Package ‘LMest’

April 19, 2020

Title Generalized Latent Markov Models

Description Latent Markov models for longitudinal continuous and categorical data. See Bartolucci, Pandolfi, Pennoni (2017)<doi:10.18637/jss.v081.i04>.

Version 2.6.1

Date 2020-04-07

Author Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Maintainer Francesco Bartolucci <bart@stat.unipg.it>

Depends R (>= 2.0.0)

Imports MASS, MultiLCIRT, stats, mvtnorm, mmm, Formula(>= 1.2-3), mix, utils, graphics

NeedsCompilation yes

Repository CRAN

LazyLoad yes

Encoding UTF-8

ByteCompile true

License GPL (>= 2)

Date/Publication 2020-04-19 16:40:02 UTC

R topics documented:

`LMest-package` *Overview of the Package LMest*

Description

The package `LMest` is a framework for specifying and fitting Latent (or Hidden) Markov (LM) models, which are tailored for the analysis of longitudinal continuous and categorical data. Covariates are also included in the model specification through suitable parameterizations.

Details

Different LM models are estimated through specific functions requiring a data frame in long format. The functions are the following:

`lmest` Function to estimate LM models for categorical responses generating the following classes:

- `LMbasic-class` for the basic LM model without covariates.
- `LMmanifest-class` for the LM model with covariates in the measurement model.
- `LMlatent-class` for the LM model with covariates in the latent model.

`lmestCont` Function to estimate LM models for continuous outcomes generating the following classes:

- `LMbasiccont-class` for the basic LM model for continuous responses without covariates.
- `LMlatentcont-class` for the LM model for continuous responses with covariates in the latent model.

`lmestMixed` Function to estimate Mixed LM models for categorical responses with discrete random effects in the latent model generating the following class:

- `LMmixed-class` for the mixed LM model.

`lmestMc` Function to estimate Markov Chain models for categorical responses generating the following classes:

- `MCbasic-class` for the MC model without covariates.
- `MCcov-class` for the MC model with covariates.

Maximum likelihood estimation of model parameters is performed through the Expectation-Maximization algorithm, which is implemented by relying on Fortran routines.

Model selection is provided by `lmest` and `lmestCont` functions. In addition, function `lmestSearch` allows us to deal with both model selction 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.

Prediction of the latent states is performed by the function `lmestDecoding`: for local and global decoding (Viterbi algorithm) from the output of functions lmest, lmestCont and lmestMixed.

The package allows us to deal with missing responses, including drop-out and non-monotonic missingness, under the missing-at-random assumption.

Standard errors for the parameter estimates are obtained by exact computation of the information matrix or through reliable numerical approximations of this matrix.

The print method shows some convergence information, and the summary method shows the estimation results.

The package also provides some real and simulated data sets that are listed using the function `data(package = "LMest")`.

Author(s)

Francesco Bartolucci, Alessio Farcomeni, Silvia Pandolfi, Fulvia Pennoni and Alessio Serafini

Maintainer: Francesco Bartolucci <bart@stat.unipg.it>

References

Bartolucci, F., Pandolfi, S. and Pennoni, F. (2017). LMest: An R Package for Latent Markov Models for Longitudinal Categorical Data, *Journal of Statistical Software*, **81**, 1-38, doi:10.18637/jss.v081.i04.

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013). *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Bartolucci, F., Farcomeni, A., and Pennoni, F. (2014). Latent Markov models: A review of a general framework for the analysis of longitudinal data with covariates (with discussion). *TEST*, **23**, 433-465.

See Also

`lmest`, `lmestCont`, `lmestMc`, `lmestMixed`, `LMmixed-class`, `LMbasic-class`,`LMbasiccont-class`, `LMlatent-class`,`LMlatentcont-class`, `LMmanifest-class`

`bootstrap` *Parametric bootstrap*

Description

Function that performs bootstrap parametric resampling to compute standard errors for the parameter estimates.

