10 time units to event type 1, 5 time units to event type 2, and 3 time units to event type 3. This person experiences the three types of events 2, 2, and 0 times, respectively. In the same way, we have to interpreted the event history information for the other cases.

### 8.1.7 Left censoring

The problem of left censoring is not so easy to deal with when the time of entry into the risk set in unknown. However, if there is information on the time that left censored cases enter into the risk set, dealing with left censoring just involves specifying a begin time and state (Guo, 1993; Vermunt, 1996b, 1997). An example is

```
* example 8.7: left censoring
man 2
dim 2 2
lab A B
```

[3] One time interval with a lower limit equal to zero and an upper limit equal to the largest observed survival time is the default setting.

```
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {A,B,T}
rec 500
rt0
dat ex87.dat
```

As can be seen, this is a proportional hazard model for a single nonrepeatable event. With `rt0` is it indicated that the records in the data file contain a starting time and state. The first five records of this file could be

```
1 2 0 0  9 0
2 2 4 0 10 1
2 1 0 0  4 1
1 1 5 0 16 1
2 2 0 0 20 0
etc.
```

Each of the five presented records has a starting state of 0. Records two and three are left censored since they enter into the risk set in the 4th and 5th time unit, respectively.

### 8.1.8 Fixed-effect approach to unobserved heterogeneity

One method for dealing with unobserved heterogeneity is by means of the random-effects approach which is discussed in the subsection 8.4.1. Another method is the fixed-effect approach which involves including a cluster-specific nuisance parameters in the hazard model (Yamaguchi, 1986; Vermunt, 1996b, 1997). This method only works if most of the clusters in the sample experience there is more than one observation and if one is only interested in the effects of covariates which either vary over time or over observations belonging to the same cluster.[4]

Suppose we have survival data for observations belonging to 200 clusters. The fixed-effect approach involves including a separate parameter for each cluster in the hazard model. Suppose that besides the unobserved heterogeneity component, we have two dichotomous covariates `A` and `B`. A model for such a situation could be

```
* example 8.8: fixed-effects approach to unobserved heterogeneity
man 2
dim 200 2 2
lab F A B
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {F,A,B,T}
rec 500
dat ex88.dat
```

As can be seen, the cluster-specific fixed effects are defined by including a covariate with 200 levels in the model. The data for the observations belonging to the first three clusters could be of the form:

```
1 2 2  14 1  * observation 1 in cluster 1
1 2 1  20 0  * observation 2 in cluster 1
```

[4] A cluster can be formed by a number of observations from the same individual, for instance, on repeatable events, but also by number of dependent observations from different individuals, for instance, from individuals belonging to the same family.

```
2 1 2   4 1  * observation 1 in cluster 2
3 1 1   9 0  * observation 3 in cluster 3
3 1 2  12 1  * observation 3 in cluster 3
3 2 1  10 1  * observation 3 in cluster 3
3 2 2  20 0  * observation 3 in cluster 3
etc.
```

## 8.2 Discrete-time logit models

### 8.2.1 As an event history model

In the case of discrete-time event history data, one generally regresses the transition probability at a particular time interval on a set of covariates by means of a logit model (Allison, 1982). A saturated discrete-time logit model for a single nonrepeatable event with two covariates $A$ and $B$ can be written as

$$\lambda_{abt} = \frac{\exp\left(u + u_a^A + u_b^B + u_t^T + u_{ab}^{AB} + u_{at}^{AT} + u_{bt}^{BT} + u_{abt}^{ABT}\right)}{1 + \exp\left(u + u_a^A + u_b^B + u_t^T + u_{ab}^{AB} + u_{at}^{AT} + u_{bt}^{BT} + u_{abt}^{ABT}\right)},$$

where $\lambda_{abt}$ is the probability of experiencing the event of interest in the $t$th time interval, given that one did not experience the event before. By omitting certain parameters or restricting the parameters in some other way, one can obtain more parsimonious specifications.

The definition of discrete-time logit models with $\ell_{EM}$ is similar to the specification of log-rate models. There is, however, one important difference, that is, the non-transitions have to be included in the list of risks. The covariates have to interact with the risk variable, which serves as a kind of dependent variable. This is the same as in standard logit models where the covariates interact with the dependent variable as well. An example of an $\ell_{EM}$ input file defining a discrete-time logit model is

```
* example 8.9: discrete-time logit model
man 2
dim 2 2
lab A B
dis
tim 5 [0,1,2,3,4,5]
ris 2 [0,0] [0,1]
haz {AR,BR,TR}
rec 500
dat ex89.dat
```

The command `dis` indicates that it is discrete-time logit model rather than a log-rate model. As can be seen, the `[0,0]` transition is specified to be one of the risks. In the hazard model, the covariates and time variable interact with the risk variable `R`.

Each of the generalization of the simple hazard model for a single nonrepeatable event discussed above can also be used in the context of discrete-time logit models.

### 8.2.2 As a log-linear path model

Rather than by above formulation, the discrete-time logit model can also be specified in terms of transitions between states occupied at particular points in time. This yields a formulation which is a special case of the log-linear path models discussed in chapter 4 (Vermunt, 1996b,

1997). Let $S_t$ denote the state occupied at time point $t$. The discrete-time logit model can now be written as

$$\pi_{s_t|abs_{t-1}} = \frac{\exp\left(u_{s_{t-1}s_t}^{S_{t-1}S_t} + u_{as_{t-1}s_t}^{AS_{t-1}S_t} + u_{bs_{t-1}s_t}^{BS_{t-1}S_t} + u_{abs_{t-1}s_t}^{ABS_{t-1}S_t}\right)}{\sum_{s_t}\exp\left(u_{s_{t-1}s_t}^{S_{t-1}S_t} + u_{as_{t-1}s_t}^{AS_{t-1}S_t} + u_{bs_{t-1}s_t}^{BS_{t-1}S_t} + u_{abs_{t-1}s_t}^{ABS_{t-1}S_t}\right)}$$

where $\pi_{s_t|abs_{t-1}}$ is the probability of occupying state $s_t$ at $T = t$ given that $A = a$, $B = b$, and $S_{t-1} = s_{t-1}$. To obtain the standard parameterization of discrete-time logit models, the $u$ parameters in which $S_t = S_{t-1}$ have to be set to zero. It should be noted this is a model for different types of transitions which, in addition, can occur several times. Models for a single nonrepeatable event, multiple-risk models, and models for a single type but repeatable event can be obtained by making certain transition probabilities structurally zero.

When the discrete-logit model is defined in this way, it can be specified as a log-linear path model. Suppose we have information on the states that a person occupies at five discrete points in time. Let denote these state by `G`, `H`, `I`, `J`, and `K`. An example of a model with two covariates `A` and `B` for the transition between these five time points is

```
* example 8.10a: discrete-time logit model as a log-linear
*               path model
man 7
dim 2 2 2 2 2 2 2
lab A B G H I J K
mod H|ABG {AGH}
    I|ABH {AHI}
    J|ABI {AIJ}
    K|ABJ {AJK}
all {spe(BGH,BHI,BIJ,BJK,1a)}
rec 500
dat ex810.dat
```

This is, actually, a non-stationary Markov model in which two external variables are used to explain individual differences in the transition probabilities between five time points. With `all`, it is specified that the effect of `B` on the transition probabilities is time homogeneous.

To obtain exactly the same parameterization of the parameters as in standard discrete-time logit models, one has to specify the model by means of user defined designs. The parameters have to be restricted in such a way that they indicate the main effect for and covariate effects on the transitions from state 1 to 2 and from state 2 to 1. This can, for instance, be accomplished as follows:

```
* example 8.10b: discrete-time logit model as a log-linear
*               path model, with standard parameterization
man 7
dim 2 2 2 2 2 2 2
lab A B G H I J K
mod H|ABG {fac(GH,2) fac(GH,2,A,c,-1)
    I|ABH {fac(HI,2) fac(HI,2,A,c,-1)
    J|ABI {fac(IJ,2) fac(IJ,2,A,c,-1)
    K|ABJ {fac(JK,2) fac(JK,2,A,c,-1)
all {fac(GH,HI,IJ,JK,2,B,c,-1)}
des [0 1 2 0  0 1 2 0  1 -1
     0 1 2 0  0 1 2 0  1 -1
     0 1 2 0  0 1 2 0  1 -1
```

```
     0 1 2 0  0 1 2 0  1 -1
     0 1 2 0  0 1 2 0  0 1 2 0  0 1 2 0  1 -1]
rec 500
dat ex810.dat
```

The factors specify effects for the (1,2) and (2,1) transitions, where `A` and `B` are used as grouping variables to generate the right interaction terms.

### 8.2.3 Other link functions

Rather than using a logit link to investigate the covariate effects on the transition probability, we can also use one of the other types of links function implemented in $\ell_{EM}$ that is, a probit, complementary log-log, or log-log link.

Suppose that we have a model for a single nonrepeatable event and that we have observations for 5 time points. At the first time point every one is in state 1, while at one of the next time points one can experience a transition form state 1 to state 2. The (2,1) transition is assumed to be impossible. A discrete-time probit model for this situation can be specified as follows

```
* example 8.11: discrete-time probit model
man 7
dim 2 2 2 2 2 2 2
lab A B G H I J K
mod H|ABG cum(b) {spe(A,1a) wei(GH)}
    I|ABH cum(b) {spe(A,1a) wei(HI)}
    J|ABI cum(b) {spe(A,1a) wei(IJ)}
    K|ABJ cum(b) {spe(A,1a) wei(JK)}
all {eff(H,I,J,K,1,B,c,-1)}
rec 500
des [1 1 1 1 1 1 1 1 1 -1]
sta log wei(GH) [0 0 -1000 0]
sta log wei(HI) [0 0 -1000 0]
sta log wei(IJ) [0 0 -1000 0]
sta log wei(JK) [0 0 -1000 0]
dat ex811.dat
```

The probit link is specified by the `cum(b)` statements after the transition probabilities. The effect of `A` is assumed to be time dependent, while the effect of B is time independent. The weight vectors, which because of the `log` statement after `sta` are in the probit scale, make the (2,1) transition for each time point equal to zero. Note that the probability that a z-value is smaller than -1000 equals zero.

A log-log model for the probability of not having event, which is equivalent to a complementary log-log model for the probability of having an event, can be obtained by replacing the `cum(b)` statements by `cum(d)`. In addition, the starting values weight vector in the log-log scale must be changed into `[0 0 1000 0]`. Note that exp(-exp(1000)) equals zero.

## 8.3 Time-varying covariates

One of the strong points of event history analysis is the possibility of using time-varying covariates. In $\ell_{EM}$, there are two methods for including time-varying covariates in log-rate or discrete-time logit models: episode splitting and expansion of the state space.

### 8.3.1 Episode splitting

Episode splitting involves creating episode records for which the covariates are constant (Blossfeld and Rohwer, 1995; Vermunt, 1996b, 1997). An episode record contains, besides the covariate values, the starting time and state and the end time and state of the episode. For each individual, we have as many episode records as the number of times that the time-varying covariates change their values plus one.

Suppose that we have a hazard model with two time-constant covariates `A` and `B` and a dichotomous time-varying covariate `C`. A $\ell_{EM}$ input file for this situation could be of the form

```
* example 8.12: time-varying covariate via episode splitting
man 3
dim 2 2 2
lab A B C
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {A,B,C,T}
rec 800
rt0
dat ex812.dat
```

The model specification is the same as in the case in which all covariates are time constant. What is different is the structure of the data. With `rt0` it is indicated that the records in data file contain a starting time and state. The first seven episode records in the data file could, for example, be

```
1 2 2  0 0 14 0  * episode 1 for case 1
1 2 1 14 0 20 0  * episode 2 for case 1
2 1 2  0 0  4 1  * episode 1 for case 2
1 1 1  0 0  9 0  * episode 1 for case 3
1 1 2  9 0 12 1  * episode 2 for case 3
2 2 1  0 0 10 1  * episode 1 for case 4
1 2 2  0 0 20 0  * episode 1 for case 5
etc.
```

The first two records belong to the first case, the third record to the second case, the fourth and fifth record to the third case, and the last two records to the fourth and fifth case, respectively. As can be seen, the first episode for case 1 ends at the 14th time unit. At that time unit the value of `C` changes from 2 to 1. Therefore, the second episode for case 1 starts at time unit 14 with a value of 1 for time-varying covariate `C`. Case 2 (episode record 3) does not have a change in `C`, while case 3 (records 4 and 5) experiences a change in `C` at time unit 9. The value of `C` do not change for cases 4 and 5.

