Illustration of the LMest package

Fulvia Pennoni*, Silvia Pandolfi° and Francesco Bartolucci°

*University of Milano-Bicocca, °University of Perugia

LMest package

- The package `LMest` allows us to specify and fit Latent (or Hidden) Markov (LM) models, which are tailored to the analysis of longitudinal continuous and categorical data

- It contains functions for parameter estimation, via the Expectation-Maximization (EM) algorithm, of the basic LM model and its extensions with individual covariates that are included by means of suitable parameterizations

- It also includes functions for simulation from these models, for performing model selection (that also includes the search of the global maximum likelihood), and for local and global decoding

LMest package

- Finally, standard errors for model parameters are available, which are based on numerical or exact methods and on parametric bootstrap

- This document introduces the LMest's basic set of tools, and shows how to apply them for the different model specifications allowed in the package, with data frames in different formats

- See `help(package="LMest")` for further details and `citation("LMest")` for main references

```
library(LMest)
```

Main functions

LMest contains the following main functions having specific tasks:

- `lmestData()`: to manipulate data in long format
- `lmestFormula()`: to build the formulas specifying the model to be estimated
- `lmest()`: to estimate LM models for categorical responses with or without covariates
- `lmestCont()`: to estimate LM models for continuous outcomes with or without covariates

Main functions

- `lmestMixed()`: to estimate LM models for categorical responses with initial and transition probabilities of the latent process allowed to vary across different latent subpopulations

- `lmestSearch()`: to search for the global maximum of the model log-likelihood and to select the optimal number of latent states

- `lmestDecoding()`: to perform local and global decoding (Viterbi algorithm)

- `bootstrap()`: to perform bootstrap parametric resampling to compute standard errors for parameter estimates

- `drawLMbasic()`: to draw samples from the basic LM model with categorical responses

Data: RLMSlong

- This dataset, already in the package, concerns the evaluation of job satisfaction of n = 1,718 individuals followed at T = 7 years from 2008 to 2014

- The data come from the Russia Longitudinal Monitoring Survey, and are documented in `?RLMSlong`

- The response variable (named `value`), corresponding to the reported job satisfaction at different time occasions, has five ordered categories from `1: absolutely satisfied` to `5: absolutely not satisfied`

- The longitudinal dataset is in long format:

```
data("RLMSlong")
dim(RLMSlong)
#> [1] 12026     4
```

```
head(RLMSlong)
#>   time id  rlms value
#> 1    1  1 IKSJQ     2
#> 2    1  2 IKSJQ     2
#> 3    1  3 IKSJQ     2
#> 4    1  4 IKSJQ     2
```

Data: PSIDlong

- This dataset, downloadable through the package `psidR`, is derived from the Panel Study of Income Dynamics (http://psidonline.isr.umich.edu)

- The data used in the following application concern n = 1,446 women who were followed from 1987 to 1993

- There are two binary response variables: `Y1Fertility` (indicating whether a woman had given birth to a child in a certain year) and `Y2Employment` (indicating whether she was employed)

- The covariates are: `X1Race` (dummy variable equal to 1 for a black woman); `X2Age` (in 1986, rescaled by its maximum value); `X3Age2` (squared age); `X4Education` (number of years of schooling); `X5Child1_2` (number of children in the family aged between 1 and 2 years, referred to the previous year); `X6Child3_5`; `X7Child6_13`; `X8Child14`; `X9Income` of the husband (in dollars, referred to the previous year, divided by 1,000)

Data: PSIDlong

```
load("PSID_long.RData")
dim(PSID_long)
#> [1] 10122    13
```

```
head(PSID_long)
#>   id time Y1Fertility Y2Employment X1Race X2Age X3Age2 X4Education X5Child1_2
#> 1  1    1           0            1      0    -8   0.64          12          0
#> 2  1    2           0            1      0    -8   0.64          12          0
#> 3  1    3           0            1      0    -8   0.64          12          0
#> 4  1    4           0            1      0    -8   0.64          12          0
#> 5  1    5           0            1      0    -8   0.64          12          0
#> 6  1    6           0            1      0    -8   0.64          12          0
#>   X6Child3_5 X7Child6_13 X8Child14 X9Income
#> 1          0           0         0   35.343
#> 2          0           2         0   28.157
#> 3          0           2         0   33.250
#> 4          0           2         0   37.018
#> 5          0           2         0   39.852
#> 6          0           1         1   38.708
```

Data: NLSYlong

- This dataset, downloadable from the package `panelr`, is derived from the National Longitudinal Survey of Youth (https://www.nlsinfo.org/content/cohorts/nlsy79)

- The data are a subset concerning n = 581 individuals who were followed from 1990 to 1994

- We consider two response variables: `anti`, providing a measure of antisocial behavior (measured on a scale ranging from 0 to 6), `self`, which is a measure of self-esteem (measured on a scale ranging from 6 to 24)

- The covariates are the following: `momage`, a continuous variable indicating the age of the mother at birth; `gender`, a dummy variable equal to 1 for males; `childage`, a continuous variable indicating the age of the child at the first interview; `momwork`, a dummy variable equal to 1 if mother works; `married`, a dummy variable equal to 1 if parents are married; `hispanic` and `black`,

two dummy variables for ethnicity; pov, a time-varying binary variable indicating the poverty status of the family at years 1990, 1992 and 1994

Data: NLSYlong

- In the following we use the function `reshape` to obtain data in long format:

```
require("panelr")
data("nlsy")
NLSYlong <- reshape(as.data.frame(nlsy),
                    times = c(1, 2, 3),direction = "long", sep  = "9",
                    varying = c("anti90","anti92", "anti94", "self90","self92", "self94",
                                "pov90","pov92","pov94"))
```

```
head(NLSYlong)
#>     momage gender childage hispanic black momwork married time anti self pov id
#> 1.0     21      1 8.000000        0     0       0       0    0    1   21   1  1
#> 2.0     22      1 8.416667        0     0       1       0    0    0   20   0  2
#> 3.0     18      0 8.083333        0     0       0       1    0    5   21   0  3
#> 4.0     24      0 8.250000        0     0       0       0    0    2   23   0  4
#> 5.0     22      1 9.333333        0     0       0       0    0    1   22   0  5
#> 6.0     24      0 8.583333        0     0       0       0    0    1   19   0  6
```

Data: data_criminal_sim

- This simulated dataset, already in the LMest package, contains n = 60,000 observations related to the complete conviction histories of a cohort of offenders followed from the age of criminal responsibility, 10 years, until 40 years, and also includes the proportion of non-offenders