Usage

```
bootstrap(est, ...)
## S3 method for class 'LMbasic'
bootstrap(est, n = 1000, B = 100, seed = NULL, ...)
## S3 method for class 'LMbasiccont'
bootstrap(est, n = 1000, B=100, seed = NULL, ...)
## S3 method for class 'LMlatent'
bootstrap(est, B = 100, seed = NULL, ...)
## S3 method for class 'LMlatentcont'
bootstrap(est, B = 100, seed = NULL, ...)
```

Arguments

`est`	an object obtained from a call to `lmest` and `lmestCont`
`n`	sample size
`B`	number of bootstrap samples
`seed`	an integer value with the random number generator state
`...`	further arguments

Value

Average of bootstrap estimates and standard errors for the model parameters in `est` object.

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
## Not run:
data("data_SRHS_long")
SRHS <- data_SRHS_long[1:2400,]
SRHS$srhs <- 5 - SRHS$srhs

out1 <- lmest(responsesFormula = srhs ~ NULL,
            index = c("id","t"),
            data = SRHS,
            k = 3,
            tol = 1e-8,
            start = 1,
            modBasic = 1,
            out_se = TRUE,
            seed = 123)

boot1 <- bootstrap(out1)

out2 <- lmest(responsesFormula = srhs ~ NULL,
             latentFormula =  ~
             I(gender - 1) +
             I( 0 + (race == 2) + (race == 3)) +
             I(0 + (education == 4)) +
             I(0 + (education == 5)) +
             I(age - 50) + I((age-50)^2/100),
             index = c("id","t"),
             data = SRHS,
             k = 2,
             paramLatent = "multilogit",
             start = 0)

boot2 <- bootstrap(out2)

require(mmm)
```

```
data(multiLongGaussian)

t <- rep(1:4, times  = max(multiLongGaussian$ID))
multiLongGaussian <- data.frame(t = t, multiLongGaussian)

out3 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                  index = c("ID", "t"),
                  data = multiLongGaussian,
                  k =3,
                  modBasic=1,
                  tol=10^-5)

boot3 <- bootstrap(out3)

out4 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                   latentFormula = ~ X + time,
                   index = c("ID", "t"),
                   data = multiLongGaussian,
                   k = 3,
                   output=TRUE)
boot4 <- bootstrap(out4)

## End(Not run)
```

`bootstrap_lm_basic` *Parametric bootstrap for the basic LM model*

Description

Function that performs bootstrap parametric resampling to compute standard errors for the parameter estimates.

The function is no longer maintained. Please look at `bootstrap` function.

Usage

```
bootstrap_lm_basic(piv, Pi, Psi, n, B = 100, start = 0, mod = 0, tol = 10^-6)
```

Arguments

`piv`	initial probability vector
`Pi`	probability transition matrices (k x k x TT)
`Psi`	matrix of conditional response probabilities (mb x k x r)
`n`	sample size
`B`	number of bootstrap samples
`start`	type of starting values (0 = deterministic, 1 = random)
`mod`	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
`tol`	tolerance level for convergence

Value

mPsi	average of bootstrap estimates of the conditional response probabilities
mpiv	average of bootstrap estimates of the initial probability vector
mPi	average of bootstrap estimates of the transition probability matrices
sePsi	standard errors for the conditional response probabilities
sepiv	standard errors for the initial probability vector
sePi	standard errors for the transition probability matrices

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:
# Example of drug consumption data
# load data
data(data_drug)
data_drug <- as.matrix(data_drug)
S <- data_drug[,1:5]-1
yv <- data_drug[,6]
n <- sum(yv)
# fit of the Basic LM model
k <- 3
out1 <- est_lm_basic(S, yv, k, mod = 1, out_se = TRUE)
out2 <- bootstrap_lm_basic(out1$piv, out1$Pi, out1$Psi, n, mod = 1, B = 1000)

## End(Not run)
```

---

bootstrap_lm_basic_cont *Parametric bootstrap for the basic LM model for continuous outcomes*

Description

Function that performs bootstrap parametric resampling to compute standard errors for the parameter estimates.

The function is no longer maintained. Please look at bootstrap function.