### 8.3.2 Expansion of the state space

Another method for dealing with time-varying covariates is via expansion of the state space. Although this method is conceptually more complicated, it has some important advantages. First of all, it makes it unnecessary to perform episode splitting with some other computer program. In addition, it makes it possible to simultaneously model the covariate and the dependent process, including the use of latent variables for dealing with unobserved heterogeneity (Vermunt 1996b, 1997).

Suppose we have the same problem as above, that is, a model with two time-constant and one time-varying covariate. Assume again that the time-varying covariate can take on two values. We can specify the same model for a single nonrepeatable event in the following way

```
* example 8.13: time-varying covariate via expansion of the
*               state space
man 2
dim 2 2
lab A B
tim 5 [0,4,8,12,16,20]
ris 4 [0,1] [2,3]
haz {A,B,R,T}
rec 500
rt0
epi 2
dat ex813.dat
```

As can be seen, we have 4 states and two transition of interest. State 0 means that the event did not occur and that the time-varying covariate has value 1, state 1 that the event did occur and that the time-varying covariate has value 1, state 2 that the event did not occur and that the time-varying covariate has value 2, and state 3 that the event did occur and that the time-varying covariate has value 2. So, in fact, we cross the normal state space of two values with the possible values of the time-varying covariate.

The transitions `[0,1]` and `[2,3]` denote the occurrence of the event of interest for individuals with values 1 and 2 on the time-varying covariate, respectively. So, in fact, the risk variable has the same function as the time-varying covariate `C` in the previous example. Of course, transitions from state 0 to states 2 and from state 2 to state 0 are possible as well. These changes in the value of the time-varying covariate for individuals who did not experience the event are, however, not modelled and therefore not specified as risks.

With `rt0` and `epi 2` it is indicated that the records in the data file contain a starting time and state and, in addition, two episodes or spells. The records for the same five cases as in the previous examples are now

```
1 2  0 2 14 0 20 0  * case 1
2 1  0 2  4 3  4 3  * case 2
1 1  0 0  9 2 12 3  * case 3
2 2  0 0 10 1 10 1  * case 4
1 2  0 2 20 2 20 2  * case 5
etc.
```

The starting time is 0 for all cases, while the starting state is either 0 or 2. Cases 1 and 3 change their values on the time-varying covariate. Cases 2, 3, and 4 experience the event, while cases 1 and 5 are censored.

### 8.3.3 In log-linear path models

If a discrete-time model is specified as a log-linear path model, the use of a time-varying covariate involves using one additional variable for each time point. Assume that we have observations on the occurrence of a single nonrepeatable event at three time points. Let `A` and `B` be time-constant covariates, `D`, `E`, and `F` the value of a time-varying covariate at each of the three time-points, and `I`, `J`, and `K` the states occupied at the three time points. A discrete-time logit model for such situation could be of the form

```
* example 8.14: log-linear path model with time-varying
*               covariates
man 8
dim 2 2 2 2 2 2 2 2
```

```
lab A B D E F I J K
mod I|ABD  {AI,BI,DI}
    J|ABIE {AJ,BJ,EJ wei(IJ)}
    K|ABJF {AK,BK,FK wei(JK)}
rec 500
dat ex810.dat
sta wei(IJ) [1 1 0 1]
sta wei(JK) [1 1 0 1]
```

In this model, the time-constant covariates and the time-varying covariates have time-specific effect. The weight vectors are used to make the (2,1) transition impossible.

## 8.4 Latent variables

One of the strong points of the $\ell_{EM}$ program is that it makes it possible to simultaneously specify a (log-linear path) model for the covariates and an event history model for the dependent process under study. This makes it straightforward to include latent variables in event history models. Below, three possible application of the use of latent variables are presented: correcting for unobserved heterogeneity, correcting for measurement error in observed covariates, and correcting for measurement error in observed states.

### 8.4.1 Unobserved heterogeneity

Hazard models with a non-parametric characterization of the distribution of the unobserved heterogeneity component can be specified by including a latent covariate in the model. This approach to unobserved heterogeneity is called a non-parametric random-effects approach (Heckman and Singer, 1982, 1984; Mare, 1994; Guo and Rodriguez, 1994; Vermunt 1996b, 1997). An example of a model for a single nonrepeatable event with two observed and one unobserved covariate is

```
* example 8.15a: unobserved heterogeneity
man 2
lat 1
dim 2 2 2
lab X A B
mod X|AB {X}
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {A,B,X,T}
rec 500
dat ex815.dat
```

As can be seen, this model contains a latent variable `X` with two classes. The model for the covariates specifies that `X` is independent of the observed covariates `A` and `B`, which is the standard assumption in models with unobserved heterogeneity. In the hazard model, the latent variable `X` can be used in the same way as the observed covariates and the time variable. Here, we specified a simple proportional hazard model.

Interesting variants can be obtained by changing the specification of the covariate or hazard part of the model. For instance, replacing the `mod` statement by

```
mod X|AB {XA,XB}
```

yields a model in which the unobserved heterogeneity is related to the observed heterogeneity. In addition, we might include interactions between X and the other variables in the hazard model, for example, by a hazard model of the form

```
haz {XA,XB,XT}
```

Another possibility is the specification of mover-stayer structures (Farewell, 1982). This involves restricting the hazard rate for one of the latent classes to zero, which can be accomplished by

```
haz {A,B,T,wei(X)}
```

in combination with

```
sth wei(X) [1 0]
```

to specify the (starting) values of the weight vector which appears in the hazard model.

Of course, it is also possible to assume the latent variable to have more than 2 latent classes. This can be accomplished by changing the number of levels of the latent variable in dim. As was already explained in the context of latent trait models, one can also approximate continuous mixing distribution by fixing the form of the latent distribution and assuming the effect of the latent variable to be linear. A model with a normally distributed mixture variable is obtained by

```
* example 8.15b: unobserved heterogeneity with normally distributed
*                mixture variable
man 2
lat 1
dim 9 2 2
lab X A B
mod X|AB {wei(X)}
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {A,B,spe(X,1b),T}
rec 500
sta wei(X) nor(1,8)
dat ex815.dat
```

For the distribution of latent variable X with 9 classes, we specify a weight vector which has a normal distribution as "starting value". In addition, the effect of X on the hazard rate is made linear by predefined design type 1b.

Although in the above example we used only one latent variable, it is also possible to specify hazard models with more than one latent covariate. This can be useful in models for competing risks, repeatable events, or multiple-state processes. Note that in models with several latent variables, we also have to specify a model for the (nonparametric) joint distribution of the unobserved variables.

### 8.4.2 Measurement error in covariates

Another application of latent class models in the context of event history analysis is correcting for measurement error in observed covariates (Vermunt 1996b, 1997). Assume that A, B, and C are imperfect indicators for the latent variable X which we want to use as a covariate in a hazard model. Such a model can be specified as follows:

```
* example 8.16: measurement error in observed covariates
man 3
lat 1
dim 2 2 2 2
lab X A B C
mod X A|X B|X C|X
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {T,X}
rec 500
dat ex816.dat
```

As can be seen, the covariate model is just a latent class model. The latent covariate `X` is used in the hazard model, which in this case is a proportional hazard model.

### 8.4.3 Measurement error in observed states

Latent variables can also be used to correct for measurement error in the observed states if a discrete-time logit model is specified as a log-linear path model (Vermunt, 1996b, 1997). An example of discrete-time logit model for transitions between latent states is

```
* example 8.17: discrete-time logit model for latent states
man 7
lat 5
dim 2 2 2 2 2 2 2 2 2 2 2 2
lab V W X Y Z A B G H I J K
mod W|ABV {AVW}
    X|ABW {AWX}
    Y|ABX {AXY}
    Z|ABY {AYZ}
    G|V
    H|W eq1 G|V
    I|X eq1 G|V
    J|Y eq1 G|V
    K|Z eq1 G|V
all {spe(BVW,BWX,BXY,BYX,1a)}
rec 500
dat ex817.dat
```

In fact, this is a latent Markov model with covariates. As can be seen the measurement error in the observed states `G`, `H`, `I`, `J`, and `K` is assumed to be time homogeneous. As in the example on transitions between observed states, the parameterization of the effects may be adapted to agree with the standard discrete-time logit model.

## 8.5 Partially missing data

Not only the latent variables approach, but also the $\ell_{EM}$ tools for dealing with partially observed data can be used in the context of event history analysis.

### 8.5.1 Missing data in covariates

One type of partially missing data that can be dealt with are missing values on the covariates in a hazard model (Schluchter and Jackson, 1989; Baker, 1994; and Vermunt, 1996b, 1997).

Suppose that we have a hazard model with two covariates `A` and `B` and that for some individuals the value of `B` is missing. Such a problem can be handled as follows with $\ell_{EM}$:

```
* example 8.18: partially missing covariate
man 2
res 1
dim 2 2 2
lab S A B
sub AB A
mod AB S|A
tim 5 [0,4,8,12,16,20]
ris 2 [0,1]
haz {A,B,T}
rec 500
dat ex818.dat
```

The `res` and `sub` statements are used to specify that there are persons with missing information on `B`. The model statement `mod AB S|A` indicates that the missing data are assumed to be missing at random (MAR). The specification of the hazard model is exactly the same as when there is no missing data.

Of course, it not a problem to specify other types of models for the response mechanism. For instance,

```
mod AB S|B
```

would yield a NMAR (not missing at random) or nonignorable response mechanism. In addition, it is possible to use the response indicator as a covariate the hazard model to check whether the hazard rate differs for individuals for which `B` is missing.

### 8.5.2 Missing data in states

In event history analysis, we are often confronted with missing data on the dependent variable. Two common forms of missing data are right censoring and left censoring. Right censoring is easy to deal with as long as it can be assumed that the missing data are MAR. Left censoring causes no problems if the time of entry into the risk set is known. If we have other types of missing data or if we want to specify a model the response mechanism, it may be useful to use the tools for dealing with partially missing developed in the context of log-linear path models. These models for nonresponse can be used if a discrete-time logit model is specified as a log-linear path model (Baker, 1994; Vermunt, 1996b, 1997).

Suppose we have a discrete-time logit model for five point in time. For all individuals we have information on the state occupied at the first point in time, but that information may be missing for any of the other four time points. An example of a model for such a situation is

```
* example 8.19: discrete-time logit model with partially
*               observed states
man 7
res 4
dim 2 2 2 2 2 2 2 2 2 2
lab S T U V A B G H I J K
sub STUV STU STV ST
    SUV  SU  SV  S
    TUV  TU  TV  T
    UV   U   V   -
```

```
mod H|ABG {AGH}
    I|ABH {AHI}
    J|ABI {AIJ}
    K|ABJ {AJK}
    S|GH    {S}
    T|SHI   {ST}
    U|STIJ  {STU}
    V|STUJK {STUV}
all {spe(BGH,BHI,BIJ,BJK,1a),spe(GS,HT,IU,JV,1a),
     spe(HS,IT,JU,KV,1a)}
rec 500
dat ex819.dat
```

As can be seen, we specified a model with four response indicators. The subgroups are specified in such a way that `S`, `T`, `U`, and `V` indicate missingness of `H`, `I`, `J`, and `K`, respectively. In the model for the response mechanism, the probability of observing a person's state is assumed to depend on the previous and the current state, which implies a nonignorable nonresponse mechanism. In addition, these effects are assumed to be time homogeneous.

# Chapter 9

# Settings

Besides the commands for specifying the log-linear and hazard model of interest and the format of the data, $\ell_{EM}$ contains a large number of additional commands. The purpose of the most important ones are described below. Most of these commands have to be used after the specification of the model and the data format, and their mutual order is free.

## 9.1 Reading data, designs, and fixed-value parameters

In the examples presented in the previous chapters, we already demonstrated the usage of the commands, `dat`, `des`, and `sta`.

With `dat`, we specified the data or the file from which the data could be read. The data could be either in the form of a frequency table or individual records. The command `dat` is always required.

The command `des` was used to specify the designs 'announced' in the model specification. The designs must be defined in the order in which they appear in the model specifications. Per submodel, first the user-defined design are expected and then the designs associated with the restricted association models.