- We consider $T = 6$ age bands of length equal to five years. For every age band, we dispose of a binary variable equal to 1 if the subject has been convicted for a crime of the following ten offense groups and to 0 otherwise

- The typologies of crime are: `y1 violence against the person`, `y2 sexual offenses`, `y3 burglary`, `y4 robbery`, `y5 theft and handling stolen goods`, `y6 fraud and forgery`, `y7 criminal damage`, `y8 drug offenses`, `y9 motoring offenses`, `y10 other offenses`

Data: data_criminal_sim

- `Gender`, is a covariate coded as 1 for male and 2 for female:

```
data(data_criminal_sim)
dim(data_criminal_sim)
#> [1] 60000    13
```

```
tail(data_criminal_sim)
#>             id sex time y1 y2 y3 y4 y5 y6 y7 y8 y9 y10
#> [59995,] 10000   2    1  0  0  0  0  0  0  0  0  0   0
#> [59996,] 10000   2    2  0  0  0  0  0  0  0  0  0   0
#> [59997,] 10000   2    3  0  0  0  0  0  0  0  0  0   0
#> [59998,] 10000   2    4  0  0  0  0  0  0  0  0  0   0
#> [59999,] 10000   2    5  0  0  0  0  0  0  0  0  0   0
#> [60000,] 10000   2    6  0  0  0  0  0  0  0  0  0   0
```

Data with `lmestData()`

- Function `lmestData()` allows us to check and prepare the data. For example, for the `NLSYlong` dataset we have:

```
dt <- lmestData(data = NLSYlong, id = "id", time="time",
                responsesFormula= anti+self ~NULL)
```

```
summary(dt, dataSummary="responses", varType=rep("c",ncol(dt$Y)))
#>
#> Data Info:
#> ----------
#>
#> Observations:        581
#> Time occasions:      3
#> Variables:           2
#>
#>
#> Summary:
#> ----------
#>
#>       anti                self
#>  Min.   :0.000000   Min.   : 6.0000
#>  1st Qu.:0.000000   1st Qu.:18.0000
#>  Median :1.000000   Median :21.0000
#>  Mean   :1.636833   Mean   :20.3502
#>  3rd Qu.:3.000000   3rd Qu.:23.0000
#>  Max.   :6.000000   Max.   :24.0000
#>
#> Summary by year:
#> ----------
```

```
#>
#>
#> Time =  0
#>
#>       anti               self
#>  Min.   :0.000000   Min.   : 9.00000
#>  1st Qu.:0.000000   1st Qu.:18.00000
#>  Median :1.000000   Median :21.00000
#>  Mean   :1.567986   Mean   :20.07057
#>  3rd Qu.:2.000000   3rd Qu.:23.00000
#>  Max.   :6.000000   Max.   :24.00000
#>
#> Time =  2
#>
#>       anti               self
#>  Min.   :0.000000   Min.   : 6.00000
#>  1st Qu.:0.000000   1st Qu.:18.00000
#>  Median :1.000000   Median :21.00000
#>  Mean   :1.595525   Mean   :20.36213
#>  3rd Qu.:3.000000   3rd Qu.:23.00000
#>  Max.   :6.000000   Max.   :24.00000
#>
#> Time =  4
#>
#>       anti               self
#>  Min.   :0.000000   Min.   : 9.0000
#>  1st Qu.:0.000000   1st Qu.:19.0000
#>  Median :1.000000   Median :21.0000
#>  Mean   :1.746988   Mean   :20.6179
#>  3rd Qu.:3.000000   3rd Qu.:23.0000
#>  Max.   :6.000000   Max.   :24.0000
```

Data with `lmestData()`

- We can display the summary of every response variable for each time occasion
- For the criminal data, if we select only females, we have

```
data_criminal_sim<-data.frame(data_criminal_sim)
crimf <- data_criminal_sim[data_criminal_sim$sex == 2,]
dt1 <- lmestData(data = crimf, id = "id", time = "time")
summary(dt1, varType = rep("d",ncol(dt1$Y)))
```

```
summary(dt1, varType = rep("d",ncol(dt1$Y)))
#>
#> Data Info:
#> ----------
#>
#> Observations:        5200
#> Time occasions:      6
#> Variables:           11
#>
#>
#> Proportion:
#> ----------
#>
#>  time     sex y1          y2          y3          y4          y5
#>  1:5200   2:1 0:0.9838782 0:0.9971474 0:0.9796154 0:0.9981731 0:0.9427564
#>  2:5200       1:0.0161218 1:0.0028526 1:0.0203846 1:0.0018269 1:0.0572436
#>  3:5200
#>  4:5200
#>  5:5200
```

```
#>  6:5200
#>  y6          y7          y8         y9          y10
#>  0:0.9909615 0:0.9839103 0:0.991859 0:0.9987179 0:0.9849679
#>  1:0.0090385 1:0.0160897 1:0.008141 1:0.0012821 1:0.0150321
#>
#>
#>
#>
#>
#> Proportion by year:
#> ----------
#>
#>
#> Time =  1
#>
#>  time     sex y1          y2          y3          y4          y5
#>  1:5200   2:1 0:0.9886538 0:0.9982692 0:0.9696154 0:0.9990385 0:0.9328846
#>               1:0.0113462 1:0.0017308 1:0.0303846 1:0.0009615 1:0.0671154
#>  y6          y7          y8          y9          y10
#>  0:0.9957692 0:0.9830769 0:0.9963462 0:0.9990385 0:0.9884615
#>  1:0.0042308 1:0.0169231 1:0.0036538 1:0.0009615 1:0.0115385
#>
#> Time =  2
#>
#>  time     sex y1          y2          y3          y4          y5
#>  2:5200   2:1 0:0.9755769 0:0.9942308 0:0.9598077 0:0.9965385 0:0.8901923
#>               1:0.0244231 1:0.0057692 1:0.0401923 1:0.0034615 1:0.1098077
#>  y6          y7          y8          y9          y10
#>  0:0.9832692 0:0.9696154 0:0.9880769 0:0.9973077 0:0.9707692
#>  1:0.0167308 1:0.0303846 1:0.0119231 1:0.0026923 1:0.0292308
#>
#> Time =  3
#>
#>  time     sex y1      y2          y3      y4          y5          y6
#>  3:5200   2:1 0:0.975 0:0.9963462 0:0.975 0:0.9973077 0:0.9290385 0:0.9871154
#>               1:0.025 1:0.0036538 1:0.025 1:0.0026923 1:0.0709615 1:0.0128846
#>  y7          y8          y9       y10
```

```
#>  0:0.9778846 0:0.9859615 0:0.9975 0:0.9773077
#>  1:0.0221154 1:0.0140385 1:0.0025 1:0.0226923
#>
#> Time =  4
#>
#>  time     sex y1       y2          y3          y4          y5
#>  4:5200   2:1 0:0.9825 0:0.9965385 0:0.9861538 0:0.9980769 0:0.9523077
#>              1:0.0175 1:0.0034615 1:0.0138462 1:0.0019231 1:0.0476923
#>  y6          y7          y8          y9          y10
#>  0:0.9901923 0:0.9867308 0:0.9907692 0:0.9994231 0:0.9876923
#>  1:0.0098077 1:0.0132692 1:0.0092308 1:0.0005769 1:0.0123077
#>
#> Time =  5
#>
#>  time     sex y1          y2          y3          y4          y5
#>  5:5200   2:1 0:0.9873077 0:0.9982692 0:0.9919231 0:0.9984615 0:0.97
#>              1:0.0126923 1:0.0017308 1:0.0080769 1:0.0015385 1:0.03
#>  y6          y7          y8          y9          y10
#>  0:0.9938462 0:0.9909615 0:0.9936538 0:0.9994231 0:0.9909615
#>  1:0.0061538 1:0.0090385 1:0.0063462 1:0.0005769 1:0.0090385
#>
#> Time =  6
#>
#>  time     sex y1          y2          y3          y4          y5
#>  6:5200   2:1 0:0.9942308 0:0.9992308 0:0.9951923 0:0.9996154 0:0.9821154
#>              1:0.0057692 1:0.0007692 1:0.0048077 1:0.0003846 1:0.0178846
#>  y6          y7          y8          y9          y10
#>  0:0.9955769 0:0.9951923 0:0.9963462 0:0.9996154 0:0.9946154
#>  1:0.0044231 1:0.0048077 1:0.0036538 1:0.0003846 1:0.0053846
```

