LMest: An R Package for Estimating Generalized Latent Markov Models

by Fulvia Pennoni, Silvia Pandolfi, and Francesco Bartolucci

Abstract We provide a detailed overview of the updated version of the R package LMest, which offers functionalities for estimating Markov chain and latent or hidden Markov models for time series and longitudinal data. This overview includes a description of the modeling structure, maximum-likelihood estimation based on the Expectation-Maximization algorithm, and related issues. Practical applications of these models are illustrated using real and simulated data with both categorical and continuous responses. The latter are handled under the assumption of the Gaussian distribution given the latent process. When describing the main functions of the package, we refer to potential applicative contexts across various fields. The LMest package introduces several key novelties compared to previous versions. It now handles missing responses under the missing-at-random assumption and provides imputed values. The implemented functions allow users to display and visualize model results. Additionally, the package includes functions to perform parametric bootstrap for inferential procedures and to simulate data with complex structures in longitudinal contexts.

1 Introduction

Markov chain (MC) and latent or hidden Markov (HM) models are gaining increasing attention due to possibility of handling the temporal structure of many observed phenomena (Meyn and Tweedie, 2012; Mor et al., 2021). In particular, with only one categorical response, an MC model allows studying the initial distribution of this response variable and its conditional distribution given the previous response. On the other hand, HM models offer a practical model-based dynamic clustering and classification method (Bouveyron et al., 2019). This class of models finds application in the analysis of time-series (Ephraim and Merhav, 2002; Zucchini et al., 2016) and longitudinal data (Wiggins, 1973; Bartolucci et al., 2013, 2022). The model formulation involves the introduction of a sequence of individual and time specific discrete latent (hidden) variables. This results in a hidden process assumed to follow a Markov chain of first order. The states of this chain correspond to latent clusters or subpopulations of individuals with similar latent characteristics. The approach can handle responses of different types, whether continuous or categorical. In the latter case, the conditional distribution of these variables is freely parameterized on the basis of conditional response probabilities.

In this article we provide a practical introduction to MC and HM models through an overview of the **LMest** package, focusing in particular on the new features of this package compared to the previous versions, such as the one illustrated in Bartolucci et al. (2017). The package allows analyzing time-series and longitudinal data by the models at issues. In the longitudinal context, recurring measurements over time on possibly multiple characteristics are taken on several sampling units (e.g., individuals), in such a way that it is possible to describe the evolution of these characteristics over time.

The package in its updated version 3.2.5 is available on CRAN at **LMest** and it is also described with detailed vignettes, including applicative examples. Each function is well documented in the help provided within the package, and several datasets from surveys and other sources are included in the package in both wide and long formats.

The current version of the package has the following features, which were also present in the previous versions:

- it is designed to estimate different model formulations, such as MC, HM, and mixed HM models, through the Expectation-Maximization (EM) algorithm (Dempster et al., 1977);
- it allows the estimation of the effect of covariates under different parameterizations and model specifications;
- model selection procedures using the Akaike Information Criterion (AIC, Akaike, 1973) and the Bayesian Information Criterion (BIC, Schwarz, 1978) are included, relying on different parameter initialization strategies;
- functions to produce local and global decoding are implemented to perform dynamic clustering;
- standard errors for the parameter estimates are obtained either through exact computation or reliable approximations of the observed information matrix; parametric bootstrap procedures (Davison and Hinkley, 1997) are also available for different model specifications;
- in order to increase the computational efficiency, Fortran routines are used to perform numerical operations.

Some important extensions have been introduced in the most recent version of the package. The features of these new functions can be summarized as follows:

- in addition to categorical data, the package can handle continuous data, assuming a Gaussian distribution for the response variables conditional on the latent process. It also accommodates missing responses, drop-out, and intermittent missingness under the missing-at-random assumption (MAR, Little and Rubin, 2020);
- when dealing with continuous outcomes, covariates may be included under different model specifications, similar to the approach used for categorical outcomes;
- simulations from almost all the available model specifications may be carried out through a suitable method;
- data in long format can be provided as input, and the model specification follows the common R style; formulas for the response variables, as well as for the initial and transition probabilities can be specified using the package **Formula** (Zeileis and Croissant, 2010);
- parameter estimates can be graphically displayed to enhance the interpretation of the results.

Overall, the current version of the package offers great flexibility in model formulation and estimation.

In the following sections, after covering the main theoretical aspects of the models, we provide an overview of the **LMest** software package through examples using univariate and multivariate data with and without missing values. The package uses the S3 object-oriented system, defining three main classes of objects. We use both synthetic and real data to illustrate the models, where the synthetic data are generated using functions available within the package. These data are similar to those used in previous applied works published in the authors' research articles or by other researchers in the field. However, we prefer to use simulated versions because the original data are not always publicly available. Additionally, using such data allows us to incorporate specific features that are useful for illustrating certain functions of the package. When describing the data, we will refer to potential real-world applications to characterize the research questions of interest, thereby better motivating the adopted models. Specifically:

- the MC model is used to analyze longitudinal data about labor market careers after graduation;
- the HM model for categorical longitudinal responses is adopted to examine the customer's purchase behavior over time;
- the HM model for continuous time-series data is used to discover financial market phases;
- the HM model for continuous longitudinal responses is used to investigate the state progression of illness in patients after treatment and the progression of students performance in different types of school.

In the R examples we use `set.seed(x)` both when we generate the data and fit the models so that all the output can be replicated. The code for replicability of the simulated data can be provided upon request.

In the following we introduce the models, examples, and codes in an expository style to demonstrate the use of specific functions of **LMest**. More details about MC and HM models are provided in Bartolucci et al. (2013, 2014b). The remainder of the paper is structured as follows: the next section presents the MC model, while the third section introduces the HM model for categorical longitudinal data. The fourth section presents the HM model for continuous responses, which allows for missing values and individual covariates. Finally, the fifth section provides a summary, mentions some of other similar packages, and discusses additional issues related to future package extensions.

2 Markov chain model

In the following we first illustrate the assumptions of the MC model for categorical responses and then an application based on a longitudinal categorical response and covariates.

2.1 Model assumptions

In the context of longitudinal data, the MC model is referred to as the transition model (Anderson, 1954) since it is of interest to estimate the transition between states of the stochastic process. With only one categorical response variable, let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{iT})'$ denote the vector of such variables for individual i, with $i = 1, \ldots, n$, where Y_{it} has k categories labeled from 1 to k. Let $\boldsymbol{x}_{it}$ denote the vector of individual covariates available at the t-th time occasion for individual i, with $t = 1, \ldots, T$.

The main assumption of the model is that, for $t = 3, \ldots, T$, Y_{it} is conditionally independent of $Y_{i1}, \ldots, Y_{i,t-2}$ given $Y_{i,t-1}$ when a first-order dependence structure is formulated. Moreover, in its basic version, the model is characterized by initial and transition probabilities that, for every $i = 1, \ldots, n$, are denoted by

$$\pi_u = p(Y_{i1} = u), \quad u = 1, \ldots, k$$

and

$$\pi_{uv} = p(Y_{it} = v | Y_{i,t-1} = u), \quad t = 2, \ldots, T, \; u, v = 1, \ldots, k,$$

respectively.

In presence of individual time-fixed and time-varying covariates, the initial and transition probabilities are denoted as $\pi_{i,u} = p(Y_{i1} = u | \boldsymbol{x}_{i1})$, $u = 1, \ldots, k$, and $\pi_{it,uv} = p(Y_{it} = v | Y_{i,t-1} = u, \boldsymbol{x}_{it})$, $t = 2, \ldots, T$, $u, v = 1, \ldots, k$, respectively. Note that these probabilities are now individual specific, and their dependence on the vector of covariates is formulated on the basis of multinomial logit models (Azzalini, 1994). In particular, for the initial probabilities we have

$$\log \frac{\pi_{i,u}}{\pi_{i,1}} = \beta_{0u} + \boldsymbol{x}_{i1}' \boldsymbol{\beta}_{1u}, \quad u = 2, \ldots, k. \tag{1}$$

Note that, as a reference category, the multinomial logit model in (1) has the first state. For the transition probabilities we assume

$$\log \frac{\pi_{it,uv}}{\pi_{it,uu}} = \gamma_{0uv} + \boldsymbol{x}_{it}' \boldsymbol{\gamma}_{1uv}, \quad t = 2, \ldots, T, \; u, v = 1, \ldots, k, \; u \neq v, \tag{2}$$

where the logits have, as reference state, that corresponding to the row of the transition matrix. In the above expressions, $\boldsymbol{\beta}_u = (\beta_{0u}, \boldsymbol{\beta}_{1u}')'$ and $\boldsymbol{\gamma}_{uv} = (\gamma_{0uv}, \boldsymbol{\gamma}_{1uv}')'$ are parameter vectors to be estimated.

Denoting by $\boldsymbol{\theta}$ the vector of all model parameters, the log-likelihood can be defined as

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} f(y_{i1}, \ldots, y_{iT} | \boldsymbol{x}_{i1}, \ldots \boldsymbol{x}_{iT}),$$

where $f(y_{i1}, \ldots, y_{iT} | \boldsymbol{x}_{i1}, \ldots \boldsymbol{x}_{iT})$ is the probability of the observed vector of response variables for unit i and y_{it} is a realization of Y_{it}. Under the assumptions formulated for the MC model, and in presence of individual covariates, this probability may be expressed as

$$f(y_{i1}, \ldots, y_{iT} | \boldsymbol{x}_{i1}, \ldots \boldsymbol{x}_{iT}) = \pi_{i,y_{i1}} \prod_{t=2}^{T} \pi_{it,y_{i,t-1}y_{it}}.$$

For the basic MC model without covariates, explicit formulas are available for the maximum likelihood estimation of the parameters. Under more complex models, for example when individual covariates are included as in Equations (1) and (2), an iterative algorithm, such as the Newton-Raphson or the Fisher-scoring, is required to maximize the above log-likelihood. As is well known, at each step of these algorithms, the parameter vector is updated by adding to its current value the inverse of the observed or expected information matrix, which is multiplied by the score vector, that is, the vector of the first derivatives of $\ell(\boldsymbol{\theta})$. These steps are performed until a suitable convergence criterion is satisfied. The corresponding asymptotic standard errors can be obtained in the usual way from the diagonal elements of the information matrix; see Bartolucci et al. (2014b) for more details.

Finally note that, although we illustrate the MC model only for the case of balanced longitudinal data, this model can obviously be applied also to unbalanced data, when we have a specific number of observations T_i for each individual i. However, for this aim, it is convenient to use the function of **LMest** for HM models under a specific constraint, as will be clarified in the following.