With `sta`, we specified 'starting values' for a weight vector defined via `wei()` and for probabilities which to be fixed to a particular value via `eq2`. Below, we discuss another possible reason for using `sta`.

## 9.2 Influencing the estimation process

### 9.2.1 Starting values: identification and local maxima

A well-known problem in the estimation of models with latent variables and other types of missing data is the occurrence of local maxima. Moreover, identification is not always assured. Therefore, it may be helpful to be able to manipulate with the starting values of the parameter estimates.

In the default setting, $\ell_{EM}$ will produce random starting values for the parameters in models with latent variables or missing data. The seed of the random number generator is initiated on the basis of the current time (seconds and hundreds of seconds). The random number generator yields starting values between -0.25 and 0.25 for the log-linear parameters. If there are no latent variables or missing data, the starting value for all (log-linear) parameters is zero.

Although for most users the default setting with respect to starting values will be sufficient, there are several commands to overrule this default setting. By putting `ran` after a submodel we get random starting values in situations in which we would not have them. In the same way, but now with `nra`, it is possible to suppress the random starting values in a particular submodel.

With the command `see <seed>`, it is possible to set the seed of the random generator to a specific value. Besides, it is possible to request more extreme starting values.

And finally, the command `sta` can be used to give starting values for specifics parameter of the log-linear, while `sth` serves the same purpose for parameters of the hazard model.

### 9.2.2 Newton-type algorithms

The main algorithm implemented in $\ell_{EM}$ is the EM algorithm, with IPF and uni-dimensional Newton in the M step of the algorithm. For most models, it is, however, possible to switch to a Newton-type algorithm after some iteration. This can be indicated by means of the command `new <iteration> <algorithm>`. The algorithms which are implemented are Newton-Raphson, BFGS, Levenberg-Maquardt, and Steepest-descent.

### 9.2.3 Convergence

The iterations are stop when a convergence criterium is reached or when a certain amount of iterations is performed. The criterium is the minimum increase in the log-likelihood function between subsequent iterations. Its default value is 0.000001. The criterium can be change with the command `cri <value criterium>`. The maximum number of iterations is set to 5000. This can be changed with `ite <maximum number of iterations>`.

### 9.2.4 Coding of parameters

The command `dum <reference categories>` can be used to change the coding scheme of the hierarchical log-linear parameters and the parameters obtained via the predefined design (`spe(..)`) type 1a. With `sca <scaling method>`, one can specify the scaling method for the parameters of log-multiplicative association models.

### 9.2.5 Others

Three additional commands for influencing the estimation process: `ste` (step size), `mit` (M iterations), `add` (add to cell frequencies), and `sim` (simulate data).

With `ste <decrease factor>`, one can change the step size used by the Newton-type algorithms, including the uni-dimensional Newton algorithm which is used in the M step of the EM algorithm. The default step size is 1.

The command `mit <number>` makes it possible to change the number of iterations in the M step of the EM algorithm. The default value is 1.

With `add <number>`, one can add a small number to each cell entry of the frequency table to be analyzed. This command can only be used in conjunction with data in the form of a frequency table. Moreover, the command must precede the specification of the frequency table.

An interesting option is the possibility to simulate data. It should be noted that this is only possible for the log-linear model and not for the hazard model. This is accomplished with the command `sim <number of cases> <file to write data>`. Of course, it is important to supply starting values for the parameters, since these will serve as the population values.

## 9.3 Output options

### 9.3.1 Suppress

The output file contains an echo of the input, statistics, frequencies, R-squared measures, log-linear parameters, hazard parameters, (conditional) probabilities, and latent class output. In some cases, we may want to suppress some of this information from the output file, for instance,

to save time or disk space. There are several commands, which do all start with the character 'n' of no, to suppress particular output sections. These no commands are: `nec` (echo of the input), `nfr` (observed and expected frequencies), `nze` (zero observed frequencies), `nR2` (R-squared measures), `nco` (conditional probabilities), `nla` (latent class output), `npa` (parameters estimates), `nse` (standard errors), and `nlo` (output for log-linear model).

### 9.3.2 Additional

Besides suppressing output, it is also possible to ask additional output. More precisely, one can request to write some information to a specified file. These write commands start with the character 'w'. We have the write commands `wda` (data), `wfr` (frequencies), `wfi` (fitted frequencies), `wma` (marginal table), `wco` (conditional probabilities), `wpo` (posterior probabilities), `wla` (latent classification probabilities), `wse` (standard errors, correlations, variances, and covariances), `wha` (hazard rates), `wsu` (survival probabilities), `wex` (exposure times), and `wfa` (failures).

# Chapter 10

# Content of the output file

During a run, the user is informed about the action that $\ell_{EM}$ is performing. This may be loading data, iteration number, computation of frequencies and statistics, standard errors, log-linear parameters, or latent class parameters. In addition, at each iteration the value of the log-likelihood function, the increase in the log-likelihood function, and the value of the likelihood-ratio chi-squared statistic are printed to the screen.

Depending on the type of application, the output file may consist of the following sections:

1. Input;
2. Statistics;
3. Frequencies;
4. Pseudo R-squared measures;
5. Log-linear parameters;
6. Hazard parameters;
7. (Conditional) probabilities;
8. Latent class output.

The next sections of this chapter describe these eight output parts.

## 10.1 Input

The first item in the output file is an echo of the input file. This item can be suppressed by means of the command `nec` (no echo).

## 10.2 Statistics

After the echo of the input file, the output file contains a set of items which are grouped under the name statistics. The first two items of the statistics part are:

- number of iterations;
- last increase in the log-likelihood.

The next part contains the statistics for the log-linear (path) model and the hazard model. Let $n_i$ be an observed cell count and $\hat{m}_i$ and estimated expected cell count. For a log-linear (path) model, the program reports

- Pearson chi-squared statistic,

$$X^2 \;=\; \sum_i \frac{(n_i - \hat{m}_i)^2}{\hat{m}_i}\,;$$

- likelihood-ratio chi-squared statistic,

$$L^2 \;=\; 2\,\sum_i n_i \log\left(\frac{n_i}{\hat{m}_i}\right)\,;$$

- Read-Cressie chi-squared statistic with $\lambda$=2/3,

$$RC^2 \;=\; \frac{2}{\lambda\,(\lambda+1)}\,\sum_i n_i \left[\left(\frac{n_i}{\hat{m}_i}\right)^{\lambda} - 1\right]\,;$$

- index of dissimilarity,

$$D \;=\; \sum_i \mathrm{abs}\,(n_i - \hat{m}_i)\,/\,(2\,N)\,;$$

- number of degrees of freedom,

$$df \;=\; \text{number of observed frequencies} - \text{number of (log-linear) parameters}\,;$$

- log-likelihood function,

$$\log \mathcal{L}_\ell \;=\; \sum_i n_i \log \hat{\pi}_i\,;$$

- number of (log-linear) parameters;
- $BIC$ based on the $L^2$ statistic,

$$BIC_1 \;=\; L^2 - df\;\log N\,;$$

- $AIC$ based on the $L^2$ statistic,

$$AIC_1 \;=\; L^2 - df\;2\,.$$

For a hazard model, the following output is reported:

- Pearson chi-squared statistic;
- likelihood-ratio chi-squared statistic;
- number of degrees of freedom;
- log-likelihood function;
- number of hazard parameters.

And finally, for the log-linear and the hazard model together, the program reports

- total number of parameters ($npar$);
- log-likelihood function ($\log \mathcal{L}$);

- number of cases ($N$);
- $BIC$ based on the log-likelihood function,

$$BIC_2 \quad = \quad -2 \ \log \mathcal{L} + npar \ \log N \, ;$$

- $AIC$ based on the log-likelihood function,

$$AIC_2 \quad = \quad -2 \ \log \mathcal{L} + npar \ 2 \, .$$

For the more information about the use and interpretation of above statistics, see handbooks on log-linear modeling, such as Agresti (1990) and Hagenaars (1990).

Besides the above-mentioned statistics, one obtains some information on the identification of the parameters, that is,

- eigenvalues of the information matrix, number of boundary or non-identified parameters, and number fitted zeros.
  - If all parameters are identified and none of the parameters is on or too near to the boundary, all eigenvalues of the information matrix should be larger than zero. In other words, zero or negative eigenvalues indicate that some parameters are either not identified or too close to the boundary to check their identification and calculate their standard errors. If a parameter is on or close to the boundary, one or more probabilities will be (near to) zero.
  - Non-identified parameters may occur in models with latent variables. They may also occur if one does not impose the required identifying restrictions on the (log-linear) parameters, that is, if the model contains redundant parameters. A third possible reason is that parameters may be inestimable as a result of observed zero cell frequencies (Clogg and Eliason, 1987).
  - The program prints a warning if some parameters are non-identifiable or near to the boundary. The program also reports the number of fitted zeros in the observed frequency table. Generally, we can correct the number of degrees of freedom by subtracting the number of fitted zeros and adding the number of non-identifiable parameters to the reported number of degrees of freedom.
  - The output file will not contain the information on identification if the model contains association models (`ass(..)`), log-multiplicative scaling factors (grouping type `b`), linear restrictions on frequencies (`lin(..)`), or constraints on the (conditional) probabilities (`eq2`, `or1`, or `or2`).
  - The computation of the information matrix and its eigenvalues can be suppressed with the command `nse` (no standard errors).

## 10.3 Frequencies

For every (nonresponse) subgroup, the program will give the following information:

- observed frequencies;
- estimated expected frequencies;
- standardized residuals,

$$r_i \quad = \quad \frac{n_i - \hat{m}_i}{\sqrt{\hat{m}_i}} \, ;$$

- Pearson chi-squared statistic;

- likelihood-ratio chi-squared statistic.

This output part, which can be quite large for huge tables, can be suppressed with the command `nfr`. One can also suppress only information for the observed zero entries by means of `nze`. If the observed frequency table has more than 1000 cells, the program will report only the non-zero observed cells.

## 10.4 Pseudo R-squared measures

For each of the conditional probabilities of a log-linear path model, the output file contains the value of five different (pseudo) R-squared measures for nominal dependent variables. These R-squared measures are based on the well-known definition of explained variance, that is,

$$R^2 = \frac{S_y^2 - S_e^2}{S_y^2}\,,$$

where $S_y^2$ and $S_e^2$ denote the total and error variance, respectively. The problem is, however, that for nominal variables there is not a single generally accepted definition of variance. The R-squared values reported in the output file are based of five different variance measures for nominal variables, namely, entropy, qualitative variance or concentration, proportion of prediction errors, log-likelihood function, and likelihood function. For references, see Magidson (1981), Gilula and Haberman (1994, 1995), Maddala (1983), and Aldrich and Nelson (1984).

Let $i$ denote a value of the dependent variable in the submodel concerned and $k$ a value of the joint independent variable. Moreover, let $\hat{\pi}_{i|k}$ denote an estimated expected conditional probability, $n_{ik}$ an observed frequency, and $n_{i+}$ a marginal frequency obtained by collapsing over the index $k$. In addition, let $p_{i+}$ $(= n_{i+}/N)$ denote the observed marginal probability of having value $i$ on the dependent variable and $p_{+k}$ the probability of having value $k$ on the joint independent variable. It should be noted that in models with latent variables or missing data, $n_{ik}$, $n_{i+}$, $p_{i+}$, $p_{+k}$ may be estimated quantities rather than observed ones.

Using entropy as variance measure, the total and error variances equal

$$\begin{aligned} S_y^2(1) &= -\sum_i p_{i+} \log p_{i+} \\ S_e^2(1) &= -\sum_k \left[\sum_i \hat{\pi}_{i|k} \log \hat{\pi}_{i|k}\right] p_{+k}\,. \end{aligned}$$

Concentration or qualitative variance is defined as:

$$\begin{aligned} S_y^2(2) &= \left(1 - \sum_i (p_{i+})^2\right)/2 \\ S_e^2(2) &= \sum_k \left[\left(1 - \sum_i (\hat{\pi}_{i|k})^2\right)/2\right] p_{+k}\,. \end{aligned}$$

The third measure uses the minimum number of classification errors, which yields

$$\begin{aligned} S_y^2(3) &= 1 - \max(p_{i+}) \\ S_e^2(3) &= \sum_k \left[1 - \max(\hat{\pi}_{i|k})\right] p_{+k}\,. \end{aligned}$$

The fourth R-squared measure uses $-2/N$ times the log-likelihood function as variance measure, that is,

$$\begin{aligned} S_y^2(4) &= -2/N \sum_k n_{i+} \log p_{i+} \\ S_e^2(4) &= -2/N \sum_k \sum_i n_{ik} \log \hat{\pi}_{i|k}\,. \end{aligned}$$

Aldrich and Nelson (1984) proposed a pseudo R-squared measure based on the log-likelihood function which is defined by $\Big(S_y^2(4) - S_e^2(4)\Big) \,/\, \Big(1 + S_y^2(4) - S_e^2(4)\Big)$ rather than by the standard R-squared formula.