Formulas with `lmestFormula()`

- Function `lmestFormula()` allows us to specify the model to be estimated. In particular, the formula for the basic LM model is specified as:

```
fmBasic <- lmestFormula(data = RLMSlong, response = "value")
```

- The formula for the LM model with all the covariates affecting the distribution of the latent process may be specified as:

```
fmLatent <- lmestFormula(data = PSID_long, response = "Y",
                         LatentInitial = "X", LatentTransition ="X")
```

- Alternatively, it is possible to specify that a subset of the covariates (for example, `X1Race`, `X2Age`, `X3Age2`, and `X9Income`) affects the initial and transition probabilities:

```
fmLatent2 <- lmestFormula(data = PSID_long, response = "Y",
                          LatentInitial = c("X1","X2","X3","X9"),
                          LatentTransition =c("X1","X2","X3","X9"))
```

Covariates in the latent Markov with `lmest()`

- Function `lmest()` allows us to estimate LM models for categorical responses with different model specifications

- These models rely on a homogenous Markov chain of first order with a finite number of states. Maximum likelihood estimation of model parameters is performed through the EM algorithm

- Standard errors for the parameter estimates are obtained by exact computation of the information matrix or through reliable numerical approximations of this matrix, by using option `out_se=TRUE`

Covariates in the latent Markov with `lmest()`

- For the data `PSID_long`, we can estimate an LM model with covariates affecting the distribution of the latent process by fixing k=3 latent states as follows:

```
mod <- lmest(responsesFormula = fmLatent$responsesFormula,
             latentFormula =  fmLatent$latentFormula,
             index = c("id","time"),
             data = PSID_long, k = 3,
             paramLatent = "multilogit",
             start = 0, out_se=TRUE)
```

- Every 10 iterations of the EM algorithm, the function displays, along with the indication of the chosen model specification and number of latent states:
  - *type of starting values used*
  - *number of iterations*
  - *value of the log-likelihood at the end of the current iteration*
  - *difference with respect to the log-likelihood at the end of the previous iteration*
  - *discrepancy between the corresponding parameter vectors*

Covariates in the latent Markov with `lmest()`

- We can display the estimation results with:

```
print(mod)
#>
#> Basic Latent Markov model with covariates in the latent model
#> Call:
#> lmest(responsesFormula = fmLatent$responsesFormula, latentFormula = fmLatent$latentFormula,
#>     data = PSID_long, index = c("id", "time"), k = 3, start = 0,
#>     paramLatent = "multilogit", out_se = TRUE)
#>
#> Available objects:
#>  [1] "lk"          "Be"          "Ga"          "Psi"         "np"
#>  [6] "k"           "aic"         "bic"         "lkv"         "n"
#> [11] "TT"          "paramLatent" "sePsi"       "seBe"        "seGa"
#> [16] "V"           "PI"          "Piv"         "Ul"          "Lk"
#> [21] "Bic"         "Aic"         "call"        "data"
#>
#> Convergence info:
#>      LogLik np k      AIC      BIC    n TT
#>   -6479.536 86 3 13131.07 13584.86 1446  7
```