2.2 Application to longitudinal data with covariates

The present example illustrates an application of the MC model with a univariate categorical response and time-fixed covariates. The data provided within the package, named `data_employment_sim`, are simulated supposing they come from a survey context, where interviews are conducted among a nationally representative sample of graduates about their employment status after graduation; for analyses of similar data see Bartolucci and Pennoni (2011). In this scenario, the binary response variable (`emp`) is recorded for three time occasions corresponding to one, two, and three years after graduation. Individual covariates, which are assumed to be time-fixed, are the geographical location of the university where individuals graduated, assuming the country is divided into two main areas (`area`: 1 for South, 2 for North), the grade at graduation categorized into three levels (`grade`: 1 for low, 2 for medium, and 3 for high), and the indicator for parents' educational university qualification (`edu`:

1 for university degree, 0 otherwise). In simulating these data, we set the sample size equal to $n = 585$, and we consider $T = 3$ time occasions.

According to the simulated context, analyzing such data with the MC model allows us to identify the employment patterns immediately after graduation and evaluate trends over time. Additionally, the research question is whether the probability of employment in the first period after graduation depends on the geographical area or on the final degree grade. Then, we can explore if and how family background influences the probability of employment.

The type of data structures that can be given in input to the estimation function is in long format, that is, a data frame with one row for each combination of unit i and time occasion t, with $i = 1, \ldots, n$ and $t = 1, \ldots, T$, so that the total number of rows of this object is nT. All columns should have names; one column must correspond to the unit identifier, typically named with `id`, and another column must correspond to the time occasions, typically named with `time`.

The data stored in the long format can be described by the usual `summary()` method after they have been prepared by function `lmestData()`, as follows:

```
data("data_employment_sim")
dt <- lmestData(responsesFormula = emp ~ area + grade + edu,
                id = "id", time = "time",
                data = data_employment_sim)
```

In particular, the frequency distribution of the response variable for each wave is reported below:

```
summary(dt, dataSummary = "responses", varType = rep("d", ncol(dt$Y)))
```

```
#>
#> Data Info:
#> ----------
#>
#> Observations:        585
#> Time occasions:      3
#> Variables:           1
#>
#>
#> Proportion:
#> ----------
#>
#>  time    emp
#>  1:585   0:0.411396
#>  2:585   1:0.588604
#>  3:585
#>
#> Proportion by year:
#> ----------
#>
#>
#> Time =  1
#>
#>  time    emp
#>  1:585   0:0.4615385
#>          1:0.5384615
#>
#> Time =  2
#>
#>  time    emp
#>  2:585   0:0.4034188
#>          1:0.5965812
#>
#> Time =  3
#>
#>  time    emp
#>  3:585   0:0.3692308
#>          1:0.6307692
```

To estimate an MC model with covariates we can use the `lmestMc()` function. In this example, the interest is in estimating the effect, on the employment at the first interview, of the area where the students graduated and of their grade, and in evaluating the impact of the parent's educational level only for the periods following the first interview. Accordingly, covariates `area` and `grade` are included in `responsesFormula` argument so as to affect the initial probabilities, while `edu` is included in the parameterization of the transition probabilities. The covariates for initial and transition probabilities are separated by the symbol "`|`". Note that we use the **Formula** package for all formula operators (Zeileis and Croissant, 2010). Moreover, the argument `index` is required, which is a character vector to specify the name of the unit identifier (first element) and that of the time occasions (second element). Standard errors for the parameter estimates are provided by setting `out_se = TRUE` as in the following:

```
mod1 <- lmestMc(responsesFormula = emp ~
                as.factor(area)  + as.factor(grade) | as.factor(edu),
                index = c("id", "time"),
                data = dt, out_se = TRUE,
                output = TRUE, seed = 345)
```

Note that, in the above code, the data object `dt` returned by the `lmestData()` function can be directly passed to the estimation function. The tolerance level to check convergence of the EM algorithm and the maximum number of iterations of this algorithm are set by default as `tol = 10^-8` and `maxit = 1000`.

Using option `output = TRUE`, the `lmestMc()` function also returns the subject-specific estimated initial probabilities, collected in object `Piv`, and the estimated subject- and time-specific transition probabilities, collected in `PI`. To summarize these results, it is convenient to calculate suitable averages of these probabilities. Provided that the estimation output is stored in object `mod1`, for the initial probabilities we have:

```
print(round(colMeans(mod1$Piv), 3))
```

```
#>     0     1
#> 0.462 0.538
```

These estimates indicate that the 46% of individuals is unemployed at the beginning of the period of observation, that is, one year after graduation.

The array `PI` containing the estimates of the transition probabilities has dimension $2 \times 2 \times 585 \times 3$, where 2 is the number of response categories, 585 is the sample size, and 3 is the number of time occasions. The averaged transition probabilities, over all individuals and time occasions, are computed as follows starting from object `mod1` where such probabilities are stored:

```
print(round(apply(mod1$PI[,,,2:3], c(1,2), mean), 3))
```

```
#>          category
#> category     0     1
#>        0 0.761 0.239
#>        1 0.088 0.912
```

From these results we observe that unemployed individuals have a probability close to 0.76 of remaining without a job between two successive time occasions. Employed individuals have a higher persistence in the same employment status. The probability to find a job from one wave to another is around 0.24.

A summary of the output of the `lmestMc()` function including parameter estimates can be printed using the `summary()` method:

```
summary(mod1)
```

```
#> Call:
#> lmestMc(responsesFormula = emp ~ as.factor(area) + as.factor(grade) |
#>     as.factor(edu), data = dt, index = c("id", "time"), out_se = TRUE,
#>     output = TRUE, seed = 345)
#>
#> Coefficients:
#>
```

```
#>  Be - Parameters affecting the logit for the initial probabilities:
#>                    logit
#>                         2
#>   (Intercept)       -0.4576
#>   as.factor(area)2   0.4382
#>   as.factor(grade)2  0.1536
#>   as.factor(grade)3  1.6676
#>
#>  Standard errors for Be:
#>                    logit
#>                         2
#>   (Intercept)       0.1681
#>   as.factor(area)2  0.1847
#>   as.factor(grade)2 0.1988
#>   as.factor(grade)3 0.2529
#>
#>  p-values for Be:
#>                    logit
#>                        2
#>   (Intercept)       0.006
#>   as.factor(area)2  0.018
#>   as.factor(grade)2 0.440
#>   as.factor(grade)3 0.000
#>
#>  Ga - Parameters affecting the logit for the transition probabilities:
#>                   logit
#>                         1       2
#>   (Intercept)     -1.6015 -2.1789
#>   as.factor(edu)1  2.1834 -2.6251
#>
#>  Standard errors for Ga:
#>                   logit
#>                        1      2
#>   (Intercept)     0.1257 0.1423
#>   as.factor(edu)1 0.3128 1.0141
#>
#>  p-values for Ga:
#>                   logit
#>                    1    2
#>   (Intercept)     0 0.00
#>   as.factor(edu)1 0 0.01
```

Note that the `summary()` method also returns the standard errors for the parameter estimates and the corresponding significance level when option `out_se = TRUE` is included in the main estimation function. The argument `Be` returned by the function provides the estimated regression parameters of the logistic model for the probability of belonging to category 1 (employed) versus category 0 (unemployed) at the first interview. For interpretation, a higher degree grade positively affects the employment probability one year after graduation. Specifically, the log-odds ratios is equal to 0.154 for those with a medium grade and 1.668 for those with a high grade, with respect to individuals with a low grade, all the other covariates held fixed. Moreover, the log-odds ratio for `area` is positive, indicating that, at the beginning of the study, individuals who graduated from a university located in the North of the country are more likely to be employed than those who graduated from a university located in the South, with other covariates held constant.

Regarding the estimates referred to the transition probabilities reported in the output `Ga`, we may conclude that, apart for the intercept, a higher level of parents' education positively affects the probability of transition from the first to the second category (being employed after the first interview). In terms of odds ratio, we have

```
round(exp(mod1$Ga[2,1]), 3)
```

```
#> [1] 8.877
```

compared to individuals with low educated parents. Moreover, having highly educated parents has a negative effect on the transition from the second to the first category:

```
round(exp(mod1$Ga[2,2]),3)

#> [1] 0.072
```

3 Hidden markov models for categorical data

We outline the main notation and assumptions of the HM models useful to understand the output of the illustrated estimation functions of the **LMest** package.

3.1 Model assumptions

For a sample of n individuals and T_i time occasions, let $\boldsymbol{Y}_{it}$, $i = 1, \ldots, n$, $t = 1, \ldots, T_i$, be the observable vector of response variables with elements Y_{ijt}, $j = 1, \ldots, r$, where r denotes the number of response variables. Note that the number of time occasions T_i is specific for each individual i. In this way, we allow for unbalanced panels that may be due to a different number of time occasions for every individual. Let also $\boldsymbol{U}_i = (U_{i1}, \ldots, U_{iT_i})'$ denote the latent process for each individual i that affects the distribution of the response variables and assumed to follow a first-order Markov chain with a certain number of latent (or hidden) states equal to k. The Markov properties of U_{it} are the same as those illustrated in Section 2.1. However, here we use the notation U_{it}, which stands for "unobserved", to emphasize that it is latent/hidden, whereas Y_{it} in Section 2.1 is observable. The response variables are conditionally independent given this latent process; this assumption is referred to as *local independence*. It means that the latent process fully explains the underlying phenomenon together with possible covariates that may be time-fixed or time-varying.

The HM model is characterized by two components: the *measurement (sub)model*, which describes the conditional distribution of the response variables given the latent process, and the *structural or latent (sub)model*, which describes the distribution of the latent process. When dealing with categorical outcomes, the formulation of the measurement (sub)model without covariates is based on the parameters

$$\phi_{jy|u} = p(Y_{ijt} = y | U_{it} = u), \quad j = 1, \ldots, r, \; t = 1, \ldots, T_i, \; u = 1, \ldots, k, \; y = 0, \ldots, l_j - 1, \tag{3}$$

where l_j corresponds to the number of response categories of the j-variable in $\boldsymbol{Y}_{it}$.

Regarding the latent (sub)model, the typical assumption is that the latent Markov chain is of first order and, for the initial and transition probabilities, we can use the same notation previously adopted for the MC model that now is formulated with reference to the latent variables rather than the observable variables. More precisely, we introduce the initial and transition probabilities

$$\begin{aligned} \pi_u &= p(U_{i1} = u), \quad u = 1, \ldots, k, \\ \pi_{uv} &= p(U_{it} = v | U_{i,t-1} = u), \quad t = 2, \ldots, T_i, \; u, v = 1, \ldots, k. \end{aligned}$$

Note that, in the basic version of the HM model, the transition probabilities are assumed to be time homogeneous to reduce the number of free parameters. However, this assumption can be relaxed if it proves to be too restrictive.

When available, individual explanatory variables, collected in the vectors $\boldsymbol{x}_{it}$, may be included in the measurement model, so as to account for the unobserved heterogeneity between units, or in the latent model, so that they affect the initial and transition probabilities of the Markov chain. In the latter case, these probabilities may be defined on the basis of the same multinomial logit parameterizations adopted for the MC model; see in particular Equations (1) and (2). The parameterization of the transition probabilities in Equation (2) is referred to as `paramLatent = "multilogit"` in the corresponding argument of the estimation function `lmest()` and is illustrated with an application to health related data in Bartolucci et al. (2017).