Usage

```
bootstrap_lm_basic_cont(piv, Pi, Mu, Si, n, B = 100, start = 0, mod = 0, tol = 10^-6)
```

Arguments

piv	initial probability vector
Pi	probability transition matrices (k x k x TT)
Mu	matrix of conditional means for the response variables (r x k)
Si	var-cov matrix common to all states (r x r)
n	sample size
B	number of bootstrap samples
start	type of starting values (0 = deterministic, 1 = random)
mod	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
tol	tolerance level for convergence

Value

mMu	average of bootstrap estimates of the conditional means of the response variables
mSi	average of bootstrap estimates of the var-cov matrix
mpiv	average of bootstrap estimates of the initial probability vector
mPi	average of bootstrap estimates of the transition probability matrices
seMu	standard errors for the conditional means of the response variables
seSi	standard errors for the var-cov matrix
sepiv	standard errors for the initial probability vector
sePi	standard errors for the transition probability matrices

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:
# Example based on multivariate longitudinal continuous data

# load data
require(mmm)
data(multiLongGaussian)
res <- long2matrices(multiLongGaussian$ID, X = cbind(multiLongGaussian$X, multiLongGaussian$time),
      Y = cbind(multiLongGaussian$resp1, multiLongGaussian$resp2))
Y <- res$YY
n <- dim(Y)[1]
# fit of the Basic LM model for continuous outcomes
k <- 3
out1 <- est_lm_basic_cont(Y, k, mod = 1)
out2 <- bootstrap_lm_basic_cont(out1$piv, out1$Pi, out1$Mu, out1$Si, n, mod = 1, B = 1000)

## End(Not run)
```

`bootstrap_lm_cov_latent` *Parametric bootstrap for LM models with individual covariates in the latent model*

Description

Function that performs bootstrap parametric resampling to compute standard errors for the parameter estimates.

The function is no longer maintained. Please look at `bootstrap` function.

Usage

```
bootstrap_lm_cov_latent(X1, X2, param = "multilogit", Psi, Be, Ga, B = 100,
                        fort = TRUE)
```

Arguments

`X1`	matrix of covariates affecting the initial probabilities (n x nc1)
`X2`	array of covariates affecting the transition probabilities (n x TT-1 x nc2)
`param`	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
`Psi`	array of conditional response probabilities (mb x k x r)
`Be`	parameters affecting the logit for the initial probabilities
`Ga`	parametes affecting the logit for the transition probabilities
`B`	number of bootstrap samples
`fort`	to use fortran routine when possible (FALSE for not use fortran)

Value

`mPsi`	average of bootstrap estimates of the conditional response probabilities
`mBe`	average of bootstrap estimates of the parameters affecting the logit for the initial probabilities
`mGa`	average of bootstrap estimates of the parameters affecting the logit for the transition probabilities
`sePsi`	standard errors for the conditional response probabilities
`seBe`	standard errors for the parameters in Be
`seGa`	standard errors for the parameters in Ga

Author(s)

Francesco Bartolucci, Silvia Pandolfi - University of Perugia (IT)

Examples

```
## Not run:
# Example based on self-rated health status (SRHS) data
# load SRHS data
data(data_SRHS_long)
dataSRHS <- data_SRHS_long

TT <- 8
head(dataSRHS)
res <- long2matrices(dataSRHS$id, X = cbind(dataSRHS$gender-1,
dataSRHS$race == 2 | dataSRHS$race == 3, dataSRHS$education == 4,
dataSRHS$education == 5, dataSRHS$age-50, (dataSRHS$age-50)^2/100),
Y = dataSRHS$srhs)

# matrix of responses (with ordered categories from 0 to 4)
S <- 5-res$YY

# matrix of covariates (for the first and the following occasions)
# colums are: gender,race,educational level (2 columns),age,age^2)
X1 <- res$XX[,1,]
X2 <- res$XX[,2:TT,]

# estimate the model
out1 <- est_lm_cov_latent(S, X1, X2, k = 2, output = TRUE, out_se = TRUE)

out2 <- bootstrap_lm_cov_latent(X1, X2, Psi = out1$Psi, Be = out1$Be, Ga = out1$Ga, B = 1000)

## End(Not run)
```

---

bootstrap_lm_cov_latent_cont *Parametric bootstrap for LM models for continuous outcomes with individual covariates in the latent model*

Description

Function that performs bootstrap parametric resampling to compute standard errors for the parameter estimates.

The function is no longer maintained. Please look at bootstrap **function**.

Usage