The fifth R-squared measure is based on the likelihood function raised to the power $-2/N$. In a linear regression model estimated via maximum likelihood assuming a normally distributed error term, such a definition of $S_y^2$ and $S_e^2$ yields the standard R-squared measure. For (product-)multinomial sampling, we get

$$\begin{aligned} S_y^2(5) &= \prod_i (p_{i+})^{-2\, n_{i+}/N} \\ S_e^2(5) &= \prod_k \prod_i (\hat{\pi}_{i|k})^{-2\, n_{ik}/N} \,. \end{aligned}$$

This variance measure has, however, one drawback: it cannot be smaller than one and, as a result, the $R^2$ value cannot become zero. This can be corrected by replacing the standard R-squared formula by $\Big(S_y^2(5) - S_e^2(5)\Big) \,/\, \Big(S_y^2(5) - 1\Big)$.

De computation of these five R-squared measures and their two variants can be suppressed by means of the command `nR2`.

## 10.5 Log-linear parameters

Under the heading log-linear parameters, the $\ell_{EM}$ output file reports the parameters estimates of the various submodels, that is, the parameters of the models which are specified for the different (conditional) probabilities. The computation of these parameters can be suppressed with the command `npa`.

The parameters may be log-linear parameters, parameters of one of the various types of regression models, threshold parameters of cumulative models for ordinal dependent variables, parameters of log-linear and log-multiplicative association models, or parameters of models with log-multiplicative scaling factors. For parameters of log-linear and logit models, also the exponent of the parameter concerned is presented.

In most cases, the program will also report the estimated standard errors of the parameters. For some types of models, however, the computation of standard errors is not implemented. More precisely, the output file will not contain standard errors if the specified model contains association models specified with `ass(..)`, log-multiplicative scaling factors, linear restriction on cell frequencies, or constraints on the conditional probabilities specified with `eq2`, `or1`, or `or2`. In fact, standard errors are given in all cases in which the information matrix is computed to check the identifiability of the model parameters. For parameters which are non-identified or near to or on the boundary, the program gives no standard errors. The computation of standard errors, which can be time consuming, can be suppressed with the command `nse`.

In situations in which the program reports standard errors, one also obtains the z-value for non-redundant parameters and the value of the Wald chi-squared statistic for sets of parameters.

For the hierarchical log-linear effects and for some of the predefined designs, the user can determine the coding scheme to be used to identify the parameters. The default coding scheme is effect coding. This can be changed with the command `dum`.

The hierarchical log-linear parameters are computed from the cumulated multipliers for each marginal cell which has to be reproduced according to the specified model, that is, for each of the minimal sufficient statistics (Vermunt, 1996b, 1997). With these multipliers it is quite easy to get correct parameter estimates, even in models with structural zeros, fixed effects, or user-defined effects. However, if the minimal sufficient statistics contain zero entries, computation of the hierarchical log-linear effects is no longer straightforward. The solution that is chosen here is to skip the margins containing zeros when calculating a particular parameters. This may lead

to parameter estimates which are no longer consistent with the requested coding scheme. Thus, one must be cautious with the interpretation of the reported effects when fitted zero margins occur. The program gives a warning if there are zeros in the fitted margins.

## 10.6 Hazard parameters

The output section on hazard parameters gives the parameters of the hazard model and their asymptotic standard errors. The parameters may be hierarchical log-linear parameters, parameters from predefined designs, parameters from user-defined designs, and parameters from association models.

It should be noted that a very specific type of coding scheme is used for the risk variable. In log-rate models with more than one risk, the time and covariate effects are parameterized as effects on the different types of events rather than as effects on the overall hazard rate and effects on the deviation from the overall hazard rate. This is simply accomplished by setting the effects on the overall hazard rate equal to zero. In discrete-time logit models, within each origin state, the no-event 'transition' is used as reference category (see Vermunt, 1996b, 1997). This means that the parameters can be interpreted as effects on the odds of having an event of the type concerned rather than not having an event.

## 10.7 (Conditional) probabilities

Besides the parameters of the various submodels, the program reports the estimated (conditional) probabilities according to the specified submodels. In situations in which the program computes standard errors, the program will also report the standard error of each of the conditional probabilities. These standard errors are computed from the variance-covariance matrix of the log-linear parameters using the delta method. The appearance of this section in the output file can be suppressed with the command `nco`.

## 10.8 Latent class output

When the log-linear path model contains latent variables, the program reports the classical latent class output. Assume that we have a log-linear path model with two latent variables with indices $x$ and $y$ and four observed variables with indices $a$, $b$, $c$, and $d$. The latent class output consists of the following three items:

- latent and conditional probabilities: $\hat{\pi}_{xy}$, $\hat{\pi}_{a|xy}$, $\hat{\pi}_{b|xy}$, $\hat{\pi}_{c|xy}$, and $\hat{\pi}_{d|xy}$;
- estimated expected proportion of classification errors if we use modal assignment,

$$E \;=\; \sum_{abcd} \left[1 - \max\left(\hat{\pi}_{xy|abcd}\right)\right] \hat{\pi}_{abcd}\,;$$

- reduction in the proportion of classification errors,

$$lambda \;=\; \frac{[1 - \max\left(\hat{\pi}_{xy}\right)] - E}{1 - \max(\hat{\pi}_{xy})}.$$

It should be noted that the latent and conditional response probabilities which are reported in this part of the output file need not to be the probabilities which appear in the specification of the log-linear path model. This has to be taken into account when interpreting the latent class output. The latent class output can be suppressed by means of the command `nla`.

# Chapter 11

# Complete command syntax

This chapter describes the complete command syntax of the $\ell_{EM}$ program. Examples of the use of this syntax can be found in the first chapters of this manual.

The $\ell_{EM}$ syntax consists of commands which have to be typed in lower case and of which only the first three characters are significant. It is recommended to use upper-case labels for the variables to prevent confusion with the $\ell_{EM}$ commands. The input file is read in free format, with spaces or commas as separation characters. Comments can be put in the input file using asterisks. When a '`*`' is encountered, the rest of the line is considered to be comment, and therefore skipped.

An input file may consists of four parts, that is, the specification of

1. log-linear (path) model,
2. event history model,
3. data format,
4. settings.

If present, these four parts must appear in this order in the input file. The next four sections of this chapter describe these four sets of commands. The last section of this chapter deals with the various types of parameterizations that can be used for restricting cell frequencies, probabilities, and hazard rates.

## 11.1 Log-linear (path) model

The first part of the input file will generally consist of the specification of a log-linear (path) model. In this part, we have to specify which types of variables we are using and which type of model we want for these variables. The commands described in this section are `res`, `lat`, `man`, `con`, `dim`, `lab`, `sub`, `mod`, and `all`. These command have to be used in this order, except of the first four commands – `res`, `lat`, `man`, and `con` – which order may be interchanged.

- `res` <number of response indicators>
  - This command specifies the number of response indicators. It must be used if one wants to use partially observed data in the analysis.
  - *Default*: 0.
- `lat` <number of latent variables>
  - This command specifies the number of latent variables which are used in the model.
  - *Default*: 0.

- `man` <number of manifest variables>
  - This command specifies the number of manifest variables which are used in the model.
  - *Default*: 0.

- `con` <number of continuous exogenous variables>
  - This command specifies the number of continuous exogenous variables which are used in the model.
  - *Default*: 0.

- `dim` <list of number of categories of the variables>
  - This command specifies the number of categories of the response indicators, latent variables, and manifest variables. The command `dim` (dimensions) is required if `res`+`lat`+`man` is larger than 0.
  - One must first specify the number of categories of the response indicators, then of the latent variables, and then of the manifest variables.
  - Note that the index of the last variable changes fastest when reading the observed frequencies, the user-defined designs, and the starting values.

- `lab` <variable labels>
  - The command `lab` (labels) makes it possible to specify variable labels, first for the response indicators, then for the latent variables, then for the manifest variables, and the last one is for the continuous covariates. The set of continuous variables has only one label.
  - The variable labels have a maximum length of 3 characters. If only labels of 1 character are used, as in the examples in the first chapters of this manual, variables need not to be separated in the model specification. If at least one label is longer than one character, the variables have to be separated by a point, '.'.
  - It is recommended to use upper case characters in the labels to prevent confusion with the $\ell_{EM}$ commands.
  - *Default*: for the response indicators, `K`, `L`, `M`, etc.; for the latent variables, `X`, `Y`, and `Z`; for the manifest variables, `A`, `B`, `C`, etc.; and for the continuous variables, `x`.

- `sub` <list of subgroups>
  - If one has indicated that there is at least one response indicator, one may define the subgroups for which the same set of variables are observed. The number of subgroups is equal to the product of the number of categories of the response indicators.
  - For every subgroup, the manifest variables whose scores are not missing have to be specified. It should be noted that the order in which the subgroups are specified determines the meaning of the response indicators. This feature has to be used if one wants to specify a model for the response mechanism.
  - If all manifest variables are missing in a particular subgroup, the subgroup concerned has to be denoted by a '-'.
  - *Default*: the different subgroups are identified on the basis of the missing data patterns which are found in the data. Note that if there are response indicators and if the command `sub` is skipped, the data must be in the form of (individual) records.

- `mod` <probability 1> <submodel 1> `ran/nra`
  <probability 2> <submodel 2> `ran/nra`
  <probability 3> <submodel 3> `ran/nra`
  etc.

  - This command makes it possible to specify a log-linear (path) model for the latent variables, manifest variables, response indicators, and continuous exogenous variables declared earlier. The specification of a log-linear path model consists of two parts: The first part

is the probability structure for the joint distribution of all variables. The second part consists of the submodels for the various (conditional) probabilities.
- The probabilities are in the form `B|CD`, where `B` is the dependent variable and `C` and `D` serve as independent variables. Thus, dependent and independent variables are separated by a '`|`'. There is only one rule with respect to the specification of the probability structure: A variable may be used only once as dependent. Variables which are used as independent but not a dependent are treated as exogenous. For the variables which are not used at all, the program sets up an additional probability in which they are assumed to be independent of the other variables. If no probability is specified, it is assumed that the submodel concerns the full table, that is, the table containing all variables.
- The submodels for the various probabilities may be of many different forms. In fact, there are eight basic types of (sub)models:

1. log-linear model: `{`<parameters>`}`,
2. unrestricted probability: <probability>,
3. logit model: <probability> `{`<parameters>`}`,
4. cumulative model: <probability> `cum(`<type>`) {`<parameters>`}`,
5. equal submodels: <probability> `eq1` <probability>,
6. restricted probabilities: <probability> `eq2`,
7. ordinal probabilities: <probability> `or1` or `or2`,
8. correspondence analysis: `cor(..)`.

The different types of restrictions are described in more detail in section 11.5. As can be seen, the specification of a submodel is optional (type 2). If no submodel is specified, a saturated model is assumed for the (conditional) probability concerned.
- The optional commands `ran` and `nra` after the submodel make it possible to change the default setting with respect to the random starting values for parameters of the submodel concerned: `ran` to request random starting values and `nra` to suppress the random starting values. The default setting is that random starting values are used if there are latent variables or partially missing data.
- *Default*: a saturated model for the full table.

- `all {`<log-linear effects>`} ran/nra`
  - The optional command `all` makes it possible to impose restrictions on the parameters across submodels. The program will find out to which probability a parameter specified with `all` belongs. Between the parentheses, one may use the commands `cov(..)`, `fac(..)`, `spe(..)`, and `ass(..)`. Again, with `ran` and `nra` one can change the default setting of random starting values.

## 11.2 Event history model

After the specification of a log-linear (path) model, one may specify a hazard model in the form of a log-rate or a discrete-time logit model. The log-rate model is also known under the name piecewise exponential survival model. The hazard model may be a model for a single event, a competing-risk model, a model for repeatable events, or a multiple-state model. In the regression model for the hazard rates, we can use the latent, manifest, and continuous variables declared in the log-linear part of input as explanatory variables. In addition, time-varying covariates can be used.