In order to make the model more parsimonious, an alternative differential logit parameterization can be used for the transition probabilities, denoted as `paramLatent = "difflogit"`. Such a parameterization relies on logits based on the difference between two vectors of parameters, for $t = 2, \ldots, T_i$:

$$\log \frac{\pi_{it,uv}}{\pi_{it,uu}} = \gamma_{0uv} + \boldsymbol{x}_{it}'(\boldsymbol{\gamma}_{1v} - \boldsymbol{\gamma}_{1u}), \quad u, v = 1, \ldots, k, \; u \neq v, \tag{4}$$

where $\boldsymbol{\gamma}_{11} = \boldsymbol{0}$ to ensure model identifiability; see Bartolucci et al. (2015) for an example of application of this parameterization.

The **LMest** package also allows including individual covariates in the measurement model through a suitable parameterization of the conditional distribution of the response variables given the latent states; see Bartolucci et al. (2017). This formulation can only be used for univariate data by setting

argument `modManifest = "LM"` in the `lmest()` function. It is also possible to indicate an alternative model specification by setting the option `modManifest = FM`, which relies on the assumption that the latent process has a distribution given by a mixture of AR(1) processes with common variance and specific correlation coefficients. See Bartolucci et al. (2014a) for an illustration of this model; see also Pennoni and Vittadini (2013).

3.2 Maximum likelihood inference

We illustrate maximum likelihood estimation in the general case in which covariates are available and are included in the distribution of the latent process. In this case, for a sample of n independent units, the model log-likelihood has the following expression:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} f(\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{iT_i} | \boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT_i}),$$

where $\boldsymbol{\theta}$ is the vector of all free model parameters and $f(\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{iT_i} | \boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT_i})$ corresponds to the manifest distribution, that is, the probability of the responses provided by subject i given the covariates. This manifest probability has expression

$$f(\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{iT_i} | \boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT_i}) = \sum_{u_1=1}^{k} \cdots \sum_{u_{T_i}=1}^{k} \pi_{i,u_1} \prod_{t=2}^{T_i} \pi_{it,u_{t-1}u_t} \prod_{t=1}^{T_i} f(\boldsymbol{y}_{it} | u_t), \tag{5}$$

where $f(\boldsymbol{y}_{it}|u)$ refers to the conditional distribution of the response variables for unit i at time occasion t given the latent process that, in the case of categorical response variables, is freely parameterized by the conditional response probabilities in Equation (3).

In order to compute $f(\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{iT_i} | \boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT_i})$ while avoiding the sum in Equation (5), which has a computational cost that exponentially increases in T_i, we rely on the Baum and Welch recursion (Baum and Petrie, 1966) that is described in Bartolucci et al. (2013), among others. The above log-likelihood function can be maximized by the EM algorithm based on the complete-data log-likelihood (Baum et al., 1970; Dempster et al., 1977). The EM algorithm is characterized by a series of iterations consisting of two steps, named E- and M-step, which are repeated until convergence. The E-step computes the conditional expected value of the complete-data log-likelihood given the current value of the parameters and the observed data. The M-step consists in maximizing this expected value so as to update the model parameters. The convergence of the EM algorithm is assessed on the basis of a suitable convergence criterion relying on the relative log-likelihood difference between two consecutive iterations. From this iterative algorithm we obtain an estimate of $\boldsymbol{\theta}$, denoted by $\hat{\boldsymbol{\theta}}$.

For HM models, initialization of the estimation algorithm is an important issue as the model log-likelihood is typically multimodal. The **LMest** package allows performing a multi-start initialization strategy, based on both deterministic and random rules, which is aimed at suitably exploring the parameter space so as to increase the chance to reach the global maximum of the model log-likelihood. With some differences, this is done by functions `lmest()` and `lmestSearch()`; the latter will be illustrated in the following.

Once the parameter estimates are computed, standard errors may be obtained in the usual way on the basis of the observed information matrix. This matrix is provided by **LMest** using either the exact computation method proposed in Bartolucci and Farcomeni (2015) or the numerical method of Bartolucci and Farcomeni (2009), depending on the complexity of the model of interest. The package also provides the `bootstrap()` method to obtain standard errors by a parametric bootstrap procedure, that is, by drawing samples from the estimated model and computing the maximum likelihood estimates for every bootstrap sample. The standard errors are obtained by computing, in a suitable way, the standard deviation of the empirical distribution so obtained; see among others, Visser and Speekenbrink (2022). An alternative nonparametric bootstrap procedure can also be used, based on drawing a large number of samples with replacement from the observed data.

Selection of the number of states is performed according to information criteria such as AIC (Akaike, 1973) and BIC (Schwarz, 1978). They are defined as follows:

$$\begin{aligned} AIC &= -2\ell(\hat{\boldsymbol{\theta}}) + 2\,\#par, \\ BIC &= -2\ell(\hat{\boldsymbol{\theta}}) + \log(n)\,\#par, \end{aligned}$$

where $\ell(\hat{\boldsymbol{\theta}})$ denotes the maximized log-likelihood of the model of interest, n is the sample size, and $\#par$ denotes the number of free parameters to be estimated. Other criteria, such as those based on entropy may be used; see, among others, Bacci et al. (2014).

Once the number of states is selected, dynamic clustering is performed by assigning every unit to

a latent state at each time occasion by means of the estimated posterior probabilities of the U_{it}. These probabilities are directly provided by the EM algorithm and are defined as

$$\hat{a}_{it,u} = p(U_{it} = u|\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{iT_i}, \boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT_i}),\ t = 1, \ldots, T_i,\ u = 1, \ldots, k, \tag{6}$$

$$\hat{b}_{it,uv} = p(U_{i,t-1} = u, U_{it} = v|\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{iT_i}, \boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT_i}),\ t = 2, \ldots, T_i,\ u, v = 1, \ldots, k.$$

Prediction of the latent states of every unit i at each time occasion t is known as *local decoding* and it is obtained as the value of u that maximizes the posterior probabilities in Equation (6). As an alternative to the local decoding, the *global decoding* may be performed, which is based on the Viterbi algorithm (Viterbi, 1967; Juang and Rabiner, 1991) to obtain the prediction of the latent trajectories of a unit across time, that is, the *a posteriori* most likely sequence of hidden states. The package provides the `lmestDecoding()` method to perform both local and global decoding based on the different model specifications, as illustrated in the following sections. Additional details on the model formulations, backward and forward recursions, and estimation can be found in Bartolucci et al. (2013).

3.3 Application to longitudinal data with covariates

As an illustration of the **LMest** package for estimation of HM models with multivariate responses and covariates, we consider simulated data provided in the package, named `data_market_sim`. These data refer to a hypothetical marketing context where a sample of $n = 200$ customers of four different brands is observed along with the price of each transaction, defined for $T = 5$ occasions, in Euros per purchase within the following ranges: [0.1, 10], (10, 30], (30, 60], (30, 100], (100, 500]. For a similar application see, among others, Paas et al. (2007); Bassi et al. (2021). Accordingly, we consider $r = 2$ response variables (items), the first variable with $l_1 = 4$ categories, one for each brand, and the second variable with $l_2 = 5$ categories, one for each range of price. The initial part of the dataset, in long format, is displayed below:

```
data("data_market_sim")
head(data_market_sim)
```

```
#>   id time brand price age income
#> 1  1    1     2     2  34     25
#> 2  1    2     0     2  35     25
#> 3  1    3     1     0  36     25
#> 4  1    4     2     2  37     25
#> 5  1    5     3     1  38     25
#> 6  2    1     1     3  35     27
```

The research questions in this context concern the evolution of customer's purchase behavior over time by exploring market segments, while also considering the role of socio-demographic characteristics defined by the available individual covariates. In particular, we include age, as time-varying continuous covariate, and income of the customer at the time of the first purchase, as time-fixed continuous covariate.

As already mentioned, the package provides function `lmestSearch()`, which searches for the global maximum of the log-likelihood of different models and selects the optimal number of hidden states. This function combines deterministic and random initializations with a rather wide tolerance level. Moreover, starting from the best solution obtained from these preliminary runs, a final run is performed, with a smaller tolerance level, in order to increase the chance of reaching the global maximum. The argument `nrep` can be provided to set the number of random initializations for each number of hidden states. The range of states to be considered may be specified in argument `k` as a vector of integers. Generally, model selection is performed by estimating the multivariate HM model without covariates, which is specified within the model formula, indicated in the argument `responsesFormula`, with `~ NULL` stating that there are no predictors for the two response variables as follows:

```
hmm <- lmestSearch(responsesFormula = brand + price ~ NULL,
                   latentFormula =  ~ NULL,
                   version = "categorical",
                   index = c("id", "time"),
                   data = data_market_sim,
                   k = 1:4, fort = TRUE,
                   seed = 12345)
```

By adding the optional `fort = TRUE` argument, a faster estimation procedure is achieved using Fortran routines. The `summary()` method returns the estimation results, displaying the AIC and BIC values for the sequence of estimated hidden states:

```
summary(hmm)
```

```
#> Call:
#> lmestSearch(responsesFormula = brand + price ~ NULL, latentFormula = ~NULL,
#>     data = data_market_sim, index = c("id", "time"), k = 1:4,
#>     version = "categorical", seed = 12345, fort = TRUE)
#>
#>  states        lk np      AIC      BIC
#>       1 -2811.001  7 5636.003 5659.091
#>       2 -2520.131 17 5074.263 5130.334
#>       3 -2445.538 29 4949.075 5044.727
#>       4 -2434.262 43 4954.524 5096.352
```

In this case, the BIC, used by default for model selection, supports a model with three hidden states. Once the model is selected, we estimate the parameters of the latent model with covariates by fixing the parameter values of the conditional response probabilities obtained under the model chosen above. In this way, the estimated subpopulations are held fixed. These conditional response probabilities are displayed below:

```
Psi <- hmm$out.single[[3]]$Psi
print(round(Psi, 2))
```

```
#> , , item = 1
#>
#>         state
#> category    1    2    3
#>        0 0.69 0.08 0.10
#>        1 0.10 0.45 0.12
#>        2 0.17 0.40 0.08
#>        3 0.05 0.07 0.70
#>        4   NA   NA   NA
#>
#> , , item = 2
#>
#>         state
#> category    1    2    3
#>        0 0.30 0.04 0.00
#>        1 0.43 0.19 0.05
#>        2 0.10 0.64 0.05
#>        3 0.12 0.11 0.30
#>        4 0.04 0.02 0.60
```

From these results, we notice that the three states may represent distinct customer segments that can be labeled as follows: low-cost market segment (1st), ordinary segment (2nd), and luxury segment (3rd). These segments are described in more detail in the following.