Covariates in the latent Markov with `lmest()`

- Results:

```
summary(mod)
#> Call:
#> lmest(responsesFormula = fmLatent$responsesFormula, latentFormula = fmLatent$latentFormula,
#>     data = PSID_long, index = c("id", "time"), k = 3, start = 0,
#>     paramLatent = "multilogit", out_se = TRUE)
#>
#> Coefficients:
#>
#>  Be - Parameters affecting the logit for the initial probabilities:
#>           logit
#>                  2       3
#>   int       -0.7772 -9.4113
#>   race       0.2008  0.0971
#>   age       -0.0732 -0.6433
#>   age^2     -0.7285 -1.8802
#>   education  0.1667  0.4680
#>   child1_2  -1.0639 -0.8229
#>   child3_5  -0.4553 -1.1313
#>   child6_13  0.1794 -5.3114
#>   child14    0.0541 -0.3946
#>   income    -0.0157 -0.0302
#>
#>  Standard errors for Be:
#>           logit
#>                 2      3
#>   int       0.7557 1.4352
```

```
#>   race      0.1957 0.2937
#>   age       0.0848 0.1574
#>   age^2     0.4000 0.5957
#>   education 0.0457 0.0709
#>   child1_2  0.1853 0.2221
#>   child3_5  0.1379 0.2621
#>   child6_13 0.7087 0.0040
#>   child14   0.1671 0.6825
#>   income    0.0050 0.0089
#>
#>  Ga - Parameters affecting the logit for the transition probabilities:
#> , , logit = 1
#>
#>            logit
#>                  2        3
#>   int       -3.8966  -1.2800
#>   race      -0.1734 -10.7354
#>   age        0.0817  -1.4384
#>   age^2      0.4880  -7.3508
#>   education  0.1844  -0.7366
#>   child1_2  -0.6265   2.1211
#>   child3_5  -0.0339  -2.1144
#>   child6_13  0.2394  -1.9266
#>   child14    0.0957  -0.3076
#>   income    -0.0104  -0.0144
#>
#> , , logit = 2
#>
#>            logit
#>                  2       3
#>   int       -2.4441 -9.2742
#>   race      -0.4889 -0.7600
#>   age        0.0562 -1.3660
#>   age^2      0.5436 -5.1836
#>   education -0.0933 -0.1280
#>   child1_2   0.2631  0.7704
#>   child3_5   0.6473  0.9291
```

```
#>   child6_13  0.2232 -0.6728
#>   child14    0.2684 -1.5742
#>   income     0.0053  0.0132
#>
#> , , logit = 3
#>
#>            logit
#>                   2       3
#>   int       -5.4270  5.8740
#>   race      -0.7386 -0.8557
#>   age       -0.6811 -0.2118
#>   age^2     -2.1880 -2.1636
#>   education -0.2457 -0.7374
#>   child1_2  -0.7313  2.0489
#>   child3_5   1.2982  4.3168
#>   child6_13 -3.0499  1.6407
#>   child14    1.7739  2.3328
#>   income     0.0491  0.0333
#>
#>
#>  Standard errors for Ga:
#> , , logit = 1
#>
#>            logit
#>                  2      3
#>   int       0.8240 3.6382
#>   race      0.2170 0.1675
#>   age       0.0828 0.9790
#>   age^2     0.3696 5.1201
#>   education 0.0470 0.3881
#>   child1_2  0.2055 0.9048
#>   child3_5  0.1365 1.3198
#>   child6_13 0.0781 1.4494
#>   child14   0.1344 0.9903
#>   income    0.0036 0.0269
#>
#> , , logit = 2
```

```
#>
#>            logit
#>                  2      3
#>   int       0.9412 3.0657
#>   race      0.2353 0.4452
#>   age       0.0942 0.4578
#>   age^2     0.4189 1.9398
#>   education 0.0573 0.1172
#>   child1_2  0.2564 0.4119
#>   child3_5  0.1896 0.3778
#>   child6_13 0.1129 0.2785
#>   child14   0.1348 0.9131
#>   income    0.0061 0.0058
#>
#> , , logit = 3
#>
#>            logit
#>                  2      3
#>   int       3.9628 3.6052
#>   race      0.6155 0.6751
#>   age       0.4696 0.3130
#>   age^2     1.5422 1.2060
#>   education 0.1319 0.2063
#>   child1_2  0.4986 0.7472
#>   child3_5  0.4699 0.9911
#>   child6_13 2.8523 0.5717
#>   child14   1.2542 1.1151
#>   income    0.0127 0.0147
#>
#>
#>  Psi - Conditional response probabilities:
#> , , item = 1
#>
#>         state
#> category      1 2      3
#>        0 0.9135 1 0.7517
#>        1 0.0865 0 0.2483
```

```
#>
#> , , item = 2
#>
#>         state
#> category      1      2      3
#>        0 0.9192 0.0265 0.1266
#>        1 0.0808 0.9735 0.8734
#>
#>
#>  Standard errors for the conditional response probability matrix:
#> , , item = 1
#>
#>         state
#> category      1 2      3
#>        0 0.0056 0 0.0199
#>        1 0.0056 0 0.0199
#>
#> , , item = 2
#>
#>         state
#> category      1      2      3
#>        0 0.0091 0.0042 0.0146
#>        1 0.0091 0.0042 0.0146
```

Covariates in the latent Markov with `lmest()`

- To get an insight on the results, we observe that the second latent state corresponds to women with the lowest propensity to fertility and the highest propensity to have a job

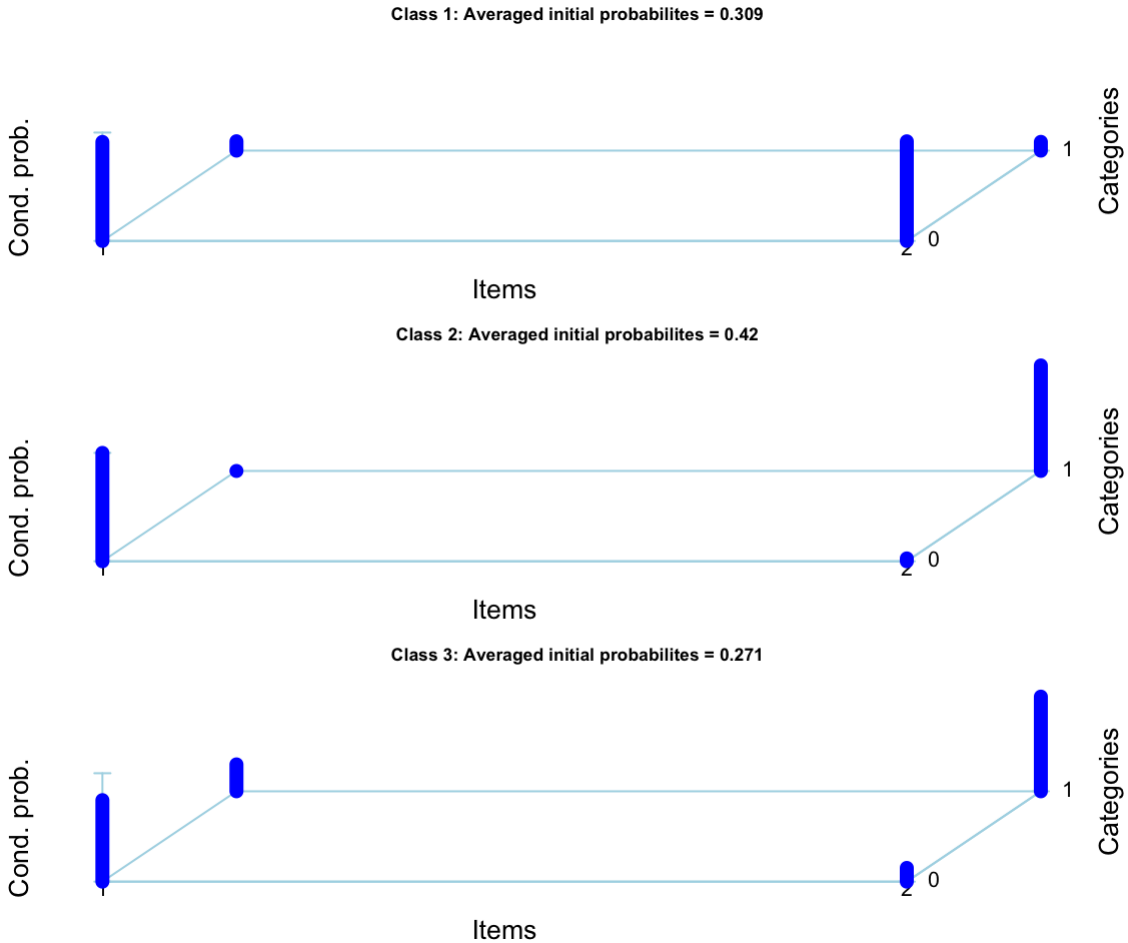
- The first state corresponds to women with both low propensity to fertility and to have a job position and the third state is associated to intermediate levels