The event history part of the $\ell_{EM}$ syntax consists of the commands `dis`, `tim`, `ris`, `mul`, `haz`, `zer`, and `exp`. The command `dis` can be used to indicate that the model is a discrete-time

logit model, `tim` to specify the time intervals, `ris` and `mul` to specify the types of events, `haz` to specify the regression model, `zer` to set the time to zero after each event, and `exp` to specify the exposure time in the time interval in which an event or censoring occurs.

- `dis`
  -By starting the specification of the hazard model with the command `dis`, we indicate that we want a discrete-time logit model.
  - *Default*: log-rate model.

- `tim` <number of time intervals> `[`<begin and end points>`]`
  - This option makes it possible to specify the time categories for the log-rate or discrete-time logit model. We can specify the number of time intervals and the begin and end points of these time intervals. Within these intervals we assume constant hazard rates or transition probabilities. Obviously, the specified end point of one time interval is the begin point of the next time interval. The end points are included in the interval, while the begin points are excluded.
  - If the option to specify the begin and end points of the time categories is used, one must specify one point more than the number of time intervals. The last specified point serves as end point for the last time interval.
  - *Default*: one time interval (=exponential survival) with begin point 0 and end point the largest observed survival time. Even if the number of time categories is specified, the second part remains optional. The default time points are: 0, 1, 2, ..., to the number of time intervals.

- `ris` <number of states> `[`<origin state>,<destination state>`]` `[.,.]` etc.
  - This option specifies the number of states and the types of risks, transitions, or events to be analyzed. The states are numbered from 0 to the number of states minus 1. The transitions to be analyzed are specified between brackets `[.,.]`
  - Time-varying covariates can be included in the model by defining them as different (sub)states. So, the number of states will generally be equal to the product of the number of categories of the time-varying covariates and the states which define the events of interest.
  - In discrete-time logit models, it is obligatory to include the self-transitions in the list of possible event.
  - *Default*: In log-rate models, 2 states and one transition, namely, `[0,1]`. In discrete-time logit models, 2 states and two transitions, namely, `[0,0]` and `[0,1]`.

- `mul` <number of types of events>
  - Rather than with `ris`, the types of events can also be specified via the command `mul`. This command makes it possible to specify multivariate event history models in a compact way. We no longer need to specify the number of states and the possible transitions among these states. We just indicate that there is a particular number of types of events.
  - When using `mul`, also the data must be defined in another way than when using `ris` (see section on data format).
  - *Default*: equal to default for command `ris`.

- `haz` <table> `{`<hazard model>`}` `ran/nra`
  - This command specifies the regression model for the hazard rates or transition probabilities. The form of the model is the same as for standard log-linear models. Between the braces, one may uses hierarchical log-linear effects, user-defined designs, predefined designs, and association models.
  - The label `T` is used for the time variable. If there is more than one possible risk, transition, or event, the label `R` is used for this risk variable.
  - It is optional to specify the table for which a hazard model is specified. This may be

useful if one wants to compare chi-squared statistics across models with different sets of covariates. If the table is not specified the table will only contain the variables which are actually used in the hazard model.

- `zer`
  - This command sets the time to zero after every event. This makes it possible to use the duration since the last event, or waiting time, as the time variable without the necessity to indicate this in the data file.
  - *Default*: the time continues after an event (process time).

- `exp` <number between 0 and 1>
  - This command makes it possible to specify the (mean) exposure time in the time unit in which censoring or an event occurs.
  - *Default*: in log-rate models, it is assumed that events an censorings occurs in the middle of the time unit, which is equivalent to an exposure time of 0.5. In discrete-time models, events and censorings are assumed to occur at the end of the time unit.

## 11.3 Data format

The $\ell_{EM}$ program accepts data in two types of formats: frequency tables and individuals records. Individual records can be used is all situations. The use of data in the form of a frequency table is not always allowed. More precisely, the data has to be in the form of individual records

1. if there are continuous covariates (`con`>0),
2. if there are missing data (`res`>0) and the subgroups are not specified with `sub`,
3. if the begin and end points for the time intervals are specified,
4. if the risks are specified with `ris` or `mul`.

But, if frequency table data is allowed, it is the default setting, which can be changed with the command `rec` (see below).

When reading the data in the form of a frequency table, it is assumed that the index of the last variable appearing in `dim` and `lab` changes first. With missing data, the data for the different subgroups are read in the order in which they are specified. It the data for a hazard model are in the form of a frequency table, the index of the time variable changes before the other variables. First, the table with number of events is read for all subgroups and then the table with number of censored observations.

Below, the commands for using data in the form of individual records are described. The command `rec` must be the first command in the data format section of the input file. The order of other optional commands is free. The commands `rco`, `ski`, and `mis` are relevant for both log-linear (path) models and hazard models, while `rt0` and `epi` are only relevant for hazard models.

- `rec` <number of records>
  - This command has two functions. It indicates that the data are in the form of individual records and specifies the number of records in the data file.

- `rco`
  - This command indicates that the records contain a count or frequency.
  - *Default*: no count.

- `ski [`<list columns to be skipped>`]`
  - This command makes it possible to skip some of the columns of the data file. This means that the data file may contain more variables than are actually used.
  - *Default*: no columns are skipped.

- `mis` <missing-value code>
  - This command makes it possible to specify a missing-data code for the manifest variables. This missing-data code is used to determine to which nonresponse subgroup a particular record belongs.
  - *Default*: 0.

- `rt0`
  - This command indicates that the data file contains information on the time and state on entry into the risk set ($t_0$ and $s_0$). This option makes it possible use left censored cases and to perform episode splitting.
  - *Default*: $t_0 = 0$ and $s_0 = 0$.

- `epi` <number of episodes>.
  - This command makes it possible to indicate the number of episodes that must be read for each record.
  - *Default*: If the events are specified with `mul`, number of types of events. Otherwise, 1.

Each record must consist of the following components (in this order):

1. if `man`>0: <values of the manifest variables>;
2. if `con`>0: <values of the continuous variables>;
3. if `rco`: <count>;
4. if `rto`: $< t_0, s_0 >$;
5. for $i = 1$ to the number of episodes: $< t_i, s_i >$.

Here, $t_i$ and $s_i$ denote the time and the state at the end of the $i$th episode.

In multivariate hazard models specified with `mul`, however, steps 4 and 5 are slightly different. Note that in that case, the number of episodes equals the number of types of events. For each type of event, we have $< t_{0i}, t_{1i}, d_i >$, where $t_{0i}$ is the starting time, $t_{1i}$ the survival time or total exposure time to event $i$, and $d_i$ the number of times that event type $i$ occurred. Specification of a starting time is only required if this is indicated with `rt0`. It should be noted that $d_i$ may serve as a standard censoring indicator for event type $i$ (a 0 means that the event did not occur while a 1 means that the event occurred), but may also be used to indicate the number times that event type $i$ occurred in the observation period. The latter use of $d_i$ makes it possible to specify (multivariate) Poisson regression models.

The episodes for each record are read until the state does not change between two episodes (censored observation) or until the specified number of episodes is reached. The program always starts on a new line when reading the information on a new record. In other words, it skips the remaining items which are on the last line of a particular record. This means that is not a problem if a record contains more episodes than are actually used, as long as they are on the same line.

## 11.4 Settings

This section presents the $\ell_{EM}$ commands which can be used to change all types of setting. The order of these commands is free and they are all optional, except of the command `dat` which is used to define the data. The command `des` is required if user-defined designs are declared in the model specification. Exceptions with respect to the free order of the commands are `add`, `sim`, `wla`, and `wpo`.

The commands to change the settings are specified in two parts. The first part consists of the input and estimation settings. The second part deals with the output settings.

### 11.4.1 Input and estimation settings

This subsection presents the commands to change input and estimation settings.

- `add` <number>
  - This option makes it possible to add a constant to each cell entry of the observed frequency table. This can be useful if fitted zeros occur as a result of zeros in the sufficient statistics. Note that a small number like 0.001 already solves this problem without disturbing the sample size too much. Agresti (1990) showed, however, that one must be cautious with the use this option.
  - This option can only be used in combination with table format data. In addition, the command `add` must be used before the command `dat`.
  - *Default*: 0.

- `cri` <minimum increase log-likelihood>
  - To change the stop criterion, the minimum increase in the log-likelihood.
  - *Default*: 0.000001.

- `dat` <file name> or `[`<data>`]`
  - This command must be used to specify the name of the file containing the data or to specify the data between square brackets.
  - If the data are in the form of a frequency table, the index of the last variable must change first. The nonresponse subgroups are read in the order in which they are specified.

- `des` <file name> or `[`<design>`]`
  - To specify the design matrix for user-defined designs, restricted grouping variables in user-defined and predefined designs, restricted association models, and equality and fixed-value restrictions on probabilities. One may specify the designs between square brackets or in a separate file.
  - Of course, it is important to specify the designs for the various effects in the correct order. The program first expects the designs for the various submodels of the log-linear path model and then for the hazard model. Within a submodel, first the designs for `cov(..)`, `fac(..)`, and `spe(..)`, then the ones for `ass(..)`, and then the design for `eq2` must be given.

- `dum` <list of reference categories>
  - The command `dum` makes it possible to change the default effect coding scheme for hierarchical log-linear parameters and log-linear parameters specified with `spe(..)` type `1a` into dummy coding.
  - For each variable, one has to specify the reference category. This has to be a value between 1 and the number of categories of the variable concerned. In that case, one obtains dummy-coded parameter estimates.
  - By indicating that the reference category is equal to -1, the coding scheme for the variable

concerned remains effect coding. This makes it possible to combine different types of coding.
- Finally, by using a reference category of 0, the parameters of the variable concerned will be identified by omitting the lower-order effects of the other variables.
- *Default*: effect coding.

- `ite` <maximum number of iterations>
  - To change the maximum number of iterations.
  - *Default*: 5000.

- `mit` <number M iterations>
  - To change the number of M iterations within one EM cycle.
  - *Default*: 1.

- `new` <iteration> <type>
  - This command makes it possible to switch from the standard algorithms (EM, IPF, and uni-dimensional Newton) to a Newton-type algorithm after a specified number of iteration.
  - Besides the iteration at which to switch, the type of Newton algorithm has to be specified with a number, that is, 1 = Newton-Raphson, 2 = Steepest-descent, 3 = BFGS, 4 = Newton-Raphson combined with EM, 5 = Levenberg-Maquardt, and 6 = Levenberg-Maquardt combined with EM.
  - *Default*: EM algorithm.

- `sca` <method>
  - This option makes it possible to change the scaling method for the row and column scores in log-multiplicative association models (RC and RC(M) models). The method must be indicated with a number: 1 = unweighted (sum of scores equal 0 and sum of squared scores equal to 1), 2 = uniform weights, 3 = marginal weights, 4 = first phi-parameter fixed to 1 and sum of scores equal to 0, 5 = no rescaling, 6 = fixed scores for first and last level.
  - *Default*: 1 = unweighted.

- `see` <number>
  - This option makes it possible to supply a seed for the pseudo random generator to get a particular series of random starting values.
  - *Default*: seed is based on the clock of the computer.

- `sim` <number of cases> <file name>
  - This option makes it possible to simulate data according to a specified log-linear path model.
  - Starting values have to be supplied for all model parameters. These starting values will serve as population values.

- `sta` <effect> or <probability> <file name> or `[`<starting values>`]`
  - With `sta` it is possible to supply starting values for the effects and probabilities included in the model. The starting values may be in a file or may be specified between square brackets, or may be in agreement with a normal distribution.
  - Examples of the specification of <effect> are `AB` (hierarchical log-linear), `fac(AB)` (user-defined design), and `spe(AB,1a)` (predefined design). An example of a <probability> is `B|A`.
  - The starting values are either probabilities or multiplicative parameters. By putting the statement `log` between `sta` and <effect>, one can supply starting values for the log-linear parameters rather than for the multiplicative parameters.
  - Rather than specifying the starting values, it is also possible to generate starting values

according to a normal distribution. This is specified be with `nor(<method>,<range>)` after <effect>. The parameter `<method>` indicates the method that must be used to derive the discretized normal probability distribution. The two methods are: 1 = rescaled density function and 2 = piece of the cumulative distribution function. The parameter `<range>` indicates the range of the normal distribution that has to be used. For example, a range 8 means that z values from -4 to 4 should be used.
- Starting values for the hazard model are specified in the same way but using the command `sth` rather than `sta`.