Function `lmest()` estimates the HM model with covariates that can affect the latent structure in various ways. In the following, we assume that age and income may influence only the transition probabilities, and do not affect the composition of the market segments at the beginning of the survey. This can be set through the `latentFormula` argument, which includes `age + income` in the parameterization of the transition probabilities. For the initial probabilities, we ignore the effects of covariates and estimate only the intercept of the multinomial logit by including `NULL` in the formula before the symbol "`|`". Note that this formulation is the same as the one expressed in Equation (1) but without the vector of covariates, whereas for the transition probabilities we consider the parametrization defined in Equation (4) obtained by including the argument `paramLatent = "difflogit"`. Moreover, as an input to the `lmest()` function, we also use the optional arguments `parInit`, which is a list of initial model parameters that also includes argument `fixPsi = TRUE` to avoid estimation of the conditional response probabilities. In this way, the EM algorithm is performed to estimate the parameters of the structural model while these probabilities are kept fixed at the value displayed above. The same arguments can be used to require the estimation of an MC model for unbalanced data as a constrained version of an HM model with a number of states equal to the number of response categories.

The HM model with three hidden states and fixed conditional response probabilities, chosen as described above, is estimated as follows:

```
mod2 <- lmest(responsesFormula = brand + price ~ NULL,
              latentFormula =  ~ NULL | age + income,
              k = 3, data = data_market_sim,
              index = c("id", "time"),
              parInit = list(Psi = Psi, fixPsi = TRUE),
              paramLatent = "difflogit",
              seed = 12345)
```

The `print()` method provides an overview of the estimation results including the maximum log-likelihood, number of free parameters, and AIC and BIC values that can be used for model selection.

```
print(mod2)
```

```
#>
#> Basic Latent Markov model with covariates in the latent model
#> Call:
#> lmest(responsesFormula = brand + price ~ NULL, latentFormula = ~NULL |
#>     age + income, data = data_market_sim, index = c("id", "time"),
#>     k = 3, paramLatent = "difflogit", parInit = list(Psi = Psi,
#>         fixPsi = TRUE), seed = 12345)
#>
#> Available objects:
#>  [1] "lk"          "Be"          "Ga"          "Psi"         "Piv"
#>  [6] "PI"          "np"          "k"           "aic"         "bic"
#> [11] "lkv"         "n"           "TT"          "paramLatent" "ns"
#> [16] "yv"          "Lk"          "Bic"         "Aic"         "call"
#> [21] "data"
#>
#> Convergence info:
#>      LogLik np k      AIC      BIC   n TT
#>   -2444.437 12 3 4912.874 4952.454 200  5
```

The estimated HM model has a maximum log-likelihood of -2,444.437 with 12 parameters so that AIC and BIC indexes are equal to 4,912.874 and 4,952.454, respectively.

The `plot()` method associated with `lmest()` provides a variety of displays. For example, a plot of the estimated conditional response probabilities can be obtained as:

```
par(mar = c(5,4,4,2) + 0.1)
plot(mod2, what = "CondProb")
```

as shown in Figure 1. According to the simulated context, from this figure we observe that the 1st state (low-cost segment), which includes the highest frequency of customers at the first time occasion (39%), is related to those who primarily purchase products of the first brand (category 0 of item 1) with low prices (categories 0 and 1 of item 2). Customers in the 2nd state (ordinary segment), corresponding to around 30% of all customers, tend to buy products of the second and third brands with medium prices. On the other hand, customers in the 3rd state (luxury segment) tend to purchase products of brand four (category 3 of item 1) with relatively high prices (categories 3 and 4 of item 2). This state includes around 31% of customers at the beginning of the period of observation.

The averaged estimated transition probabilities may be obtained by the following command:

```
print(round(apply(mod2$PI[,,2:5], c(1,2), mean), 3))
```

```
#>      state
#> state     1     2     3
#>     1 0.092 0.486 0.422
#>     2 0.028 0.878 0.093
#>     3 0.000 0.106 0.894
```

A plot showing the corresponding path diagram of the estimated transition matrix can be obtained as

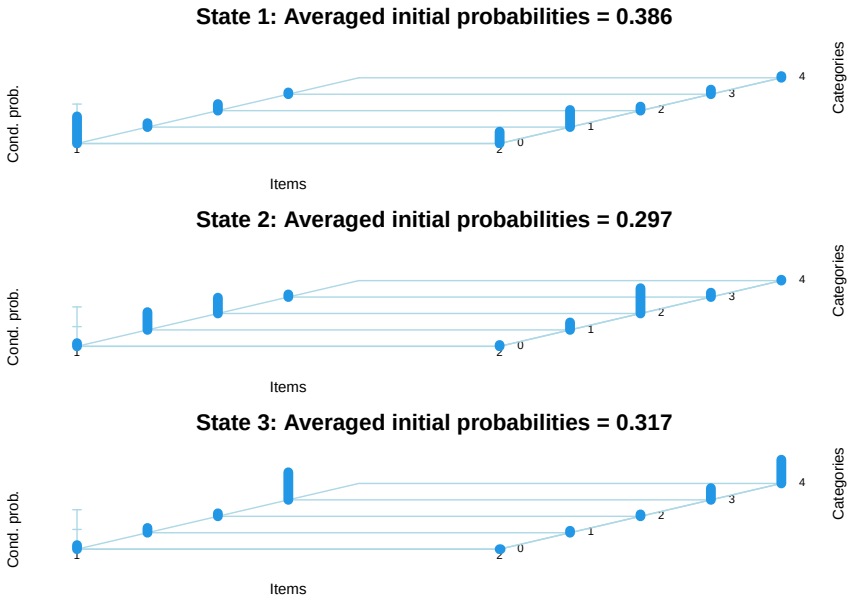

Figure 1: Plot of the estimated conditional response probabilities: on the left side there is item 1 (brand), and on the right side item 2 (price), respectively. The top, middle, and bottom panels correspond to the estimates for the 1st state (low-cost segment), 2nd state (ordinary segment), and 3rd state (luxury segment), respectively.

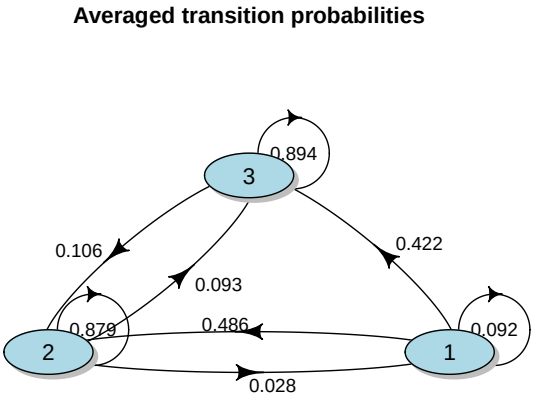

Figure 2: Path diagram of averaged estimated transition probabilities: nodes represent hidden states, and arrows are present when the estimated transition probability between two states is not null. Self-arrows indicate persistence in the same state if estimated. The reported numbers refer to the estimated values of the transition probability matrix.

```
par(mar = c(2,1,5,1))
plot(mod2, what = "transitions")
```

as illustrated in Figure 2. From the results shown in this figure, we observe that customers have a fairly high probability of persistence in the 2nd and 3rd state from one time period to the next, as the estimated diagonal elements of the transition probability matrix are fairly high (0.88 and 0.89, respectively). This suggests that these two market segments are quite stable across customers over years. Conversely, the highest estimated probabilities outside of the main diagonal refer to the transition from the 1st state (low-cost segment) to the 2nd state (ordinary) and from the 1st to the 3rd (luxury). This shows that customers of the 1st segment have a greater tendency to change their behavior over time.

The `summary()` method returns the main estimation results:

```
summary(mod2)

#> Call:
#> lmest(responsesFormula = brand + price ~ NULL, latentFormula = ~NULL |
#>     age + income, data = data_market_sim, index = c("id", "time"),
```

```
#>     k = 3, paramLatent = "difflogit", parInit = list(Psi = Psi,
#>         fixPsi = TRUE), seed = 12345)
#>
#> Coefficients:
#>
#>  Be - Parameters affecting the logit for the initial probabilities:
#>              logit
#>                    2      3
#>   (Intercept) -0.2604 -0.196
#>
#>  Ga0 - Intercept affecting the logit for the transition probabilities:
#>            logit
#> (Intercept)       2       3
#>           1 -3.2126 -3.9261
#>           2  1.4353 -2.8156
#>           3 -5.4099 -1.5630
#>
#>  Ga1 - Regression parameters affecting the logit for the transition probabilities:
#>         logit
#>               2      3
#>   age    0.0836 0.0897
#>   income 0.0563 0.0674
#>
#>  Psi - Conditional response probabilities:
#> , , item = 1
#>
#>         state
#> category      1      2      3
#>        0 0.6875 0.0816 0.1031
#>        1 0.1000 0.4512 0.1216
#>        2 0.1674 0.4010 0.0759
#>        3 0.0450 0.0663 0.6993
#>        4     NA     NA     NA
#>
#> , , item = 2
#>
#>         state
#> category      1      2      3
#>        0 0.3016 0.0353 0.0000
#>        1 0.4344 0.1895 0.0505
#>        2 0.1033 0.6407 0.0489
#>        3 0.1172 0.1113 0.2981
#>        4 0.0436 0.0232 0.6025
```

The output `Ga` contains the estimated parameters affecting the distribution of the transition probabilities based on the difflogit parameterization. More in detail, `Ga0` refers to the intercepts, while `Ga1` refers to the regression coefficients. Each column of `Ga1` can be interpreted as a general measure of attraction of the corresponding state. From these results, we observe that the estimated effects of the age and income of the customer are relatively small for each logit. Both covariates have a positive impact, indicating that as age or income increases, the probability of moving to more valuable segments also increases, while holding other parameters constant. Thus, older customers or those with higher income are more likely to move to higher-value segments.

4 Hidden markov models for continuous data assuming a conditional gaussian distribution

In this section, we illustrate the main notation used for HM models for continuous outcomes and discuss how to handle missing responses under the MAR assumption (Little and Rubin, 2020). We also illustrate the inclusion of covariates in both the latent and measurement models. Finally, we introduce three examples of the application of these models using the appropriate functions of the **LMest** package.

4.1 Model assumptions and inference

When dealing with continuous outcomes, let $\boldsymbol{Y}_{it} = (Y_{i1t}, \ldots, Y_{irt})'$ denote the vector of r continuous response variables measured at time t for subject i. As usual, under the local independence assumption, the response vectors $\boldsymbol{Y}_{i1}, \ldots, \boldsymbol{Y}_{iT_i}$ are assumed to be conditionally independent given the latent process $\boldsymbol{U}_i$. Moreover, for the measurement model, we assume a conditional Gaussian distribution, that is,

$$\boldsymbol{Y}_{it}|U_{it} = u \sim N(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u), \quad u = 1, \ldots, k. \tag{7}$$

The parameters of the measurement (sub)model are the conditional means $\boldsymbol{\mu}_u$, $u = 1, \ldots, k$, which are state specific, and the variance-covariance matrices $\boldsymbol{\Sigma}_u$, which may be assumed to be constant across states under the assumption of homoschedasticity, that is, $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$. This assumption, which reduces the number of estimated parameters and may be reasonable in many applied contexts, is adopted in the **LMest** package.