```
bootstrap_lm_cov_latent_cont(X1, X2, param = "multilogit", Mu, Si, Be, Ga, B = 100)
```

Arguments

X1	matrix of covariates affecting the initial probabilities (n x nc1)
X2	array of covariates affecting the transition probabilities (n x TT-1 x nc2)

param	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
Mu	matrix of conditional means for the response variables (r x k)
Si	var-cov matrix common to all states (r x r)
Be	parameters affecting the logit for the initial probabilities
Ga	parametes affecting the logit for the transition probabilities
B	number of bootstrap samples

Value

mMu	average of bootstrap estimates of the conditional means for the response variables
mSi	average of bootstrap estimates of the var-cov matrix
mBe	average of bootstrap estimates of the parameters affecting the logit for the initial probabilities
mGa	average of bootstrap estimates of the parameters affecting the logit for the transition probabilities
seMu	standard errors for the conditional means
seSi	standard errors for the var-cov matrix
seBe	standard errors for the parameters in Be
seGa	standard errors for the parameters in Ga

Author(s)

Francesco Bartolucci, Silvia Pandolfi - University of Perugia (IT)

Examples

```
## Not run:
# Example based on multivariate longitudinal continuous data

# load data
require(mmm)
data(multiLongGaussian)
TT <- 4
res <- long2matrices(multiLongGaussian$ID, X = cbind(multiLongGaussian$X, multiLongGaussian$time),
                    Y = cbind(multiLongGaussian$resp1, multiLongGaussian$resp2))
Y <- res$YY
X1 <- res$XX[,1,]
X2 <- res$XX[,2:TT,]

# estimate the model
est <- est_lm_cov_latent_cont(Y, X1, X2, k = 3, output = TRUE)
out <- bootstrap_lm_cov_latent_cont(X1, X2, Mu = est$Mu, Si = est$Si,
                                    Be = est$Be, Ga = est$Ga, B = 1000)
```

```
## End(Not run)
```

`data_criminal_sim` *Criminal dataset*

Description

Simulated dataset about crimes committed by a cohort of subjects.

Usage

```
data(data_criminal_sim)
```

Format

A data frame with 60000 observations on the following 13 variables.

`id` subject id

`sex` gender of the subject

`time` occasion of observation

`y1` crime of type 1 (violence against the person)

`y2` crime of type 2 (sexual offences)

`y3` crime of type 3 (burglary)

`y4` crime of type 4 (robbery)

`y5` crime of type 5 (theft and handling stolen goods)

`y6` crime of type 6 (fraud and forgery)

`y7` crime of type 7 (criminal demage)

`y8` crime of type 8 (drug offences)

`y9` crime of type 9 (motoring offences)

`y10` crime of type 10 (other offences)

References

Bartolucci, F., Pennoni, F. and Francis, B. (2007), A latent Markov model for detecting patterns of criminal activity, *Journal of the Royal Statistical Society, series A*, **170**, pp. 115-132.

Examples

```
data(data_criminal_sim)
```

data_drug — *Dataset about marijuana consumption*

Description

Longitudinal dataset about marijuana consumption measured by ordinal variables with 3 categories with increasing levels of consumption (1 "never in the past year", 2 "no more than once in a month in the past year", 3 "more than once a month in the past year").

Usage

```
data(data_drug)
```

Format

A data frame with 51 observations on the following 6 variables.

- `V1` reported drug use at the 1st occasion
- `V2` reported drug use at the 2nd occasion
- `V3` reported drug use at the 3rd occasion
- `V4` reported drug use at the 4th occasion
- `V5` reported drug use at the 5th occasion
- `V6` frequency of the response configuration

Source

Elliot, D. S., Huizinga, D. and Menard, S. (1989) *Multiple Problem Youth: Delinquency, Substance Use, and Mental Health Problems.* New York: Springer.

References

Bartolucci, F. (2006) Likelihood inference for a class of latent markov models under linear hypotheses on the transition probabilities. *Journal of the Royal Statistical Society, series B*, **68**, 155-178.

Examples