- The averaged estimated transition matrix shows a quite high persistence in the same latent state with the exception of the third state for which we observe a probability of around 0.6 of moving towards the second state

Covariates in the latent Markov with `lmest()`

- A plot of the conditional response probabilities referred to the categories of the multivariate response is obtained as:

```
plot(mod,what="Cond.Prob.")
```

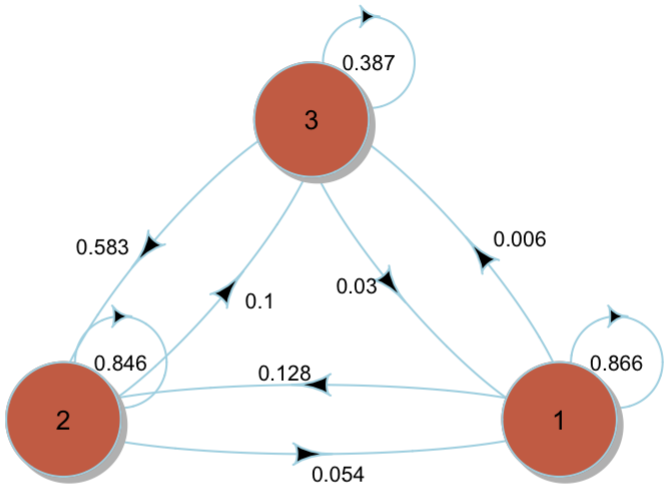

Class 1: Averaged initial probabilites = 0.309
Cond. prob.
Items
Categories
1
0
Class 2: Averaged initial probabilites = 0.42
Cond. prob.
Items
Categories
1
0
Class 3: Averaged initial probabilites = 0.271
Cond. prob.
Items
Categories
1
0

Covariates in the latent Markov with `lmest()`

- A path diagram of the estimated transition probabilities is obtained as:

```
plot(mod,what="Transitions")
```

Averaged transition probabilities



Latent Markov model for continuous outcomes and covariates with

`lmestCont()`

- The LM model for continuous outcomes may be estimated by using function `lmestCont()`

- For the data `NLSYlong` we estimate a multivariate LM model with covariates in the latent process

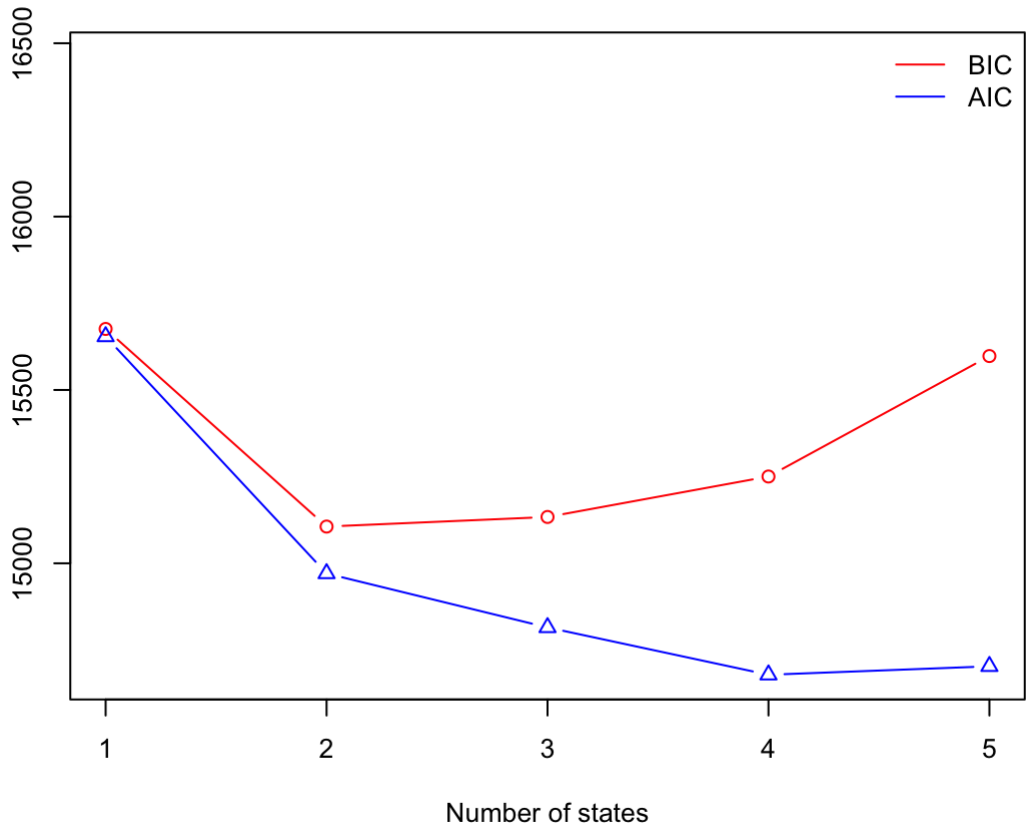
- The selection of the number of latent states can be performed by setting option `k` in a suitable way:

```
mod1 <- lmestCont(responsesFormula = anti + self  ~ NULL,
                  latentFormula = ~ gender + childage + hispanic + black+ pov +
                    momwork + married|   gender + childage + hispanic + black+ pov+
                    momwork + married,
                  index = c("id", "time"), data = dt$data,
                  k = 1:5, modBasic=1,
                  out_se = TRUE, output = TRUE, tol=10^-5)
```

Function `lmestCont()`

- We can display the estimation results with a plot of the BIC and AIC indices:

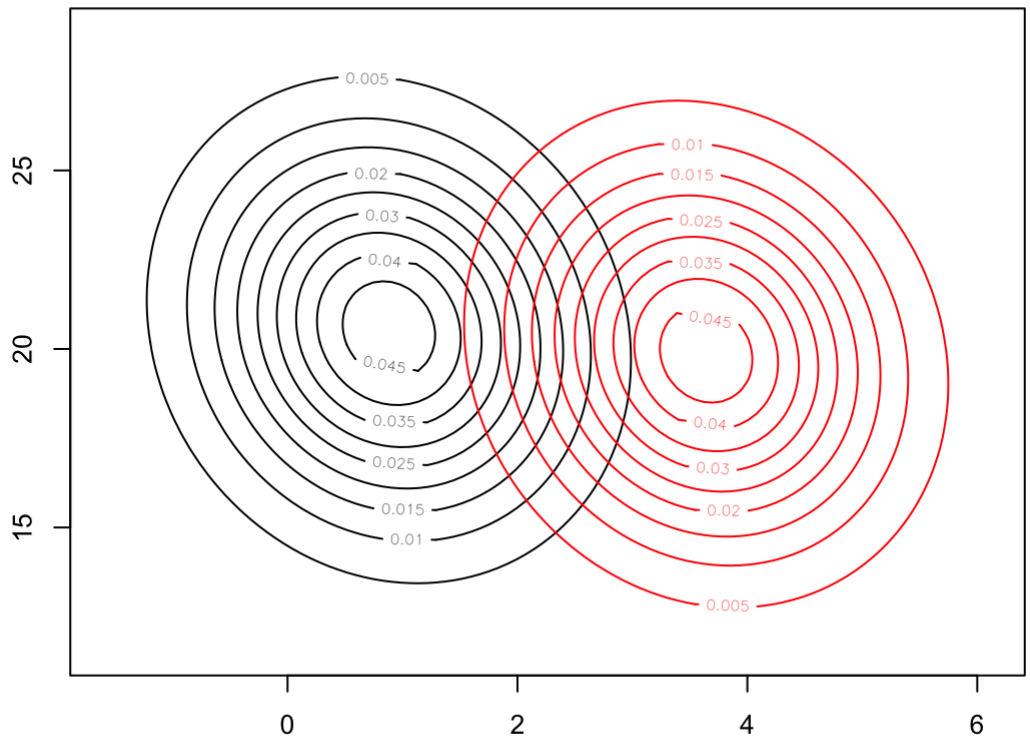
```
plot(mod1,what="modSel")
```



Function `lmestCont()`

- A plot of the ellipse of the bivariate densities is obtained as:

```
plot(mod1,what="Density")
```

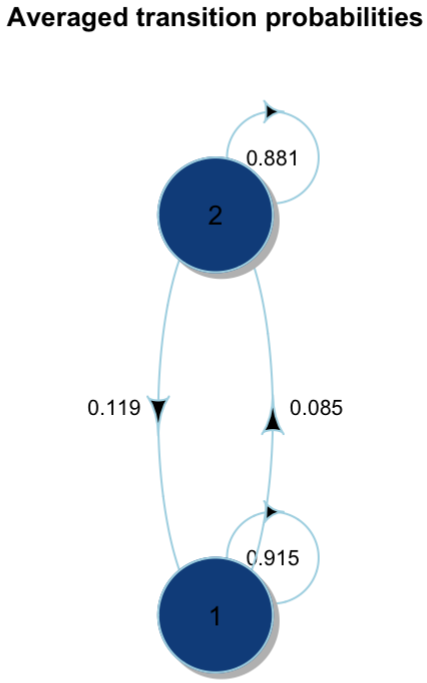


- The latent states are ordered according to increasing values of antisocial behavior

Function `lmestCont()`

- The path diagram of the estimated transition probabilities is obtained as:

```
plot(mod1,what="Transitions")
```



- We observe a probability of around 0.12 of moving from the second to the first state over years, indicating that they progress towards a more social behvior

Function `lmestCont()`

- Other results and asymptotic tests:

```
kable(round(TabBe,3))
```

	estBe(2)	s.e.Be(2)	t-test(2)
int	-0.622	1.784	-0.348
gender	-0.982	0.251	-3.914
childage	-0.046	0.196	-0.236
hispanic	-0.624	0.352	-1.773
black	0.098	0.269	0.363
pov	0.804	0.264	3.052
momwork	0.466	0.248	1.880
married	-0.034	0.282	-0.121

- The argument `Be`, returned by the function, contains the estimated regression parameters affecting the distribution of the initial probabilities:

- *gender log-odds ratios are all negative and significant, indicating that males tend to be in the second state at the first wave*

Function `lmestCont()`

```
kable(round(TabGa1,3))
```

	estGa(1)	estGa2(2)	s.e.Ga(1)	s.e.Ga(2)	t-test(1)	t-test(2)
int	-7.602	4.183	3.140	3.423	-2.421	1.222
gender	-0.522	-0.493	0.355	0.719	-1.469	-0.686
childage	0.595	-0.722	0.334	0.392	1.781	-1.840
hispanic	-0.388	0.354	0.477	0.828	-0.814	0.427
black	-0.246	0.496	0.436	0.698	-0.565	0.711
pov	1.195	0.409	0.409	0.643	2.924	0.636
momwork	-0.449	-0.093	0.397	0.541	-1.132	-0.172
married	-0.413	0.096	0.438	0.623	-0.942	0.154

- Output Ga contains the estimated parameters affecting the distribution of the transition probabilities:

- *the parameter estimate measuring the influence of the poverty on the transition from the first to the second state is positive and significant, indicating that young people having a family with poor economic conditions tend to move from the first to the second state more likely than those with good economic conditions*

Mixed Latent Markov model

`lmestMixed()`

- Function `lmestMixed()` allows us to estimate mixed LM models for categorical responses to take into account additional sources of (time-fixed) dependence in the data

- For the data `data_criminal_sim` we are interested to evaluate the patterns of criminal behavior among individuals

- At this aim, we estimate a model with k_1 = 2 latent classes and k_2 = 2 latent states, restricting the analysis to females

```
responsesFormula <- lmestFormula(data = crimf,response = "y")$responsesFormula

out <- lmestMixed(responsesFormula = responsesFormula,
                  index = c("id","time"),
                  k1 = 2, k2 = 2,
                  data = crimf)
```