The parameters of the latent (sub)model are again the initial and transition probabilities of the Markov chain, as specified in the previous sections. As usual, when individual covariates collected in vectors $\mathbf{x}_{it}$ are available, they may be included in the distribution of the latent variables, so as to affect the initial and transition probabilities. It is also possible to assume that individual covariates affect the distribution of the response variables given the latent process, thereby accounting for unobserved heterogeneity. The latter formulation is based on the assumption that

$$E(Y_{it}|U_{it} = u, \boldsymbol{x}_{it}) = \alpha_u + \boldsymbol{x}_{it}'\boldsymbol{\beta}, \quad u = 1, \ldots, k,$$

which naturally extends to the multivariate case.

The manifest distribution may be expressed with a formula that closely recalls the one used in Equation (5) for HM models for categorical responses, but with $f(\boldsymbol{y}_{it}|u)$ that here denotes the density of the multivariate Gaussian distribution based on assumption in Equation (7), possibly dependent on the covariates. Maximum likelihood estimation is carried out as illustrated in Section 3.2 through the EM algorithm and its extended version when individual covariates are included in the model. The **LMest** package can also handle missing values in the response variables. When the outcomes are continuous, we consider two different types of missing pattern under the MAR assumption: partially missing outcomes at a given time occasion and completely missing outcomes, that is, when individuals do not respond at one or more time occasions (intermittent pattern). Following Pandolfi et al. (2023), in the presence of missing responses, it is convenient to partition each response vector $\boldsymbol{Y}_{it}$ as $(\boldsymbol{Y}_{it}^o, \boldsymbol{Y}_{it}^m)'$, where $\boldsymbol{Y}_{it}^o$ is the (sub)-vector of observed variables, and $\boldsymbol{Y}_{it}^m$ refers to the missing data. The conditional mean vectors and variance-covariance matrix may be decomposed into observed and missing components as follows:

$$\boldsymbol{\mu}_u = \begin{pmatrix} \boldsymbol{\mu}_u^o \\ \boldsymbol{\mu}_u^m \end{pmatrix}, \quad \boldsymbol{\Sigma}_u = \begin{pmatrix} \boldsymbol{\Sigma}_u^{oo} & \boldsymbol{\Sigma}_u^{om} \\ \boldsymbol{\Sigma}_u^{mo} & \boldsymbol{\Sigma}_u^{mm} \end{pmatrix},$$

where, for instance, $\boldsymbol{\Sigma}_u^{om}$ is the block of $\boldsymbol{\Sigma}_u$ collecting covariances between each observed and missing response. In this way, for the observed responses, we have that:

$$\boldsymbol{Y}_{it}^o|U_{it} = u \sim N(\boldsymbol{\mu}_u^o, \boldsymbol{\Sigma}_u^{oo}), \quad u = 1, \ldots, k.$$

The manifest distribution of the responses is now expressed with reference to the observed data. Model inference is carried out using an extended version of the EM algorithm based on suitable recursions, as illustrated in Pandolfi et al. (2023). In particular, at the E-step the algorithm also computes additional expected values arising from the MAR assumption for the missing observations. Under this model formulation, it is also possible to perform a type of multiple imputation, which allows us to predict the missing responses either conditionally or unconditionally on the predicted states. Within the **LMest** package, when the missing data are dealt with under the MAR assumption, the unconditional prediction of the missing responses is performed by computing

$$\tilde{\boldsymbol{y}}_{it} = \sum_{u=1}^{k} \hat{a}_{it,u} E(\boldsymbol{Y}_{it}|\boldsymbol{y}_{it}^o, u),$$

where $\hat{a}_{it,u} = p(U_{it}|\boldsymbol{y}_{i1}^o, \ldots, \boldsymbol{y}_{iT_i}^o, \boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{iT_i})$ and $E(\boldsymbol{Y}_{it}|\boldsymbol{y}_{it}^o, u)$ are posterior expected values computed at the E-step.

Finally, both local and global decoding may be performed as usual. In this context, it is important to note that the prediction of the latent states is also carried out for units with missing responses.

4.2 Application to time-series

This section illustrates an application of the HM model to multivariate time-series data; for more details, see Pennoni et al. (2022). Specifically, we use real data from Yahoo Finance, covering S&P 500 Index (SP500TR) and Intel stock prices (INTC). These data are sourced from the R package **quantmod** (Ryan and Ulrich, 2022). The dataset includes the closing prices for each trading day from March to August 2022. For the following application, we then compute the percentage returns based on these closing prices.

The analysis of financial time-series data is typically focused on identifying market phases associated with the volatility of returns. In this context, latent states are referred to as regimes, and it is valuable to detect any abrupt and persistent changes in these regimes.

```
require(quantmod)
SP500_22 <- getSymbols("^SP500TR",
                       env = NULL,
                       from = "2022-03-01",
                       to = "2022-08-31",
                       periodicity = "daily")
dim(SP500_22)

#> [1] 127   6

INCT_22 <- getSymbols("INTC",
                      env = NULL,
                      from = "2022-03-01",
                      to = "2022-08-31",
                      periodicity = "daily")

sp_sreturns <- dailyReturn(SP500_22)*100
intc_sreturns <- dailyReturn(INCT_22)*100

data_fin <- cbind(1, 1:length(sp_sreturns), sp_sreturns, intc_sreturns)

names(data_fin) <- c("id", "time",  "SP", "INCT")
head(round(data_fin, 3))

#>            id time     SP   INCT
#> 2022-03-01  1    1 -1.304 -1.515
#> 2022-03-02  1    2  1.868  4.378
#> 2022-03-03  1    3 -0.513 -1.923
#> 2022-03-04  1    4 -0.786  0.292
#> 2022-03-07  1    5 -2.951 -0.811
#> 2022-03-08  1    6 -0.721 -0.378

data_fin[which.min(data_fin$SP),]

#>            id time        SP      INCT
#> 2022-05-18  1   56 -4.016448 -4.617124

data_fin[which.min(data_fin$INCT),]

#>            id time       SP      INCT
#> 2022-07-29  1  105 1.431582 -8.562069
```

The two series are displayed in Figure 3 and 4, while descriptive statistics are reported in the following:

```
#>       Index                  SP                INCT
#>  Min.   :2022-03-01   Min.   :-4.01645   Min.   :-8.5621
#>  1st Qu.:2022-04-13   1st Qu.:-0.89097   1st Qu.:-1.8511
#>  Median :2022-05-31   Median :-0.06888   Median :-0.1265
#>  Mean   :2022-05-30   Mean   :-0.05265   Mean   :-0.2769
#>  3rd Qu.:2022-07-16   3rd Qu.: 1.09352   3rd Qu.: 1.1456
#>  Max.   :2022-08-30   Max.   : 3.05815   Max.   : 6.9401
```

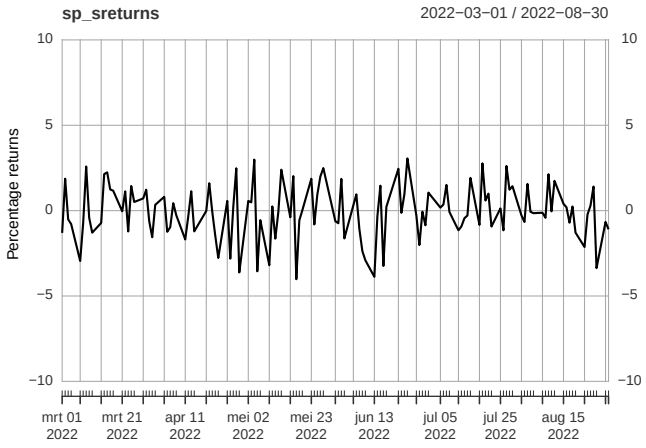

Figure 3: Observed percentage returns of Standard and Poor's 500 Index from March to August 2022 (127 days).

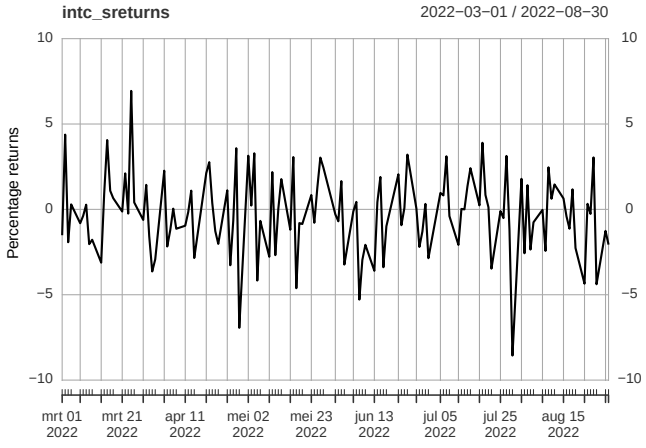

Figure 4: Observed percentage returns of Intel stock from March to August 2022 (127 days).

We can easily estimate the HM model for continuous data with three hidden states, assuming they may represent bear, bull, and intermediate market regimes. The function of the package that can be used to estimate this model is `lmestCont()`; the basic version of the model, without covariates, can be specified by setting the argument `responsesFormula = SP + INCT ~ NULL`.

As for the other estimation functions, the argument `index` is required to specify the name of the unit identifier and the indicator of the time occasions. The model can be fitted with the constraint of time-homogeneity, by setting the argument `modBasic = 1`, which ensures that the transition probabilities do not depend on t. Argument `tol = 10^-6` is used to set the tolerance level for the convergence of the algorithm; by default, this is set to `tol = 10^-10`. We can also specify `maxit`, which is an integer that sets the maximum number of EM iterations; by default, this is set to 5,000.

```
mod3 <- lmestCont(responsesFormula = SP + INCT ~ NULL,
                  data = data_fin,
                  index = c("id", "time"),
                  k = 3, modBasic = 1, tol = 10^-6)
```

A summary of the estimation results is obtained with the `summary()` method:

```
summary(mod3)
```

```
#> Call:
#> lmestCont(responsesFormula = SP + INCT ~ NULL, data = data_fin,
#>     index = c("id", "time"), k = 3, modBasic = 1, tol = 10^-6)
#>
#> Coefficients:
#>
#> Initial probabilities:
#>      est_piv
#> [1,]       1
#> [2,]       0
#> [3,]       0
#>
#> Transition probabilities:
#>      state
#> state      1      2      3
#>     1 0.1265 0.2715 0.6020
#>     2 0.0389 0.9610 0.0001
#>     3 0.5113 0.0001 0.4887
#>
#>  Mu - Conditional response means:
#>       state
#> item         1       2      3
#>   SP   -2.5609  0.2353 0.6531
#>   INCT -2.9240 -0.0996 1.1140
#>
#>  Si - Variance-covariance matrix:
#>        [,1]   [,2]
#> [1,] 1.5253 1.7211
#> [2,] 1.7211 4.3718
```

This summary output shows the estimated initial and transition probabilities of the three latent states, the estimated cluster means, and the variance-covariance matrix. The three states represent different market regimes that can be interpreted according to the estimated means as follows: the 1st state identifies negative returns, the 2nd state corresponds to positive returns for S&P 500 (SP) and negative for Intel stock (INCT), and the 3rd state identifies positive returns for both. The main transitions are from the 1st to the 2nd state (0.272), from the 1st to the 3rd (0.602) state, and from the 3rd to the 1st state (0.511).