```
data(data_drug)
```

`data_SRHS_long` *Self-reported health status dataset*

Description

Dataset about self-reported health status derived from the Health and Retirement Study conducted by the University of Michigan.

Usage

```
data(data_SRHS_long)
```

Format

A data frame with 56592 observations on the following 6 variables.

`t` occasion of observation

`id` subject id

`gender` sex of the subject coded as 1 for "male", 2 for "female"

`race` race coded as 1 for "white", 2 for "black", 3 for "others"

`education` educational level coded as 1 for "high school", 2 for "general educational diploma", 3 for "high school graduate", 4 for "some college", 5 for "college and above"

`age` age at the different time occasions

`srhs` self-reported health status at the different time occasions coded as 1 for "excellent", 2 for "very good", 3 for "good", 4 for "fair", 5 for "poor"

References

Bartolucci, F., Bacci, S. and Pennoni, F. (2014) Longitudinal analysis of the self-reported health status by mixture latent autoregressive models, *Journal of the Royal Statistical Society - series C*, **63**, pp. 267-288

Examples

```
data(data_SRHS_long)
```

decoding *Perform local and global decoding*

Description

Function that performs local and global decoding (Viterbi) from the output of est_lm_basic, est_lm_cov_latent, est_lm_cov_manifest, and est_lm_mixed.

The function is no longer maintained. Please look at lmestDecoding function

Usage

```
decoding(est, Y, X1 = NULL, X2 = NULL, fort = TRUE)
```

Arguments

est	output from est_lm_basic, est_lm_cov_latent, est_lm_cov_manifest, or est_lm_mixed
Y	single vector or matrix of responses
X1	matrix of covariates on the initial probabilities (est_lm_cov_latent) or on the responses (est_lm_cov_manifest)
X2	array of covariates on the transition probabilites
fort	to use Fortran routines

Value

Ul	matrix of local decoded states corresponding to each row of Y
Ug	matrix of global decoded states corresponding to each row of Y

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

References

Viterbi A. (1967) Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, **13**, 260-269.

Juan B., Rabiner L. (1991) Hidden Markov Models for Speech Recognition. *Technometrics*, **33**, 251-272.

Examples

```
## Not run:
# example for the output from est_lm_basic
data(data_drug)
data_drug <- as.matrix(data_drug)
S <- data_drug[,1:5]-1
yv <- data_drug[,6]
n <- sum(yv)
# fit the Basic LM model
k <- 3
est <- est_lm_basic(S, yv, k, mod = 1)
# decoding for a single sequence
out1 <- decoding(est, S[1,])
# decoding for all sequences
out2 <- decoding(est, S)


# example for the output from est_lm_cov_latent with difflogit parametrization
data(data_SRHS_long)
dataSRHS <- data_SRHS_long[1:1600,]

TT <- 8
head(dataSRHS)
res <- long2matrices(dataSRHS$id, X = cbind(dataSRHS$gender-1,
dataSRHS$race == 2 | dataSRHS$race == 3, dataSRHS$education == 4,
dataSRHS$education == 5, dataSRHS$age-50,(dataSRHS$age-50)^2/100),
Y= dataSRHS$srhs)

# matrix of responses (with ordered categories from 0 to 4)
S <- 5-res$YY

# matrix of covariates (for the first and the following occasions)
# colums are: gender,race,educational level (2 columns),age,age^2)
X1 <- res$XX[,1,]
X2 <- res$XX[,2:TT,]

# estimate the model
est <- est_lm_cov_latent(S, X1, X2, k = 2, output = TRUE, param = "difflogit")
# decoding for a single sequence
out1 <- decoding(est, S[1,,], X1[1,], X2[1,,])
# decoding for all sequences
out2 <- decoding(est, S, X1, X2)

## End(Not run)
```

`drawLMbasic` *Draw samples from the basic LM model*

Description

Function that draws samples from the basic LM model.

Usage