- `ste` <decrease factor>
  - To decrease the step size of the uni-dimensional and the other types Newton algorithms by some factor. For example, a decrease factor of 2 will make the step size 2 times smaller. The use of a smaller step size may be necessary when an algorithm fails to converge. Larger step sizes can sometimes lead to faster convergence.
  - *Default*: 1.

### 11.4.2 Output settings

This subsection describes the commands which can be used to suppress parts of the standard output and to request additional output. Note than the commands for suppressing output start with a '`n`' from no. The commands for writing additional output to a specified file start with a '`w`' from write.

- `nco`
  - Suppress printing of the (conditional) probabilities to the output file.

- `nec`
  - Suppress writing of an echo of the input to the output file.

- `nfr`
  - Suppress writing of the observed and estimated expected frequencies to the output file.

- `nla`
  - Suppress writing of the latent class output to the output file.

- `nlo`
  - Suppress writing of the parameters of the log-linear model, the observed and estimated expected frequencies, the R-squared measures, the conditional probabilities, and the latent class parameters to the output file.

- `npa`
  - Suppress writing of the log-linear and hazard parameters to the output file.

- `nR2`
  - Suppress writing of the R-squared measures to the output file.

- `nse`
  - Suppress writing of the standard errors and the information on identification (eigenvalues and fitted zeros) to the output file.

- `nze`
  - Suppress writing of the observed and estimated expected frequencies for the zero observed cell entries to the output file.

- `wco` <file name>
  - Write the estimated conditional probabilities to a file.

- `wda` <file name>
  - Write the observed frequency table to file in record format with a count.

- `wex` <file name>
  - Write the (estimated) observed exposure times to a file.

- `wfa` <file name>
  - Write the (estimated) observed failures to a file.

- `wfi` <file name>
  - Write the estimated expected, or fitted, frequencies for the complete table to a file.

- `wfr` <file name>
  - Write the (estimated) observed frequencies for the various submodels to a file.

- `wha` <file name>
  - Write the estimated expected hazard rates to a file.

- `wma` <margin> <file name>
  - Write a specific margin of the estimated expected frequencies to a file.

- `wla` <file name>
  - Write the latent classification probabilities, the modal class, the classification error for the modal allocation, a random assignment, and the classification error for the random assignment to a file. Depending on the data format, this is done for each observed cell entry or each record.

- `wpo` <file name>
  - Write the posterior probabilities to a file. This is done for every observed cell entry or every record, depending on the data format.

- `wsu` <file name>
  - Write the estimated expected survival probabilities for each origin state to a file.

## 11.5 Types of restrictions or parameterizations

This section describes the types of restrictions or parameterization that can be used for restricting cell frequencies, (conditional) probabilities, and hazard rates.

### 11.5.1 Hierarchical log-linear effects

The simplest type of restrictions are restrictions in the form of hierarchical log-linear effects. Hierarchical log-linear parameters must specified between the parentheses of the (sub)model concerned, indicating the margins which have to reproduced according to the specified model. Hierarchical log-linear effects, which may be used in log-linear models, logit models, and hazard models, are fitted with the Iterative Proportional Fitting Algorithm (IPF) or one of the multidimensional Newton methods. The first chapters of this manual give many examples on the use or hierarchical log-linear effects.

### 11.5.2 User-defined designs

User-defined designs can be specified for the effects in log-linear models, logit models, regression models with cumulative link functions, and hazard models. There are two commands for specifying user-defined design `cov(..)` and `fac(..)`, which have to be used between the parentheses for the (sub)model concerned. Covariates (`cov(..)`) make it possible to specify standard interval level designs, while factors (`fac(..)`) can be used to specify dummy coded nominal design in a very compact way. The parameters of user-defined designs are fitted by means of the uni-dimensional Newton algorithm (Goodman, 1979; Vermunt, 1996b, 1997) or one the multi-dimensional Newton methods. The first chapters of this manual give many examples on the use of user-defined designs.

The complete syntax of `cov(..)` and `fac(..)` is

```
cov(<margins>,<# of effects>,<group margin>,<a/b/c>,<# of groups>)
fac(<margins>,<# of effects>,<group margin>,<a/b/c>,<# of groups>).
```

The parameter `<margins>` indicates the margins for which the user-defined design will be specified. The margins are separated by a space or a comma. The fact that one may specify more than one margin makes it possible to impose restrictions between parameters belonging to different sets of variables.

The second parameter concerns the number of effects, or the number of (log-linear) parameters.

The third parameters is `<group margin>`. The optional specification of a group margin makes it possible let parameters vary among levels of some other variables, that is, to specify higher-order effects. If such a set of grouping variables is used, also the type of interaction has to be specified with a letter: `a`, `b`, or `c`. The letter `a` means that there is no higher-order interaction, which is the same as not using a grouping variable at all. Interaction type `b` yields a set of log-multiplicative scaling factors for the (joint) grouping variable (Xie, 1992; Vermunt, 1996b, 1997). And finally, `c` yields a standard higher-order interaction effect. These three interaction types is sometimes referred to as homogeneous, simple heterogeneous, and heterogeneous models. It should be noted that for identification the first scaling factor is always fixed to one in simple heterogeneous models. In heterogeneous models, no identifying restrictions are imposed on the higher-order interaction terms.

The last parameter, `<# of groups>`, makes it is possible to further restrict the higher-order interactions introduced by including a grouping variables. A positive number defines the number of different groups, while a negative number indicates the number of interval designs that will be defined for the (joint) grouping variable.

The user-defined designs, including the possible restrictions on the group margin, must be specified with the command `des`. For each user-defined design, the program will first read the designs for each of the effects for the first margin, then for the second margin, etc. Let $N$ be the number of margins in the user-defined design concerned, $T_n$ the number of categories in margin $n$, and $K$ the number of effects in the user-defined design concerned. For `cov(..)`, $\ell_{EM}$ expects $K$ effects consisting of $T_n$ numbers for each $n$. Thus, the design for a `cov(..)` statement will consist of $\sum_n^N KT_n$ numbers.

For `fac(..)`, $\ell_{EM}$ expects $T_n$ numbers for each $n$, where the numbers must be integers ranging from 0 to $K$. A 0 means that no effect is specified for the marginal cell concerned. The other numbers indicate to which of the $K$ effects a particular marginal cell contributes.

If restrictions are imposed on the (joint) grouping variable, a design for the grouping variable has to given after the design for the effects. If a positive value is specified for `<# of groups>`, the design matrix must contain one number for each category of the (joint) group variable, where a 0 means no effect for the group concerned and a number between 1 and the specified number groups indicates to which subgroup each of the cells in group the margin belongs. If a negative

value is specified for the number of groups, an interval level design is expected for the grouping variable. The number of entries of this design equals the absolute value of the specified number of groups times the number of cells in the group margin.

### 11.5.3 Predefined designs

Besides the possible of specifying a design matrix with `cov(..)` or `fac(..)`, the $\ell_{EM}$ program contains predefined designs for a number of common non-hierarchical (log-linear) effects. These predefined design can be called with the command `spe(..)`. The complete syntax of this command, which can be used between the parentheses of log-linear models, logit models, regression models with cumulative link functions, and hazard models, is

```
spe(<margins>,<type of effect>,<group margin>,<a/b/c>,<# of groups>).
```

The specification of `<margins>` and higher-order interaction terms via (joint) grouping variable is the same as in `cov(..)` and `fac(..)`. The difference between the specification of predefined compared to user-defined designs is the parameter `<type of effect>`, which replaces the parameter `<# of effects>`. This parameter indicates the type of log-linear interaction that one wants to include in the (sub)model concerned. The possible values of `<type of effect>` are:

- 1a. standard log-linear parameters,
- 1b. linearly restricted log-linear parameters,
- 2a. total-score parameters,
- 2b. linearly restricted total-score parameter,
- 3a. symmetric association parameters,
- 4a. symmetric association parameters without main diagonal,
- 5a. main diagonal parameters,
- 6a. ranking parameters with objects as variables,
- 7a. ranking parameters with rankings as variables,
- 8a. difference parameters,
- 8b. linearly restricted difference parameter,
- 9a. absolute-difference parameters,
- 9b. linearly restricted absolute-difference parameters.

The parameter of these predefined design by means of the uni-dimensional Newton algorithm or one of the multi-dimensional Newton methods.

### 11.5.4 Association models

Another type of interaction terms that can be specified with $\ell_{EM}$ are row-column association models (Goodman, 1979; Clogg, 1982; Xie, 1992; Clogg and Shihadeh, 1994; Vermunt, 1996b, 1997). These models for restricting bivariate associations can be defined by the commands `ass1(..)`, `ass2(..)`, and `ass3(..)`, which may be used within the parentheses (`{..}`) of log-linear, logit, or hazard models. Association models are estimated by means of uni-dimensional Newton.

The command `ass1(..)` yields log-linear association models, `ass2(..)` log-multiplicative association models, and `ass3(..)` log-linear association models with a set log-multiplicative scaling factors for a grouping variable. Note that the association models restrict the two-variable interaction terms. This implies that, generally, also the one-variable effects for the row and the column variable must be incorporated in the model.

In the `ass1(.)` and `ass3(..)` interactions, the fixed row and column scores have a mean of 0 and a mutual distance of 1. The log-linear association parameters are identified by letting them sum to zero. In RC models specified with `ass2(.)`, the row and column scores are scaled in such a way that their sum is 0 and their sum of squares is 1. This default setting can be changed with the command `sca`. In RC(M) models, the scores for different dimensions are orthogonalized by means of a singular value decomposition (Becker, 1990).[1]

The complete syntax of the three association commands is:

```
ass1/2/3(<row margin>,<column margin>,<group margin>,<type of model>,
         <type of symmetry>,<# of rows>,<# of columns>,<# of groups>)
```

First, the (joint) row and column variables must be specified. The optional specification of a (joint) group variable makes is possible to test whether some of the parameters differ among subgroups.

The fourth parameter `<type of model>` consists of a combination of a number and a letter, for example, `4a`. The meaning of the numbers is the following:

- 2. linear-by-linear or uniform association,
- 3. row association,
- 4. column association,
- 5. row and column association,
- 6. equal row and column association.

Thus, the number specifies the type of association model. Models 2 to 5 are the same as in the article of Clogg (1982). In model 6, the row and column parameters are constraint to be equal.

The letter is used to denote the type of interaction with the grouping variable, that is,

- a. homogeneous (2,3,4,5,6),
- b. simple heterogeneous (2,3,4,5,6),
- c. heterogeneous row and/or column (3,4,5,6),
- d. heterogeneous column (5),
- e. heterogeneous row and column (5).

The numbers between braces indicate the types of association models for which the letter concerned is relevant. In homogeneous models, the row and/or column parameters are assumed to be equal between groups, while heterogeneous indicates that they differ per group. Simple heterogeneous is situated between homogeneous and heterogeneous: the row and/or column parameters are equal for every group, but the association parameter, which can be log-linear (`ass1(..)`) or log-multiplicative (`ass2(..)` and `ass3(..)`) differs per group.

Equality constraints on the row and/or column parameters in different partial associations can be imposed by means of `<symmetry>`.[2] The `<symmetry>` parameter can take one of three values:

[1] It should be noted that a RC(M) model has to be specified by using the command `ass2(..)` M times.

[2] Clogg (1982) used the term symmetry for models with row and/or column scores which are equal in different partial association.

- a. symmetric row and/or column parameters (3,4,5,6),
- b. symmetric row parameters (5),
- c. symmetric column parameters (5).

If this option is used, the row and/or column parameters for the partial association concerned are restricted to be equal to the parameters for the same variable in the other partial association where `<symmetry>` is used as well.

The last three optional parameters – `<# of rows>`, `<# of columns>`, and `<# of groups>` – can be used to further restrict the row, column, and group parameters. Their functioning is similar to the specification of the number of groups in `cov(..)` and `fac(..)`. They can be used to specify nominal or interval designs for the row, column, and group variables. Equality restrictions involve specifying the number of different rows, columns, or groups with a positive integer. A negative integer indicates that an interval design will be given for the set of parameters concerned, where the absolute value of the specified number is the number of effects. And finally, a 0 means that the set of parameters concerned is not restricted. If these options are used in combination with `<symmetry>`, the restrictions must be specified in the first partial association where the row or column concerned appears.