The contour plot of the estimated overall density, with weights given by the estimated marginal probabilities of the latent states, is shown in Figure 5. It is obtained by the `plot()` method using the argument `what = "density"` as follows:

```
plot(mod3, what = "density")
```

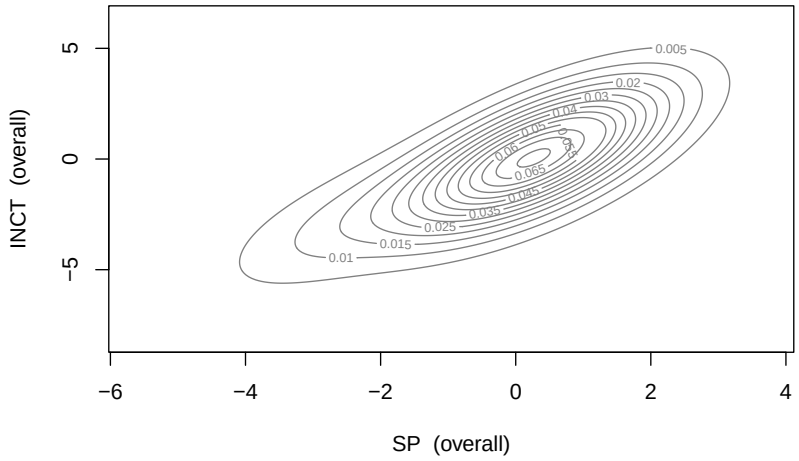

Figure 5: Estimated density surface of the bivariate distribution of Standard and Poor's 500 Index and Intel stock prices observed from March to August 2022 (127 days)

The density regions of each estimated component (regime), represented as a contour plot, can be visualized by setting the argument `components = c(1, 2, 3)`. The resulting plot is shown in Figure 6.

```
par(mar = c(3,4,2,2), oma = c(0,1,0,1))
plot(mod3, what = "density", components = c(1,2,3))
```

In the context of financial time-series, a relevant issue is path prediction, that is, finding the most likely sequence of latent states for a given unit on the basis of the observed values. As already mentioned, both local and global decoding are implemented in the `lmestDecoding()` method, which allows us to predict the sequence of latent states for a sample unit on the basis of the output of the different estimation functions, as follows:

```
deco3 <- lmestDecoding(mod3)
```

Object `deco3` contains the local (object `Ul`) and global (object `Ug`) decoding (obtained through the Viterbi algorithm). For example, for the local decoding, we have:

```
deco3$Ul
```

```
#>   [1] 1 3 3 3 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
#>  [38] 1 3 1 3 3 1 3 3 3 1 3 1 3 1 3 3 3 3 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 3
#>  [75] 3 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
#> [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
```

These sequences can be depicted as follows:

```
data_fin$Ul <- deco3$Ul
plot(data_fin$Ul, ylab="State")
```

From Figure 7 we can observe distinct periods characterized by different market regimes, as indicated by the decoded state sequence. Initially, a neutral market regime is observed, followed by alternating bullish and bearish market regimes, and finally, a neutral market regime predominates during the last period.

4.3 Application to longitudinal data with missing values and covariates in the latent (sub)models

Data provided in the package, named `data_heart_sim`, are simulated from a hypothetical observational retrospective study designed to assess the progression of health states of individuals after treatment. The dataset includes observations on systolic and diastolic blood pressure (variables `sap` and `dap` in mmHg) and heart rate (variable `hr` in bpm), as well as covariates such as fluid administration (variable `fluid` in ml/kg/h), gender (variable `gender`: 1 for male, 2 for female), and age (variable `age` in years). Note that `fluid` is a time-varying covariate. The initial part of the dataset is as follow:

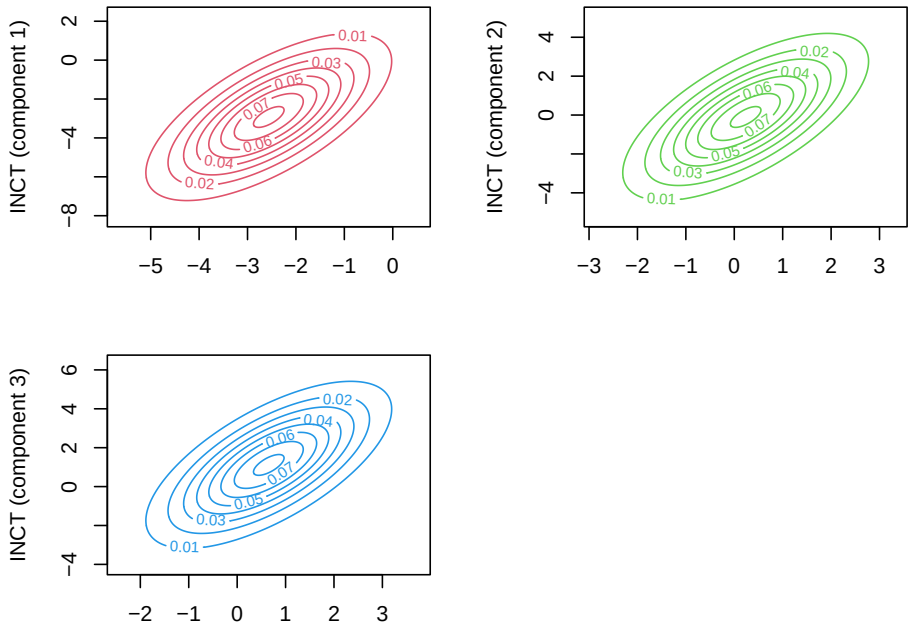

Figure 6: Estimated density surfaces for each hidden state (component) represented as contour levels of Standard and Poor's 500 Index and Intel stock prices observed from March to August 2022 (127 days).

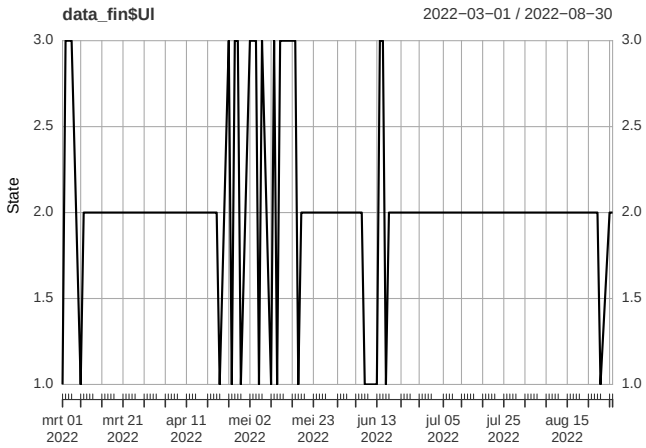

Figure 7: Predicted sequence of hidden states from March to August 2022 (127 days) under the HM model with $k = 3$ estimated for Standard and Poor's 500 Index and Intel stock prices.

```
data("data_heart_sim")
head(data_heart_sim)

#>   id time sap dap  hr fluid gender age
#> 1  1    1 117  67  75  2800      1  56
#> 2  1    2 111  69  93  1750      1  56
#> 3  1    3 102  60 108  2320      1  56
#> 4  1    4 123  78 105  2600      1  56
#> 5  1    5 102  57  99  1700      1  56
#> 6  1    6  86  60  96  2210      1  56
```

We assume that these measurements are recorded daily after a particular surgery for up to six days, so that $T = 6$, and some patients have missing records for one or more outcomes. The number of patients is $n = 125$. Missing responses are denoted with `NA` in the data provided to the estimation function. Note that the package does not support missing values in the covariates, which should be imputed separately. The data are simulated to include a proportion of missing values of around 15%.

We estimate an HM model with three hidden states, including covariates that affect both the initial and transition probabilities, and where missing data are handled under the MAR assumption. This model is specified using the full set of responses through the function `lmestCont()` as follows:

```
mod4 <- lmestCont(responsesFormula = sap + dap + hr  ~ NULL,
                  latentFormula = ~ fluid + as.factor(gender) + age,
                  data = data_heart_sim,
                  index = c("id", "time"),
                  k = 3, miss.imp = FALSE)
```

Note that it is possible to specify how to deal with missing responses by setting the logical argument `miss.imp`. The MAR assumption is adopted if this argument is set to `FALSE` (the default value). Otherwise, missing outcomes are imputed using the `imp.mix()` function from the package **mix** (Schafer, 2024) before starting the estimation.

The following command provides a summary of the estimation results:

```
summary(mod4)

#> Call:
#> lmestCont(responsesFormula = sap + dap + hr ~ NULL, latentFormula = ~fluid +
#>     as.factor(gender) + age, data = data_heart_sim, index = c("id",
#>     "time"), k = 3, miss.imp = FALSE)
#>
#> Coefficients:
#>
#>  Be - Parameters affecting the logit for the initial probabilities:
#>                     logit
#>                           2       3
#>   (Intercept)        -0.0348  0.0258
#>   fluid               0.0003  0.0001
#>   as.factor(gender)2 -0.4704  0.3428
#>   age                -0.0290 -0.0096
#>
#>  Ga - Parameters affecting the logit for the transition probabilities:
#> , , logit = 1
#>
#>                     logit
#>                           2       3
#>   (Intercept)        -2.1318 -2.3075
#>   fluid              -0.0007  0.0001
#>   as.factor(gender)2  2.1343  0.0536
#>   age                -0.0047  0.0076
#>
#> , , logit = 2
#>
#>                     logit
#>                           2       3
```

```
#>   (Intercept)        -0.9958 -1.9856
#>   fluid              -0.0025 -0.0001
#>   as.factor(gender)2  4.3148  2.4297
#>   age                -0.0493 -0.0641
#>
#> , , logit = 3
#>
#>                     logit
#>                         2       3
#>   (Intercept)        -2.2754 -2.4416
#>   fluid               0.0008  0.0007
#>   as.factor(gender)2  0.3736 -1.9600
#>   age                -0.0700 -0.0342
#>
#>
#>  Mu - Conditional response means:
#>      state
#>              1        2        3
#>   sap 101.0068 104.9304 127.5081
#>   dap  61.9488  66.9759  73.6769
#>   hr   74.2144 102.1859  83.0373
#>
#>  Si - Variance-covariance matrix:
#>          [,1]     [,2]     [,3]
#> [1,] 197.0381  55.0938  12.7835
#> [2,]  55.0938 107.8801  21.0515
#> [3,]  12.7835  21.0515 123.8274
```

Considering the estimated conditional means under the HM model with $k = 3$ latent states, we observe that these states are ordered increasingly according to the values of the estimated means for both systolic (`sap`) and diastolic (`dap`) blood pressure. The estimated regression parameters related to the initial probabilities are stored in object `Be`. The log-odds ratio for gender is positive for the 3rd state and negative for the 2nd, indicating that females are more likely to be in the 3rd and 1st states one day after surgery, holding other covariates constant.