Mixed Latent Markov model

`lmestMixed()`

- Results:

```
summary(out)
#> Call:
#> lmestMixed(responsesFormula = responsesFormula, data = crimf,
#>     index = c("id", "time"), k1 = 2, k2 = 2)
#>
#> Coefficients:
#>
#> Mass probabilities:
#> [1] 0.2175 0.7825
#>
#> Initial probabilities:
#>    u
#> v   1      2
#>   1 1 0.9087
#>   2 0 0.0913
#>
#> Transition probabilities:
#> , , u = 1
#>
#>    v1
#> v0       1      2
#>   1 0.8525 0.1475
#>   2 0.6414 0.3586
#>
#> , , u = 2
#>
```

```
#>    v1
#> v0       1      2
#>   1 1.0000 0.0000
#>   2 0.3382 0.6618
#>
#>
#> Conditional response probabilities:
#> , , j = 1
#>
#>    v
#> y        1      2
#>   0 0.9952 0.8242
#>   1 0.0048 0.1758
#>
#> , , j = 2
#>
#>    v
#> y        1      2
#>   0 0.9983 0.9809
#>   1 0.0017 0.0191
#>
#> , , j = 3
#>
#>    v
#> y        1      2
#>   0 0.9963 0.7436
#>   1 0.0037 0.2564
#>
#> , , j = 4
#>
#>    v
#> y        1      2
#>   0 0.9999 0.9737
#>   1 0.0001 0.0263
#>
#> , , j = 5
#>
```

```
#>    v
#> y        1      2
#>   0 0.9773 0.4546
#>   1 0.0227 0.5454
#>
#> , , j = 6
#>
#>    v
#> y        1      2
#>   0 0.9982 0.8892
#>   1 0.0018 0.1108
#>
#> , , j = 7
#>
#>    v
#> y        1      2
#>   0 0.9957 0.8177
#>   1 0.0043 0.1823
#>
#> , , j = 8
#>
#>    v
#> y        1      2
#>   0 0.9976 0.9105
#>   1 0.0024 0.0895
#>
#> , , j = 9
#>
#>    v
#> y        1      2
#>   0 0.9999 0.9815
#>   1 0.0001 0.0185
#>
#> , , j = 10
#>
#>    v
#> y        1      2
```

```
#>   0 0.9987 0.7912
#>   1 0.0013 0.2088
```

Mixed Latent Markov model

`lmestMixed()`

```
kable(round(out$Psi[2, , ], 3))
```

1	2	3	4	5	6	7	8	9	10
0.005	0.002	0.004	0.000	0.023	0.002	0.004	0.002	0.000	0.001
0.176	0.019	0.256	0.026	0.545	0.111	0.182	0.090	0.019	0.209

- We can identify the first latent state as that of females with null or very low tendency to commit crimes, whereas the second latent state corresponds to criminals having mainly as type of activity: theft, burglary, and other offenses

- According to the estimated transition matrix, females classified in the first cluster present a higher probability (of around 0.64) to move from the second to the first latent state than those assigned to the second cluster (of around 0.34), revealing a more pronounced tendency to commit less crimes over time

Model selection with `lmestSearch()`

- Function `lmestSearch()` allows us to deal with both model selection and multimodality of the likelihood function

- Two main criteria are provided to select the number of latent states: the Akaike Information Criterion and the Bayesian Information Criterion

- For example, for the `RLMSlong` dataset we can estimate the basic LM model for increasing values of the latent states k ranging from 1 to 4:

- We can display the results of the model selection with:

```
summary(outs)
#> Call:
#> lmestSearch(responsesFormula = fmBasic$responsesFormula, data = RLMSlong,
#>     index = c("id", "time"), k = 1:4, version = "categorical",
#>     seed = 123, modBasic = 1)
#>
#>  states        lk np      AIC      BIC
#>       1 -14943.73  4 29895.45 29917.25
#>       2 -13921.09 11 27864.18 27924.12
#>       3 -13557.20 20 27154.41 27263.39
#>       4 -13392.93 31 26847.86 27016.78
```

- The minimum BIC index corresponds to the model with k=3 latent states and the model has 20 free parameters

Model selection with `lmestSearch()`

- The estimation results for the selected number of states may be displayed as follows:

```
mod3 <- outs$out.single[[3]]
summary(mod3)
#> Call:
#> NULL
#>
#> Coefficients:
#>
#> Initial probabilities:
#>      est_piv
#> [1,]  0.4624
#> [2,]  0.1737
#> [3,]  0.3639
#>
#> Transition probabilities:
#> , , time = 2
#>
#>      state
#> state      1      2      3
#>     1 0.9336 0.0191 0.0473
#>     2 0.0770 0.8761 0.0469
#>     3 0.0992 0.0145 0.8863
#>
#> , , time = 3
#>
#>      state
#> state      1      2      3
#>     1 0.9336 0.0191 0.0473
#>     2 0.0770 0.8761 0.0469
#>     3 0.0992 0.0145 0.8863
```