The nominal or interval designs for the row, column, or group variable are read from the design matrix specified with `des`. They are read in the order in which they are specified, after the user-defined designs for the submodel concerned. A nominal design for restricted rows or columns consists of numbers between 1 and the number of different scores, where equal numbers indicate that rows and columns are equal. A nominal design for the interaction with the group variable consist of numbers from 0 to the number of different groups, where a 0 means that no association parameter has to be estimated for the group concerned. An interval level design for the row, column, or group variable consists the specified number of effects times the number of cells in the margin concerned numbers.

### 11.5.5 Weight vectors

A weight vector can be used for several purposes, such as the specification exposure times and risk populations, fixed effects, structural zeros, and cell weights for a weighted analysis. In $\ell_{EM}$, the use of a weight vector involves two steps. First, with the command `wei(<margin>)` between the parentheses of a log-linear, logit, or hazard model, it has to be indicated that a set of cell weights will be specified for the margin concerned. The second step involves specification of the cell weights as 'starting values', that is, by means of the command `sta`.

The only necessary modification of the estimation algorithm in situations in which there are cell weights is that the cell weights have to be used as starting values for the expected cell frequencies. The estimation of the other parameters proceeds in the usual way.

### 11.5.6 Linear restrictions

Linear restrictions on cell frequencies and proportions can be imposed by the command `lin(..)`, which complete syntax is

```
lin(<margins>,<# of constraints>).
```

We have to specify the margins involved in the linear constraints and the number of constraints. The contrasts, which indicate which linear combinations of (marginal) cells sum to zero, have to be given in the design matrix.

Maximum likelihood estimates of the cell frequencies or proportions under these types of constraints are obtained by solving the Lagranger equation with a simple uni-dimensional Newton algorithm.

### 11.5.7 Cumulative link functions

Regression models for ordinal dependent variables based on cumulative link functions can be specified with submodels of the form

```
<probability> cum(<type>) {<parameters>}.
```

First, we have to give the conditional probability for which a regression model will be specified. With `cum(<type>)`, the type of cumulative link function must be specified. The parameter `<type>` can take one of nine values,

- a. logit,
- b. probit,
- c. complementary log-log,
- d. log-log,
- e. logit with equidistant thresholds,
- f. probit with equidistant thresholds,
- g. complementary log-log with equidistant thresholds,
- h. log-log with equidistant thresholds,
- i. linear regression.

The parameters of the model concerned have to be specified between parentheses using the commands `cov(..)`, `fac(..)`, and `spe(..)`. The threshold parameters are automatically included in cumulative models.

The parameters of regression models with cumulative link function may either be estimated with uni-dimensional Newton or one of the multi-dimensional Newton algorithms.

### 11.5.8 Equal submodels

The complete set of parameters of a specific submodel can be made equal to the parameters of another submodel by means of the command `eq1`. This command is used as follows:

```
<probability 1> eq1 <probability 2>.
```

In fact, we indicate that the complete probability set 2 is equal to probability set 1. It should be noted that this is only possible if the two sets of probabilities have exactly the same structure, that is, if the order and the number of the categories of the dependent and independent variables is the same.

Imposing this type of equality constraints just involves pooling the data for the sets of probabilities concerned (Vermunt, 1997).

### 11.5.9 Equalities and fixed-values restrictions on probabilities

With the command `eq2` one can specify equality and fixed-value restrictions on probabilities. This command is used in a model after the specification of a probability, that is,

```
<probability> eq2.
```

In the designs matrix, we can indicate which probabilities are free, which ones are restricted to be equal, and which ones are restricted to a specific value. Free parameters are denoted by a 0, fixed values by a -1, and equalities by positive integers, where equal numbers indicate that parameters are equal. It should be noted that equality restrictions cannot only be imposed within a submodel, but also across submodels. For the fixed-value restrictions, one also has to supply starting values. These probabilities will be fixed to their starting values.

The algorithm implemented in $\ell_{EM}$ to obtain maximum likelihood estimates for probabilities under this type of equality and fixed-value restrictions is based on the Lagranger likelihood equations given by Mooijaart and Van der Heijden (1992). As explained in Vermunt (1997), these likelihood equations can be solved by means of uni-dimensional Newton.

### 11.5.10 Ordinal restrictions on probabilities

The commands `or1` and `or2` can be used to impose nonparametric ordinal restrictions on conditional probabilities consisting of one dependent and one independent variable. With

```
<probability> or1/or2,
```

one can request such an ordinal (sub)model. Note that `or1` indicates a positive and `or2` a negative relationship between the two variables.

Maximum likelihood estimates for probabilities under these types of nonparametric order restrictions are obtained by an uni-dimensional Newton algorithm which activates an equality constraint appearing in the Lagranger likelihood function when the inequality constraint concerned is violated. At each iteration it is checked whether the active constraints can be deactivated.

### 11.5.11 Correspondence analysis

Besides for estimating log-linear models, regression models, and path models with and without latent variables, the $\ell_{EM}$ program can be used for performing correspondence analysis and some variants of it. The solution for these types of analyses is obtained by means of singular value decomposition of specific deviation matrices (Greenacre, 1984; Gifi, 1990).

The specification of such an analysis is done as follows:

```
mod cor(<type>,<# dim. variables/categories>,<# dim. objects>).
```

Thus, rather than specifying any other type of model after the `mod` statement, we indicate that we want to perform correspondence analysis. The parameter `<type>` can have one of four different values, that is,

- 1. correspondence or (canonical) correlation analysis,
- 2. multiple correspondence analysis,
- 3. association analysis with marginal weights,
- 4. association analysis with uniform weights.

It is also possible to omit all parameters, that is, to specify just `mod cor`. In that case, the program will assume type `1` for two-way tables and type `2` for higher-way tables.

The second parameter makes it possible to indicate the number of dimensions for which one wants output on variables and categories. The last parameter, which is only relevant in combination with type `2`, can be used to indicate the number of dimensions for which one wants objects scores. Note that a value of zero suppresses the computation of object scores. The default number of dimensions for both output parts is two.

# Bibliography


Agresti, A. (1990). *Categorical data analysis*. New York: Wiley.

Agresti, A. (1993). Computing conditional maximum likelihood estimates for generalized Rasch models using simple loglinear models with diagonal parameters. *Scandinavian Journal of Statistics*, 20, 63-71.

Aldrich, J.H., and Nelson, F.D. (1984). *Linear probability, logit, and probit models*. Beverly Hills, London: Sage Publications Inc..

Allison, P.D. (1982). Discrete-time methods for the analysis of event histories. S. Leinhardt (ed.), *Sociological Methodology 1982*, 61-98. San Francisco: Jossey-Bass.

Baker, S.G. (1994). Regression analysis of grouped survival data with incomplete covariates: nonignorable missing-data and censoring mechanisms. *Biometrics*, 50, 821-826.

Baker, S.G., and Laird, N.M. (1988). Regression analysis for categorical variables with outcome subject to nonignorable nonresponse. *Journal of the American Statistical Association*, 83, 62-69.

Baker, S.G., Wax, Y., and Patterson, B.H. (1993). Regression analysis of grouped survival data: informative censoring and double sampling. *Biometrics*, 49, 379-389.

Becker, M.P. (1990). Algorithm AS253: Maximum likelihood estimation of the RC(M) association model. *Applied Statistics*, 39, 152-167.

Becker, M.P., and Clogg, C.C. (1989). Analysis of sets of two-way contingency tables using association models. *Journal of the American Statistical Association*, 84, 142-151.

Bergsma, W. (1997). *Marginal models for categorical variables*. Tilburg: Tilburg University Press.

Bishop, R.J., Fienberg, S.E., and Holland, P.W. (1975). *Discrete multivariate analysis: theory and practice*. Cambridge, Mass.: MIT Press.

Blossfeld, H.P., and Rohwer, G. (1995). *Techniques of event history modeling*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Böckenholt, U., and Langeheine, R. (1996). Latent change in recurrent choice data. *Psychometrika*, 61, 285-302.

Bye, B.V., Gallicchio, S.J., and Dykacz, J.M. (1985). Multiple-indicator, multiple-cause models for a single latent variable with ordinal indicators. *Sociological Methods and Research*, 13, 487-509.

Clogg, C.C. (1979). Some latent structure models for the analysis of Likert-type data. *Social Sciences Research*, 8, 287-301.

Clogg, C.C. (1981). New developments in latent structure analysis. D.J. Jackson and E.F. Borgotta (eds.), *Factor analysis and measurement in sociological research*, 215-246. Beverly Hills: Sage Publications.

Clogg, C.C. (1982). Some models for the analysis of association in multiway cross-classifications having ordered categories. *Journal of the American Statistical Association*, 77, 803-815.

Clogg, C.C., and Eliason, S.R. (1987). Some common problems in log-linear analysis. *Sociological Methods and Research*, 16, 8-14.

Clogg, C.C., and Goodman, L.A. (1985). Simultaneous latent structure analysis in several groups. N.B. Tuma (ed.), *Sociological Methodology 1985*, 81-110. San Fransisco: Jossey-Bass.

Clogg, C.C., and Shihadeh, E.S. (1994). *Statistical models for ordinal data*. Thousand Oakes, CA: Sage Publications.

Conaway, M.R. (1992). The analysis of repeated categorical measurements subject to nonignorable nonresponse. *Journal of the American Statistical Association*, 87, 817-824.

Christoffersen, A. (1975). Factor analysis of dichotomised variables. *Psychometrika*, 40, 5-32.

Croon, M.A. (1989). Latent class models for the analysis of rankings. G. De Soete, H. Feger, and K.C. Klauer (eds.), *New developments in psychological choice modeling*, 99-121. Elsevier Science Publishers B.V. (North-Holland).

Croon, M. (1990). Latent class analysis with ordered latent classes. *British Journal of Mathematical and Statistical Psychology*, 43, 171-192.

Croon, M.A., and Luijkx, R. (1993). Latent structure models for ranking data. M.A. Fligner and J.S. Verducci (eds.), *Probability models and statistical analysis of ranking data*, 53-74. New York: Springer-Verlag.

Dayton, C.M., and Macready, G.B. (1988). Concomitant-variable latent-class models. *Journal of the American Statistical Assocaition*, 83, 173-178.

De Leeuw, J., Van der Heijden, P.G.M., and Verboon, P. (1990). A latent time-budget model. *Statitistical Neerlandica*, 44, 1-22.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Ser. B.*, 39, 1-38.

Farewell, V.T. (1982). The use of mixture models for the analysis of survival data with long-term survivors. *Biometrics*, 38, 1041-1046.

Fay, R.E. (1986). Causal models for patterns of nonresponse. *Journal of the American Statistical Association*, 81, 354-365.

Fienberg, S.E., and Mason, W. (1979). Identification and estimation of age-period-cohort models in the analysis of discrete archival data. *Sociological Methodology 1979*, 1-67.

Formann, A.K. (1992). Linear logistic latent class analysis for polytomous data. *Journal of the American Statistical Association*, 87, 476-486.

Fuchs, C. (1982). Maximum likelihood estimation and model selection in contingency tables with missing data. *Journal of the American Statistical Association*, 77, 270-278.

Ganzeboom, H.B.G., and Luijkx, R. (1995). Intergenerationele beroepsmobiliteit in Nederland: patronen en historische veranderingen. J. Dronkers en W.C. Ultee (eds.), *Verschuivende ongelijkheid in Nederland*, 14-30. Assen: Van Gorcum.

Gifi, A. (1990). *Nonlinear multivariate analysis*. New York: John Wiley.

Gilula, Z., and Haberman, S.J. (1994). Conditional log-linear models for analyzing panel data. *Journal of the American Statistical Association*, 89, 645-656.

Gilula, Z., and Haberman, S.J. (1995). Dispersion of categorical variables and penalty functions: derivation, estimation, and comparability. *Journal of the American Statistical Association*, 90, 1447-1452.

Gong, G., Whittemore, A.S., and Grosser, S. (1990). Censored survival data with misclassified covariates: a case study of breast-cancer mortality. *Journal of the American Statistiical Association*, 85, 20-28.

Goodman, L.A. (1973). The analysis of multidimensional contingenty tables when some variables are posterior to others: a modified path analysis approach. *Biometrika*, 60, 179-192.