```
drawLMbasic(piv, Pi, Psi, n, est = NULL, format = c("long","matrices"), seed = NULL)
```

Arguments

piv	vector of initial probabilities of the latent Markov chain
Pi	set of transition probabilities matrices (k x k x TT)
Psi	array of conditional response probabitlies (mb x k x r)
n	sample size
est	object of class LMbasic (LMbasic-class)
format	character string indicating the format of final responses matrix
seed	an integer value with the random number generator state

Value

Y	matrix of response configurations unit by unit
S	matrix of distinct response configurations
yv	corresponding vector of frequencies
piv	vector of initial probabilities of the latent Markov chain
Pi	set of transition probabilities matrices (k x k x TT)
Psi	array of conditional response probabitlies (mb x k x r)
n	sample size
est	object of class LMbasic (LMbasic-class)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
## Not run:

# draw a sample for 1000 units and only one response variable
n <- 1000
TT <- 6
k <- 2
r <- 1 #number of response variables
mb <- 3 #maximum number of response categories

piv <- c(0.7,0.3)
Pi <- matrix(c(0.9,0.1,0.1,0.9), k, k)
Pi <- array(Pi, c(k, k, TT))
Pi[,,1] <- 0
Psi <- matrix(c(0.7,0.2,0.1,0.5,0.4,0.1), mb, k)
Psi <- array(Psi, c(mb, k, r))
```

```
out <- drawLMbasic(piv, Pi, Psi, n = 1000)


data("data_SRHS_long")
SRHS <- data_SRHS_long[1:2400,]

SRHS$srhs <- 5 - SRHS$srhs

est <- lmest(responsesFormula = srhs ~ NULL,
             index = c("id","t"),
             data = SRHS,
             k = 3)

out1 <- drawLMbasic(est = est, format = "matrices", seed = 4321, n = 100)

## End(Not run)
```

`drawLMbasiccont` *Draw samples from the basic LM model for continuous outcomes*

Description

Function that draws samples from the basic LM model for continuous outcomes with specific parameters.

Usage

```
drawLMbasiccont(piv, Pi, Mu, Si, n, est = NULL,
                format = c("long","matrices"), seed = NULL)
```

Arguments

`piv`	vector of initial probabilities of the latent Markov chain
`Pi`	set of transition probabilities matrices (k x k x TT)
`Mu`	matrix of conditional means for the response variables (r x k)
`Si`	var-cov matrix common to all states (r x r)
`n`	sample size
`est`	object of class `LMbasiccont` (`LMbasiccont-class`)
`format`	character string indicating the format of final responses matrix
`seed`	an integer value with the random number generator state

Value

Y	array of continuous outcomes (n x TT x r)
piv	vector of initial probabilities of the latent Markov chain
Pi	set of transition probabilities matrices (k x k x TT)
Mu	matrix of conditional means for the response variables (r x k)
Si	var-cov matrix common to all states (r x r)
n	sample size
est	object of class LMbasiccont (LMbasiccont-class)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
## Not run:
# draw a sample for 1000 units and 3 response variable
n <- 1000
TT <- 5
k <- 2
r <- 3 #number of response variables

piv <- c(0.7,0.3)
Pi <- matrix(c(0.9,0.1,0.1,0.9), k, k)
Pi <- array(Pi, c(k, k, TT))
Pi[,,1] <- 0
Mu <- matrix(c(-2,-2,0,0,2,2), r, k)
Si <- diag(r)
out <- drawLMbasiccont(piv, Pi, Mu, Si, n)


require(mmm)
data(multiLongGaussian)

t <- rep(1:4, times = max(multiLongGaussian$ID))
multiLongGaussian <- data.frame(t = t, multiLongGaussian)

est <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                 index = c("ID", "t"),
                 data = multiLongGaussian,
                 k = 3,
                 modBasic = 1,
                 tol = 10^-5)

out2 <- drawLMbasiccont(est = est, n = 100, format = "long", seed = 4321)


## End(Not run)
```

drawLMlatent *Draw samples from LM model with covariates in the latent model*

Description

Function that draws samples from the LM model with individual covariates with specific parameters.

Usage