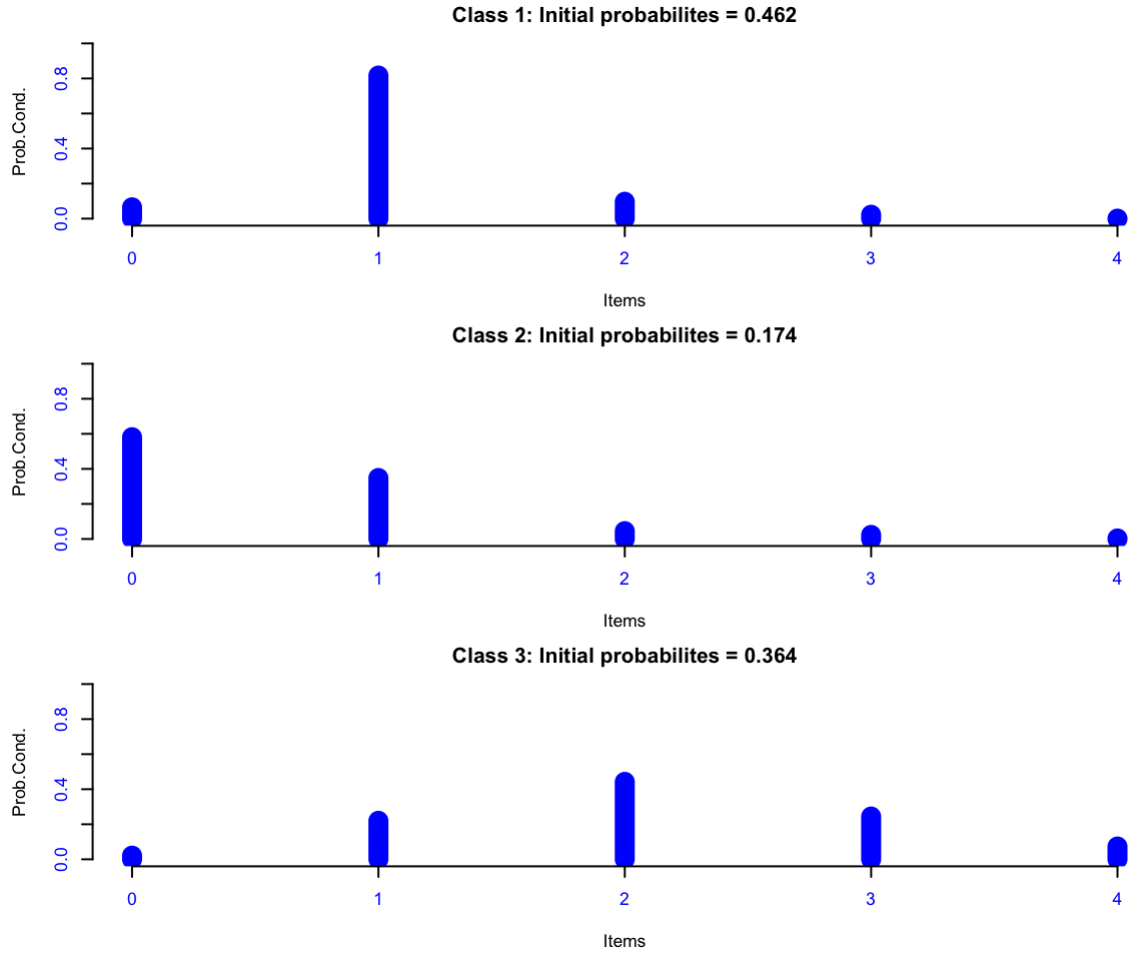
```
#>
#> , , time = 4
#>
#>      state
#> state      1      2      3
#>     1 0.9336 0.0191 0.0473
#>     2 0.0770 0.8761 0.0469
#>     3 0.0992 0.0145 0.8863
#>
#> , , time = 5
#>
#>      state
#> state      1      2      3
#>     1 0.9336 0.0191 0.0473
#>     2 0.0770 0.8761 0.0469
#>     3 0.0992 0.0145 0.8863
#>
#> , , time = 6
#>
#>      state
#> state      1      2      3
#>     1 0.9336 0.0191 0.0473
#>     2 0.0770 0.8761 0.0469
#>     3 0.0992 0.0145 0.8863
#>
#> , , time = 7
#>
#>      state
#> state      1      2      3
#>     1 0.9336 0.0191 0.0473
#>     2 0.0770 0.8761 0.0469
#>     3 0.0992 0.0145 0.8863
#>
#>
#> Conditional response probabilities:
#> , , item = 1
#>
```

```
#>         state
#> category      1      2      3
#>        0 0.0646 0.5809 0.0209
#>        1 0.8165 0.3480 0.2197
#>        2 0.0965 0.0448 0.4420
#>        3 0.0224 0.0233 0.2442
#>        4 0.0000 0.0031 0.0731
```

Model selection with `lmestSearch()`

- A plot of the conditional response probabilities referred to the categories of the univariate response is obtained with:

```
plot(mod3,what="Cond.Prob.")
```


Class 1: Initial probabilites = 0.462
Prob.Cond.
0.0
0.4
0.8
0
1
2
3
4
Items
Class 2: Initial probabilites = 0.174
Prob.Cond.
0.0
0.4
0.8
0
1
2
3
4
Items
Class 3: Initial probabilites = 0.364
Prob.Cond.
0.0
0.4
0.8
0
1
2
3
4
Items

Local and global decoding with `lmestDecoding()`

- Function `lmestDecoding()` allows us to predict the sequence of latent states for the sample units on the basis of the output of the main estimation functions, and so to perform a "dynamic pattern recognition"

- The local and global decoding (Viterbi algorithm) are given by:

```
dec <- lmestDecoding(mod)
```

```
dec <- lmestDecoding(mod)

head(dec$Ul)
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
#> [1,]    2    2    2    2    2    2    2
#> [2,]    3    3    2    1    1    1    2
#> [3,]    2    2    2    2    2    2    2
#> [4,]    1    1    2    2    2    2    2
#> [5,]    2    2    2    2    2    2    2
#> [6,]    2    2    2    2    2    2    2
```

```

head(dec$Ug)
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
#> [1,]    2    2    2    2    2    2    2
```

```
#> [2,]    3    3    2    1    1    1    2
#> [3,]    2    2    2    2    2    2    2
#> [4,]    1    1    2    2    2    2    2
#> [5,]    2    2    2    2    2    2    2
#> [6,]    2    2    2    2    2    2    2
```

Bootstrapping with `bootstrap()`

- Function `bootstrap()` allows us to obtain standard errors by parametric bootstrap. Just for illustrative purposes, we may obtain B=50 bootstrap samples from the output `out1` of the model `lmestCont()` with the `NLSYlong` data as follows:

```
mboot <- bootstrap(mod1, n = 581, B = 50, seed = 172)
```

- Object `seMu` contains the estimated standard errors for the conditional means:

```
round(mboot$seMu,3)
#>      state
#> item     1     2
#>    1 0.036 0.061
#>    2 0.089 0.183
```

Draw samples with `drawLMbasic()`

- Function `drawLMbasic()` allows us to draw sample from the estimated basic LM model

- For example, considering the LM model illustrated with the `RLMSlong` dataset, we can obtain a sample of responses of size n = 100 as follows:

```
draw3 <- drawLMbasic(est = mod3, format = "matrices", seed = 4321, n = 100)
head(draw3$Y)
```

```
head(draw3$Y)
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
#> [1,]    2    0    1    2    1    0    1
#> [2,]    0    0    0    0    1    0    0
#> [3,]    1    2    1    0    1    2    1
#> [4,]    0    1    1    0    0    0    1
#> [5,]    1    2    1    1    1    1    1
#> [6,]    1    1    1    1    2    1    2
```

- Every line of Y contains the sample responses of each unit for the seven time occasions

Citation of the LMest package

```
citation(package = "LMest")
#>
#> To cite LMest in publications use:
#>
#> Bartolucci F, Pandolfi S, Pennoni F (2017). "LMest: An R Package for
#> Latent Markov Models for Longitudinal Categorical Data." _Journal of
#> Statistical Software_, *81*(4), 1-38. doi: 10.18637/jss.v081.i04 (URL:
#> https://doi.org/10.18637/jss.v081.i04).
#>
#> A BibTeX entry for LaTeX users is
#>
#>   @Article{,
#>     title = {{LMest}: An {R} Package for Latent Markov Models for Longitudinal Categorical Data},
#>     author = {Francesco Bartolucci and Silvia Pandolfi and Fulvia Pennoni},
#>     journal = {Journal of Statistical Software},
#>     year = {2017},
#>     volume = {81},
#>     number = {4},
#>     pages = {1--38},
#>     doi = {10.18637/jss.v081.i04},
#>   }
```