Goodman, L.A. (1974). Exploratory latent structure analysis using both indentifiable and unidentifiable models. *Biometrika*, 61, 215-231.

Goodman, L.A. (1978). *Analysing qualitative/categorical variables: loglinear models and latent structure analysis*. Cambridge: Abt.

Goodman, L.A. (1979). Simple models for the analysis of association in cross-classifications having ordered categories. *Journal of the American Statistical Association*, 74, 537-552.

Goodman, L.A. (1986). Some useful extensions of the usual correspondence analysis approach and the usual log-linear approach in the analysis of contingency tables. *International Statistical Review*, 54, 243-309.

Goodman, L.A. (1991). Measures, models, and graphical displays in the analysis of cross-classified data. *Journal of the American Statistical Association*, 86, 1085-1111.

Greenacre, M.J. (1984). *Theory and applications of correspondence analysis*. London: Academic Press.

Guo, G. (1993). Event-history analysis for left-truncated data. P.V. Marsden (ed.), *Sociological Methodology 1993*, 217-243. Oxford: Basil Blackwell.

Guo, G., and Rodriguez, G. (1994). Estimating a multivariate proportional hazards model for clustered data using the EM algorithm, with an application to child survival in Guatemala. *Journal of the American Statistical Association*, 87, 969-976.

Haber, M., and Brown, M.B. (1986). Maximum likelihood methods for log-linear models when expected frequencies are subject to linear contraints. *Journal of the American Statistical Association*, 81, 477-482.

Haberman, S.J. (1978). *Analysis of qualitative data, Vol. 1, Introduction topics*. New York, San Francisco, London: Academic Press.

Haberman, S.J. (1979). *Analysis of qualitative data, Vol 2, New developments*. New York: Academic Press.

Hagenaars, J.A. (1986). Symmetry, quasi-symmetry and marginal homogeneity on the latent level. *Social Science Research*, 15, 241-255.

Hagenaars, J.A. (1988). Latent structure models with direct effects between indicators: local dependence models. *Sociological Methods and Research*, 16, 379-405.

Hagenaars, J.A. (1990). *Categorical longitudinal data - loglinear analysis of panel, trend and cohort data.*. Newbury Park: Sage.

Hagenaars, J.A. (1993). *Loglinear models with latent variables*. Newbury Park: CA: Sage.

Heckman, J.J., and Singer, B. (1982). Population heterogeneity in demographic models. K. Land and A. Rogers (eds.), *Multidimensional mathematical demography*. New York: Academic Press.

Heckman, J.J., and Singer, B. (1984). The identifiability of the proportional hazard model. *Review of Economic Studies*, 51, 231-241.

Heinen, T. (1996). *Latent class and discrete latent trait models: similarities and differences*. Thousand Oakes: Sage Publications.

Jöreskog, K.G., and Sörbom, D. (1988). *Lisrel 7: a guide to the programm and applications*.

Kamakura, W.A., Wedel, M., and Agrawal, J. (1992). *Concomitant variable latent class models for the external analysis of choice data*. University of Groningen, The Netherlands: Research Memorandum nr 486, Institute of Economic Research.

Kelderman, H. (1984). Log-linear Rasch model tests. *Psychometrika*, 49, 223-245.

Knoke, D., and Burke, P.J. (1980). *Log-linear models, Sage University Paper 20*. Beverly Hills: Sage publications.

Laird, N., and Oliver, D. (1981). Covariance analysis of censored survival data using log-linear analysis techniques. *Journal of the American Statistical Association*, 76, 231-240.

Lang, J.B., and Agresti, A. (1994). Simultaneously modeling joint and marginal distributions of multivariate categorical responses. *Journal of the American Statistical Association*, 89, 625-632.

Langeheine, R., and Van de Pol, F. (1990). A unifying framework for Markov modeling in discrete space and discrete time. *Sociological Methods and Research*, 18, 416-441.

Langeheine, R., and Van de Pol, F. (1994). Discrete-time mixed Markov latent class models. A. Dale and R.B. Davies (eds.), *Analyzing social and political change: a casebook of methods*, 171-197.

Larson, M.G. (1984). Covariate analysis of competing-risk data with log-linear models. *Biometrics*, 40, 459-469.

Lawless, J.F. (1982). *Statistical models and methods for lifetime data*. New York: Wiley.

Lazarsfeld, P.F., and Henry, N.W. (1968). *Latent structure analysis*. Boston: Houghton Mill.

Lindsay, B., Clogg, C.C., and Grego, J. (1991). Semiparametric estimation in the Rasch model and related models, including a simple latent class model for item analysis. *Journal of the American Statistical Association*, 86, 96-107.

Little, R.J., and Rubin, D.B. (1987). *Statistical analysis with missing data*. New York: Wiley.

Luijkx, R. (1994). *Comparative loglinear analyses of social mobility and heterogamy*. Tilburg: Tilburg University Press.

Maddala, G.S. (1983). *Limited dependent and qualitative variables in econometrics*. Cambridge: Cambridge University Press.

Magidson, J. (1981). Qualitative variance, entropy, and correlation ratios for nominal dependent variables. *Social Science Research*, 10, 177-194.

Mare, R.D. (1994). Discrete-time bivariate hazards with unobserved heterogeneity: a partially observed contingency table approach. P.V. Marsden (ed.), *Sociological Methodology 1994*, 341-385. Oxford: Basil Blackwell.

McCullagh, P., and Nelder, J.A. (1989). *Generalized linear models*. London: Chapman & Hall, second edition.

McCutcheon, A.L. (1987). *Latent class analysis, Sage University Paper*. Newbury Park: Sagee Publications.

McCutcheon, A.L. (1988). Sexual morality, pro-life values and attitudes toward abortion. *Sociological Methods and Research*, 16, 256-275.

Mellenbergh, G.J., and Vijn, P. (1981). The Rasch model as a log-linear model. *Applied Psychological Measurement*, 5, 369-376.

Meng, X.L., and Rubin, D.B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80, 267-278.

Mislevy, R.J. (1985). Estimation of latent group effects. *Journal of the American Statistical Association*, 80, 993-997.

Mislevy, R.J. (1986). Recent developments in the factor analysis of categorical variables. *Journal of Educational Statistics*, 11, 3-31.

Mislevy, R.J., and Bock, R.D. (1990). *BILOG 3: Item analysis and test scoring with binary logistic models*. Mooresville: Scientific Software, Inc..

Mooijaart, A., and Van der Heijden, P.G.M. (1992). The EM algorithm for latent class models with constraints. *Psychometrica*, 57, 261-271.

Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical and continuous latent variable indicators. *Psychometrika*, 49, 115-132.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1986). *Numerical recipes: The art of scientific computing*. Cambridge: Cambridge University Press.

Poulsen, C.A. (1982). *Latent structure analysis with choice modelling*. Aarhus: Aarhus School of Business Administratioon and Economics.

Proctor, C.H. (1970). A probabilistic formulation and statistical analysis of Guttman scaling. *Psychometrika*, 35, 73-78.

Rindskopf, D. (1990). Nonstandard loglinear models. *Psychological Bulletin*, 108, 150-162.

Rost, J. (1990). Rasch models in latent classes: an integration of two approaches to item analysis. *Journal of Applied Psychological Measurement*, 14, 271-282.

Rohwer, G. (1993). *TDA Working Papers*.

Schluchter, M.D., and Jackson, K.L. (1989). Log-linear analysis of censored survival data with partially observed covariates. *Journal of the American Statistical Association*, 84, 42-52.

Takane, Y., and De Leeuw, J. (1987). On the relationship between item response theory and factor analysis of dicrete variables. *Psychometrika*, 52, 393-408.

Thissen, D. (1988). *MULTILOG: Multiple, categorical item analysis and test scoring using item response theory*. Mooresville: Scientific Software, Inc..

Tuma, N.B., and Hannan, M.T. (1984). *Social dynamics: models and methods*. New York: Academic Press.

Van de Pol, F., and De Leeuw, J. (1986). A latent Markov model to correct for measurement error. *Sociological Methods and Research*, 15, 118-141.

Van de Pol, F., and Langeheine, R. (1990). Mixed Markov latent class models. C.C. Clogg (ed.), *Sociological Methodology 1990*. Oxford: Basil Blackwell.

Van der Heijden, P.G.M., and Dessens, J. (1994). *Incorporating continuous explanatory variables in latent class analysis*. Utrecht: Methods Series MS-94-5, Utrecht University.

Van der Heijden, P.G.M, Mooijaart, A., and De Leeuw, J. (1992). Constraint latent budget analysis. *Sociological Methodology 1992*.

Vermunt, J.K. (1996a). Causal log-linear modeling with latent variables and missing data. U.Engel and J. Reinecke (eds.), *Analisis of change: advanced techniques in panel data analysis*, 35-60. Berlin/New York: Walter de Gruyter.

Vermunt, J.K. (1996b). *Log-linear event history analysis: a general approach with missing data, unobserved heterogeneity, and latent variables*. Tilburg: Tilburg University Press.

Vermunt, J.K. (1997). *Log-linear models for event histories*. Thousand Oakes: Sage Publications.

Vermunt, J.K., and Georg, W. (1995). *Analyzing categorical panel data by means of causal log-linear models with latent variables: An application to the change in yougth-centrism*. WORC Paper 95.06.012/7.

Vermunt, J.K., Langeheine, R., and Böckenholt, U. (1995). *Discrete-time discrete-state latent Markov models with time-constant and time-varying covariates*. WORC Paper 95.06.013/7, Tilburg University.

Wedel, M., DeSarbo, W.S., Bult, J.R., and Ramaswamy, V. (1993). A latent class Poisson regression model for heterogeneous count data with an application to direct mail. *Journal of Applied Econometrics*, 8, 397-411.

Wermuth, N., and Lauritzen, S.L. (1983). Graphical and recursive models for contingency tables. *Biometrika*, 70, 537-552.

Wermuth, N., and Lauritzen, S.L. (1990). On substantive research hyphotheses, conditional independence graphs and graphical chain models. *Journal of the Royal Statistical Association B*, 52, 21-50.

Wiggins, L.M. (1973). *Panel analysis*. Amsterdam: Elsevier.

Willekens, F., and Shah, M.R. (1983). *A note on log-linear modelling of rates and proportions*. Voorburg: Working paper no. 36, NIDI.

Wong, R.S.K. (1995). Extensions in the use of log-multiplicative scaled association models in multiway contingency tables. *Sociological Methods and Research*, 23, 507-538.

Xie, Yu (1992). The log-multiplicative layer effects model for comparing mobility tables. *American Sociological Review*, 57, 380-395.

Yamaguchi, K. (1986). Alternative approaches to unobserved heterogeneity in the analysis of repeatable events. N.B. Tuma (ed.), *Sociological Methodology 1986*, 213-249. Washington, DC.: American Sociological Association.

Yamaguchi, K. (1991). Event history analysis. *Applied Social Research Methods, Volume 28*. Newbury Park: Sage Publications.

Zwinderman, A.H. (1991). A generalized Rasch model for manifest predictors. *Psychometrika*, 56, 589-600.

## Disclaimer and bugs

It should be emphasized that $\ell_{EM}$ is distributed without any warranty on the part of the author. Although the most important parts of program have been tested thoroughly, errors remain unavoidable. The author would be very grateful if the user would be kind enough to send a report of detected errors, enclosing input and data files. Any further suggestions for the improvement of the program are welcome too. Please send to:

Jeroen K. Vermunt
Department of Methodology and Statistics
Faculty of Social and Behavioural Sciences
Tilburg University
P.O. Box 90153
5000 LE Tilburg
The Netherlands
E-mail: j.k.vermunt@uvt.nl

## Referencing to $\ell_{EM}$

Development of the $\ell_{EM}$ program has been an enormous amount of work. I would therefore be grateful if you would add an appropriate reference to my work if you found $\ell_{EM}$ useful for your statistical analysis. When you report results obtained with $\ell_{EM}$, you should refer to this manual as "Vermunt, J.K. (1997). LEM: A General Program for the Analysis of Categorical Data. Department of Methodology and Statistics, Tilburg University" and/or to my published Ph.D. dissertation "Vermunt, J.K. (1997). Log-linear Models for Event Histories. Thousand Oakes: Sage Publications", which describes the models and algorithms implemented in $\ell_{EM}$.