The output `Ga` refers to the estimated regression parameters on the transition probabilities. For example, the values in the 1st column of `Ga[„3]` measure the influence of each covariate on the transition from the 3rd to the 1st state. For females, the probability of this transition is higher than for males and age has a negative effect on this transition.

Parametric bootstrap standard errors for the parameter estimates can be obtained using the `bootstrap()` method by specifying, as input argument, the object returned by the call of `lmestCont()`. The number of bootstrap samples can be specified through argument `B`. The call is as follows:

```
boot <- bootstrap(mod4, B = 20)
```

Results of function `bootstrap()` include the average of bootstrap estimates and the standard errors for the model parameters. For example, the estimated standard errors for the regression parameters affecting the logits on initial and transition probabilities can be obtained as

```
round(boot$seBe, 3)
```

```
#>                     logit
#>                         2     3
#>   (Intercept)        0.014 0.031
#>   fluid              0.000 0.000
#>   as.factor(gender)2 0.206 0.402
#>   age                0.012 0.011
```

```
round(boot$seGa, 3)
```

```
#> , , logit = 1
#>
#>                     logit
#>                         2     3
```

```
#>   (Intercept)        0.011 0.036
#>   fluid              0.000 0.001
#>   as.factor(gender)2 0.099 0.520
#>   age                0.015 0.026
#>
#> , , logit = 2
#>
#>                     logit
#>                         2     3
#>   (Intercept)        4.240 0.705
#>   fluid              0.006 0.004
#>   as.factor(gender)2 4.477 2.196
#>   age                0.159 0.301
#>
#> , , logit = 3
#>
#>                     logit
#>                         2     3
#>   (Intercept)        0.031 0.042
#>   fluid              0.004 0.001
#>   as.factor(gender)2 0.245 0.629
#>   age                0.318 0.069
```

4.4 Application to longitudinal data with imputed missing values and covariates in the measurement (sub)model

We examine data collected by the Minnesota Department of Education for all Minnesota schools during the period 2008-2010, as illustrated in Roback and Legler (2021). The dataset is available in the GitHub repository at the link provided in the following chunk.

```
urlfile <- "https://raw.githubusercontent.com/proback/BeyondMLR/master/data/chart_wide_condense.csv"

data <- read.csv(url(urlfile))
n <- nrow(data); TT <- 3
data_school <- data.frame(id = rep(1:n, each = TT), time = rep(1:TT, n),
                          chart = rep(data$charter, each = TT),
                          sped = rep(data$schPctsped, each = TT),
                          math = c(t(data[,c("MathAvgScore.0",
                                             "MathAvgScore.1", "MathAvgScore.2")])))
```

The final part of the dataset in long format is displayed in the following:

```
round(tail(data_school), 3)

#>       id time chart  sped  math
#> 1849 617    1     1 0.105    NA
#> 1850 617    2     1 0.105    NA
#> 1851 617    3     1 0.105 651.4
#> 1852 618    1     1 0.455    NA
#> 1853 618    2     1 0.455    NA
#> 1854 618    3     1 0.455 631.2
```

The data refer to 618 schools, and the aim is to compare student performance in charter schools versus public schools by analyzing Minnesota Comprehensive Assessment average math scores from 6th to 8th graders in each school. The response variable (`math`) is recorded at three time occasions corresponding to years 2008, 2009, and 2010. Type of school (`chart`: coded as 1 for a charter school, and 0 for a public school), and the proportion of special education students in a school (`sped`), based on 2010 figures, are time-fixed covariates. In this example, we investigate the effects of the two covariates on the test performance, specifically examining whether there are differences between charter and public schools.

We observe that the school with id 618 has missing records for the math scores at both the first and second occasions. Additionally, there are 121 missing math scores out of a total of 1,854 records.

```
table(complete.cases(data_school$math))

#>
#> FALSE  TRUE
#>   121  1733
```

We estimate an HM model with two latent states, handling missing values through imputation, and accounting for the effect of covariates on the measurement model. This is done using the `lmestCont()` function with the following options:

```
mod5 <- lmestCont(responsesFormula = math ~ chart + sped,
                  data = data_school,
                  index = c("id", "time"),
                  k = 2, miss.imp = TRUE)
```

The summary output presents the estimated model parameters, including the following details:

```
summary(mod5)

#> Call:
#> lmestCont(responsesFormula = math ~ chart + sped, data = data_school,
#>     index = c("id", "time"), k = 2, miss.imp = TRUE)
#>
#> Coefficients:
#>
#> Initial probabilities:
#>      est_piv
#> [1,]  0.1652
#> [2,]  0.8348
#>
#> Transition probabilities:
#> , , time = 2
#>
#>      state
#> state      1      2
#>     1 0.9233 0.0767
#>     2 0.0130 0.9870
#>
#> , , time = 3
#>
#>      state
#> state      1      2
#>     1 0.7419 0.2581
#>     2 0.0085 0.9915
#>
#>
#>  Al - Intercepts:
#>
#> state     [,1]
#>     1 644.8250
#>     2 656.8341
#>
#>  Be - Regression parameters:
#>          [,1]
#> [1,]  -2.8784
#> [2,] -12.8279
#>
#>  Si - Variance-covariance matrix:
#>         [,1]
#> [1,] 24.0767
```

The estimated intercepts, collected in object `Al`, indicate that the 1st state corresponds to lower performing schools. Additionally, the analysis reveals a negative effect of attending a non-charter

public school on the average math score, while controlling for the percentage of special education students. Covariate sped has also a negative effect on the math score. Specifically, an increase of 1% in the proportion of special education students at a school is associated with a decrease of 12.83 points in the estimated mean math scores.

From the estimated initial probabilities, we infer that, in 2008, around 17% of schools belong to the 1st state. The two estimated transition matrices indicate a significant increase in the average math score from 2009 to 2010, as evidenced by the transition probability from the 1st to the 2nd state, which is 0.26.

The output collected in `mod5` also includes an object named `Yimp`, which is an array of dimension $n \times T \times k$ containing the imputed data. For instance, the imputed values for unit 618 at the first and second time occasion are 662.966 and 661.465, respectively.

```
round(mod5$Yimp[618,,], 3)

#> [1] 662.966 661.465 631.200
```

5 Discussion

The **LMest** package is designed for Markov chain (MC) and hidden Markov (HM) models. It includes functions for maximum likelihood estimation of various versions of these models, both with and without covariates, and under different constraints. In particular, the package allows analyzing longitudinal and time-series data through MC models for univariate categorical responses and HM models for both categorical and continuous responses, so covering a wide range of applications.

The primary features of the models at issue, and of HM models in particular, are the great flexibility and the capability of performing dynamic model-based clustering with each cluster corresponding to a support point of the Markov chain. HM models are estimated using the Expectation-Maximization (EM) algorithm. Given that the likelihood is often multimodal, the package provides appropriate criteria to assess the convergence and choose different sets of starting values for the model parameters. Moreover, in order to optimize computational efficiency, the Fortran language is used for many numerical operations. The package also includes functions for simulating data from different versions of the HM models, as well as for displaying and visualizing parameter estimates. Parametric bootstrap can also be applied to provide standard errors for the parameters estimates, offering an alternative to using the information matrix.

In addition to the previous features, the **LMest** package also allows estimation of a model for longitudinal categorical data based on a mixture of latent AR(1) processes to dynamically account for unobserved heterogeneity. Each mixture component has a specific mean and correlation coefficient, but these components share a common variance. Therefore this model is based on a continuous latent process and is tailored for univariate data, in which the response variable has an ordinal nature. As another extension, it is possible to estimate mixed HM models (van de Pol and Langeheine, 1990) through function `lmestMixed()`, which is formulated to account for additional sources of time-fixed dependency in the data. The parameters of the latent process can vary across different latent subpopulations defined by an additional latent variable.

It is worth mentioning that, within the package, it is possible to handle intermittent, entirely or partially, missing values in the responses under the missing-at-random assumption and the EM algorithm for parameter estimation is suitable adapted also for providing an imputation of the missing responses. Weighted maximum likelihood can also be performed, which can be useful in many applied contexts, such as when longitudinal survey data are analyzed. In this case, individual survey weights (Kaplan and Ferguson, 1999), which refer to the probability of each unit being sampled in the reference population, are available; see among others, Pennoni and Genge (2020) and Pennoni and Nakai (2023). By using suitable weights, it is also possible to carry out causal inference with observational longitudinal data (Robins, 1997), in a potential outcome framework (Rubin, 1974), on the basis of a propensity score method. For applications of this type see Bartolucci et al. (2016, 2023) and Pennoni et al. (2023).

As alternative packages to **LMest**, we mention **march** (Maitre et al., 2020), which provides tools for fitting discrete-time Markov chains, semi-Markov chains, and higher-order models. Moreover, we mention the **mstate** package (Putter et al., 2007) designed for modeling and analyzing multi-state models, which are generalizations of Markov chains, useful in the context of survival analysis, and the **ctmcd** package (Pfeuffer, 2024), which provides methods for parameter estimation, simulation, and detailed analysis of continuous-time Markov models.

For the HM model, it is also worth considering the **depmixS4** package (Visser and Speekenbrink, 2022). **LMest** and **depmixS4** differ from others packages mainly in how covariates are parameterized

in the latent and the manifest models. Additionally, **depmixS4** can handle a more general class of distributional assumptions when covariates are included in the measurement model. We also highlight the recent proposal of Turner (2024) with the **eglhmm** package, which allows us to estimate extended generalized linear HM models for data conforming to various distributions. Another available package of interest is **msm** (Christopher H. Jackson, 2011), which is designed for estimating continuous-time Markov and HM models for longitudinal data. It provides both maximum likelihood estimation and Bayesian inference under various models, including multi-state models. Additionally, the **HiddenMarkov** package (Harte, 2021) offers tools for modeling time series data and supports various distributions for the observations, including Gaussian and Poisson distributions. Lastly, package **seqHMM** (Helske and Helske, 2019) is designed for fitting HM models and mixture HM models specifically for social sequence data and other types of categorical data.

The extensive development and availability of packages for HM models on CRAN highlights the broad applicability across diverse fields of such models. Their flexibility and robustness in handling various types of data, combined with their capability to uncover underlying unobserved processes, demonstrate the popularity of these models and their value among researchers and practitioners. Finally, as possible extension of the **LMest** package we consider that to handle informative missing data, as in the case of dropout from a longitudinal study. As described in Pandolfi et al. (2023), dropout can be accounted for by considering an absorbing state, which allows modeling survival time without specifying a separate survival model component. This extension will be included in future updates of the package. The package will also be extended to allow for variable selection, as recently proposed in Pennoni et al. (2024), where a greedy search algorithm is implemented to identify the most important response variables. Other forthcoming extensions include functions to estimate models for data with a multilevel structure, as proposed in Bartolucci et al. (2011), additional parameterizations for the covariates affecting the latent (sub)model, as described in Bartolucci et al. (2024b), and methods for dealing with compositional data, as proposed in Bartolucci et al. (2024a).

Acknowledgments

The authors are grateful for the financial support from the grant "*Hidden Markov Models for Early Warning Systems*" of Ministero dell'Università e della Ricerca (PRIN 2022TZEXKF) funded by the European Union - Next Generation EU, Mission 4, Component 2, CUP J53D23004990006.