```
drawLMlatent(Psi, Be, Ga, latentFormula, data, index,
             paramLatent = c("multilogit","difflogit"), est = NULL,
             format = c("long","matrices"), fort = TRUE, seed = NULL)
```

Arguments

Psi	array of conditional response probabilities (mb x k x r)
Be	parameters affecting the logit for the initial probabilities
Ga	parametes affecting the logit for the transition probabilities
latentFormula	a symbolic description of the model to be fitted. Detailed description is given in lmest
data	a data frame in long format, with rows corresponding to observations and columns corresponding to variables, a column corresponding to time occasions and a column containing the unit identifier
index	a character vector with two elements indicating the name of the "id" column as first element and the "time" column as second element
paramLatent	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
est	object of class LMlatent (LMlatent-class)
format	character string indicating the format of final responses matrix
fort	to use fortran routine when possible (FALSE for not use fortran)
seed	an integer value with the random number generator state

Value

Y	matrix of response configurations
U	matrix containing the sequence of latent states (n x TT)
Psi	array of conditional response probabilities (mb x k x r)
Be	parameters affecting the logit for the initial probabilities
Ga	parametes affecting the logit for the transition probabilities
latentFormula	a symbolic description of the model to be fitted. Detailed description is given in lmest

`data`	a data frame in long format, with rows corresponding to observations and columns corresponding to variables, a column corresponding to time occasions and a column containing the unit identifier
`est`	object of class `LMlatent` (`LMlatent-class`)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
## Not run:
data(data_SRHS_long)
dataSRHS <- data_SRHS_long
data_SRHS_long$srhs <- 5 - data_SRHS_long$srhs
est  <- lmest(responsesFormula = srhs ~ NULL,
              latentFormula =  ~
              I(gender - 1) +
              I( 0 + (race == 2) + (race == 3)) +
              I(0 + (education == 4)) +
              I(0 + (education == 5)) +
              I(age - 50) + I((age-50)^2/100),
              index = c("id","t"),
              data = data_SRHS_long,
              k = 2,
              paramLatent = "multilogit",
              start = 0)

out <- drawLMlatent(est = est,format = "matrices",seed = 4321)

out1 <- drawLMlatent(Psi = est$Psi,Be = est$Be,Ga = est$Ga,
                     data = data_SRHS_long, index = c("id","t"),
                     latentFormula =  ~
                     I(gender - 1) +
                     I( 0 + (race == 2) + (race == 3)) +
                     I(0 + (education == 4)) +
                     I(0 + (education == 5)) +
                     I(age - 50) + I((age-50)^2/100),
                     paramLatent = "multilogit", format = "matrice",
                     seed = 4321)


## End(Not run)
```

`drawLMlatentcont` *Draw samples from LM model for continuous outcomes with covariates in the latent model*

Description

Function that draws samples from the LM model for continuous outcomes with individual covariates with specific parameters.

Usage

```
drawLMlatentcont(Mu, Si, Be, Ga, latentFormula, data, index,
                 paramLatent = c("multilogit","difflogit"), est = NULL,
                 format = c("long","matrices"), fort = TRUE, seed = NULL)
```

Arguments

`Mu`	array of conditional means for the response variables (r x k)
`Si`	var-cov matrix common to all states (r x r)
`Be`	parameters affecting the logit for the initial probabilities
`Ga`	parametes affecting the logit for the transition probabilities
`latentFormula`	a symbolic description of the model to be fitted. A detailed description is given in `lmestCont`
`data`	a data frame in long format, with rows corresponding to observations and columns corresponding to variables, a column corresponding to time occasions and a column containing the unit identifier
`index`	a character vector with two elements indicating the name of the "id" column as first element and the "time" column as second element
`paramLatent`	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
`est`	object of class `LMlatentcont` (`LMlatentcont-class`)
`format`	character string indicating the format of final responses matrix
`fort`	to use fortran routine when possible (FALSE for not use fortran)
`seed`	an integer value with the random number generator state