References

H. Akaike. Information theory as an extension of the maximum likelihood principle. In B. N. Petrov and C. F., editors, *Second International symposium on information theory*, pages 267–281, Budapest, 1973. Akademiai Kiado. [p1, 8]

T. Anderson. Probability models for analyzing time changes in attitudes. In P. F. Lazarsfelsd, editor, *Mathematical Thinking in the Social Science*, pages 17–66. New York; Free Press, 1954. [p2]

A. Azzalini. Logistic regression for autocorrelated data with application to repeated measures. *Biometrika*, 81:767–775, 1994. [p3]

S. Bacci, S. Pandolfi, and F. Pennoni. A comparison of some criteria for states selection in the latent Markov model for longitudinal data. *Advances in Data Analysis and Classification*, 8:125–145, 2014. [p8]

F. Bartolucci and A. Farcomeni. A multivariate extension of the dynamic logit model for longitudinal data based on a latent Markov heterogeneity structure. *Journal of the American Statistical Association*, 104:816–831, 2009. [p8]

F. Bartolucci and A. Farcomeni. Information matrix for hidden Markov models with covariates. *Statistics and Computing*, 25:515–526, 2015. [p8]

F. Bartolucci and F. Pennoni. Impact evaluation of job training programs by a latent variable model. In *New Perspectives in Statistical Modeling and Data Analysis: Proceedings of the 7th Conference of the Classification and Data Analysis Group of the Italian Statistical Society, Catania, September 9-11, 2009*, pages 65–73. Springer-Verlag, 2011. [p3]

F. Bartolucci, F. Pennoni, and G. Vittadini. Assessment of school performance through a multilevel latent Markov Rasch model. *Journal of Educational and Behavioural Statistics*, 36:491–522, 2011. [p25]

F. Bartolucci, A. Farcomeni, and F. Pennoni. *Latent Markov Models for Longitudinal Data*. Chapman & Hall/CRC Press, Boca Raton, FL, 2013. [p1, 2, 8, 9]

F. Bartolucci, S. Bacci, and F. Pennoni. Longitudinal analysis of self-reported health status by mixture latent auto-regressive models. *Journal of the Royal Statistical Society, Series C*, 63:267–288, 2014a. [p8]

F. Bartolucci, A. Farcomeni, and F. Pennoni. Latent Markov models: A review of a general framework for the analysis of longitudinal data with covariates. *TEST*, 23:433–465, 2014b. [p2, 3]

F. Bartolucci, G. E. Montanari, and S. Pandolfi. Three-step estimation of latent Markov models with covariates. *Computational Statistics & Data Analysis*, 83:287–301, 2015. [p7]

F. Bartolucci, F. Pennoni, and G. Vittadini. Causal latent Markov model for the comparison of multiple treatments in observational longitudinal studies. *Journal of Educational and Behavioral Statistics*, 41: 146–179, 2016. [p24]

F. Bartolucci, S. Pandolfi, and F. Pennoni. LMest: An R package for latent Markov models for longitudinal categorical data. *Journal of Statistical Software*, 81:1–38, 2017. [p1, 7]

F. Bartolucci, S. Pandolfi, and F. Pennoni. Discrete latent variable models. *Annual Review of Statistics and its Application*, 9:425–452, 2022. [p1]

F. Bartolucci, F. Pennoni, and G. Vittadini. A causal latent transition model with multivariate outcomes and unobserved heterogeneity: Application to human capital development. *Journal of Educational and Behavioral Statistics*, 48:387–419, 2023. [p24]

F. Bartolucci, M. Greenacre, S. Pandolfi, and F. Pennoni. Hidden Markov and related discrete latent variable models: An application to compositional data. In G. Giordano, M. La Rocca, B. Niglio, M. Restaino, and M. Vichi, editors, *Studies in Classification, Data Analysis and Knowledge Organization*, CLADAG 2023, pages 1–8. Springer, 2024a. [p25]

F. Bartolucci, S. Pandolfi, and F. Pennoni. Parsimonious parametrizations of transition matrices of markov chain and hidden markov models. *Submitted*, pages 1–25, 2024b. [p25]

F. Bassi, F. Pennoni, and L. Rossetto. Market segmentation and dynamic analysis of sparkling wine purchases in italy. *Journal of Wine Economics*, 16:283–304, 2021. [p9]

L. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966. [p8]

L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970. [p8]

C. Bouveyron, G. Celeux, T. B. Murphy, and A. E. Raftery. *Model-based Clustering and Classification for Data Science: with Applications in R*. Cambridge University Press, Cambridge, UK, 2019. [p1]

Christopher H. Jackson. Multi-state models for panel data: The msm package for R. *Journal of Statistical Software*, 8:1–29, 2011. doi: 10.18637/jss.v038.i08. [p25]

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, Cambridge, MA, 1997. [p1]

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977. [p1, 8]

Y. Ephraim and N. Merhav. Hidden Markov processes. *IEEE Transactions on Information Theory*, 48: 1518–1569, 2002. [p1]

D. Harte. *HiddenMarkov: Hidden Markov Models*. Statistics Research Associates, Wellington, 2021. URL https://www.statsresearch.co.nz/dsh/sslib/. R package version 1.8-13. [p25]

J. Helske and S. Helske. seqhmm: Mixture hidden Markov models for social sequence data and other multivariate, multichannel categorical time series. *Journal of Statistical Software*, 88:1–32, 2019. [p25]

B. Juang and L. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33:251–272, 1991. [p9]

D. Kaplan and A. J. Ferguson. On the utilization of sample weights in latent variable models. *Structural equation modeling: A Multidisciplinary Journal*, 6:305–321, 1999. [p24]

R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley Sons, Hoboken, NJ, 2020. [p2, 13]

O. Maitre, K. Emery, with contributions from O. Buschor, and A. Berchtold. *march: Markov Chains*, 2020. URL https://CRAN.R-project.org/package=march. R package version 3.3.2. [p24]

S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer Science & Business Media, London, 2012. [p1]

B. Mor, S. Garhwal, and A. Kumar. A systematic review of hidden Markov models and their applications. *Archives of Computational Methods in Engineering*, 28:1429–1448, 2021. [p1]

L. J. Paas, J. K. Vermunt, and T. H. A. Bilmolt. Discrete time, discrete state latent Markov modelling for assessing and predicting household acquisitions of financial products. *Journal of the Royal Statistical Society, Series A*, 170:955–974, 2007. [p9]

S. Pandolfi, F. Bartolucci, and F. Pennoni. A hidden Markov model for continuous longitudinal data with missing responses and dropout. *Biometrical Journal*, 5:1–28, 2023. [p14, 25]

F. Pennoni and E. Genge. Analysing the course of public trust via hidden Markov models: a focus on the Polish society. *Statistical Methods & Applications*, 29:399–425, 2020. [p24]

F. Pennoni and M. Nakai. Exploring heterogeneity in happiness: Evidence from a Japanese longitudinal survey. In *Facets of Behaviormetrics: The 50th Anniversary of the Behaviormetric Society*, pages 193–217. Springer, 2023. [p24]

F. Pennoni and G. Vittadini. Two competing models for ordinal longitudinal data with time-varying latent effects: An application to evaluate hospital efficiency. *Quaderni di Statistica*, pages 53–68, 2013. [p8]

F. Pennoni, F. Bartolucci, G. Forte, and F. Ametrano. Exploring the dependencies among main cryptocurrency log-returns: A hidden Markov model. *Economic Notes*, 51:1–27, 2022. [p15]

F. Pennoni, L. J. Paas, and F. Bartolucci. A causal hidden Markov model for assessing effects of multiple direct mail campaigns. *TEST*, pages 1–29, 2023. [p24]

F. Pennoni, F. Bartolucci, and S. Pandolfi. Variable selection for hidden Markov models with continuous variables and missing data. *Journal of Classification*, pages 1–28, 2024. doi: https://doi.org/10.1007/s00357-024-09464-4. [p25]

M. Pfeuffer. *ctmcd: Estimating the Parameters of a Continuous-Time Markov Chain from Discrete-Time Data*, 2024. URL https://CRAN.R-project.org/package=ctmcd. R package version 1.4.4. [p24]

H. Putter, R. B. Geskus, and M. Fiocco. Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine*, 26:2389–2430, 2007. [p24]

P. Roback and J. Legler. *Beyond Multiple Linear Regression: Applied Generalized Linear Models and Multilevel Models in R*. Chapman and Hall/CRC Press, New York, 2021. [p22]

J. Robins. Causal inference from complex longitudinal data. In M. Berkame, editor, *Latent Variable Modeling and Applications to Causality*, volume 120 of *Lecture Notes in Statistics*, pages 69–117. Springer, New York, 1997. [p24]

D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66:688–701, 1974. [p24]

J. A. Ryan and J. M. Ulrich. *quantmod: Quantitative Financial Modelling Framework*, 2022. URL https://CRAN.R-project.org/package=quantmod. R package version 0.4.26. [p15]

J. L. Schafer. *mix: Estimation/Multiple Imputation for Mixed Categorical and Continuous Data*, 2024. URL https://CRAN.R-project.org/package=mix. R package version 1.0-12. [p20]

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978. [p1, 8]

R. Turner. *eglhmm: Extended Generalised Linear Hidden Markov Models*, 2024. URL https://CRAN.R-project.org/package=eglhmm. R package version 0.1-3. [p25]

F. van de Pol and R. Langeheine. Mixed Markov latent class models. *Sociological Methodology*, 20:213–247, 1990. [p24]

I. Visser and M. Speekenbrink. *Mixture and Hidden Markov Models with R*. Springer, Cham, CH, 2022. [p8, 24]

A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967. [p9]

L. Wiggins. *Panel Analysis: Latent Probability Models for Attitude and Behaviour Processes*. Elsevier, Amsterdam, 1973. [p1]

A. Zeileis and Y. Croissant. Extended model formulas in R: Multiple parts and multiple responses. *Journal of Statistical Software*, 34:1–13, 2010. doi: 10.18637/jss.v034.i01. [p2, 5]

W. Zucchini, I. L. MacDonald, and R. Langrock. *Hidden Markov Models for Time Series: An Introduction using R*. CRC press, Boca Raton, FL, 2016. [p1]

Fulvia Pennoni
University of Milano-Bicocca
Department of Statistics and Quantitative Methods
Milan, Italy
https://sites.google.com/view/fulviapennoni/home
ORCiD: 0000-0002-6631-7211
fulvia.pennoni@unimib.it

Silvia Pandolfi
University of Perugia
Department of Economics
Perugia, Italy
https://sites.google.com/site/spandolfihome/home
ORCiD: 0000-0002-6631-1211
silvia.pandolfi@unipg.it

Francesco Bartolucci
University of Perugia
Department of Economics, Via A. Pascoli 8
Perugia, Italy
https://sites.google.com/site/bartstatistics/
ORCiD: 0000-1721-1511-1101
francesco.bartolucci@unipg.it