Value

`Y`	array of continuous outcomes (n x TT x r)
`U`	matrix containing the sequence of latent states (n x TT)
`Mu`	array of conditional means for the response variables (r x k)
`Si`	var-cov matrix common to all states (r x r)
`Be`	parameters affecting the logit for the initial probabilities
`Ga`	parametes affecting the logit for the transition probabilities
`latentFormula`	a symbolic description of the model to be fitted. A detailed description is given in `lmestCont`
`data`	a data frame in long format, with rows corresponding to observations and columns corresponding to variables, a column corresponding to time occasions and a column containing the unit identifier
`est`	object of class `LMlatentcont` (`LMlatentcont-class`)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
## Not run:
require(mmm)
data(multiLongGaussian)

t <- rep(1:4, times  = max(multiLongGaussian$ID))
multiLongGaussian <- data.frame(t = t, multiLongGaussian)

est <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                 latentFormula = ~ X + time,
                 index = c("ID", "t"),
                 data = multiLongGaussian,
                 k = 3)

out <- drawLMlatentcont(est = est,format = "matrices", seed = 4321)
out1 <- drawLMlatentcont(latentFormula = ~ X + time,data = multiLongGaussian,
                         index = c("ID", "t"),
                         Mu = est$Mu,Si = est$Si,
                     Be = est$Be,Ga = est$Ga, fort=TRUE, seed = 4321, format = "matrices")


## End(Not run)
```

---

drawLMmixed *Draws samples from the mixed LM model*

Description

Function that draws samples from the mixed LM model with specific parameters.

Usage

```
drawLMmixed(la, Piv, Pi, Psi, n, TT, est = NULL,
             format = c("long","matrices"), seed = NULL)
```

Arguments

la	vector of mass probabilities for the first latent variable
Piv	matrix of initial probabilities of the latent Markov chain (k2 x k1)
Pi	set of transition matrices (k2 x k2 x k1)
Psi	array of conditional response probabitlies (mb x k2 x r)
n	sample size

TT	number of time occasions
est	object of class LMmixed (LMmixed-class)
format	character string indicating the format of final responses matrix
seed	an integer value with the random number generator state

Value

Y	matrix of response configurations unit by unit
S	matrix of distinct response configurations
yv	corresponding vector of frequencies
la	vector of mass probabilities for the first latent variable
Piv	matrix of initial probabilities of the latent Markov chain (k2 x k1)
Pi	set of transition matrices (k2 x k2 x k1)
Psi	array of conditional response probabitlies (mb x k2 x r)
n	sample size
TT	number of time occasions
est	object of class LMmixed (LMmixed-class)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
# draw a sample for 1000 units and only one response variable and 5 time occasions
k1 <- 2
k2 <- 3
la <- rep(1/k1, k1)
Piv <- matrix(1/k2, k2, k1)
Pi <- array(0, c(k2, k2, k1))
Pi[,,1] <- diag(k2)
Pi[,,2] <- 1/k2
Psi <- cbind(c(0.6,0.3,0.1), c(0.1,0.3,0.6), c(0.3,0.6,0.1))
out <- drawLMmixed(la, Piv, Pi, Psi, n = 1000, TT = 5)


## Not run:
# Example based on criminal data

data(data_criminal_sim)
data_criminal_sim = data.frame(data_criminal_sim)
# Estimate mixed LM model for females

responsesFormula <- lmestFormula(data = data_criminal_sim,
                                  response = "y")$responsesFormula
est <- lmestMixed(responsesFormula = responsesFormula,
                  index = c("id","time"),
```

```
                      k1 = 2,
                      k2 = 2,
                      data = data_criminal_sim[data_criminal_sim$sex == 2,])

out <- drawLMmixed(est = est, n = 100,  seed = 4321)


## End(Not run)
```

---

draw_lm_basic *Draw samples from the basic LM model*

Description

Function that draws samples from the basic LM model with specific parameters.

The function is no longer maintained. Please look at `drawLMbasic` **function.**

Usage

```
draw_lm_basic(piv, Pi, Psi, n)
```

Arguments

`piv`	vector of initial probabilities of the latent Markov chain
`Pi`	set of transition probabilities matrices (k x k x TT)
`Psi`	array of conditional response probabitlies (mb x k x r)
`n`	sample size

Value

`Y`	matrix of response configurations unit by unit
`S`	matrix of distinct response configurations
`yv`	corresponding vector of frequencies

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:
# draw a sample for 1000 units and only one response variable
n <- 1000
TT <- 6
k <- 2
r <- 1 #number of response variables
```

```
mb <- 3 #maximum number of response categories

piv <- c(0.7, 0.3)
Pi <- matrix(c(0.9,0.1,0.1,0.9), k, k)
Pi <- array(Pi, c(k, k, TT))
Pi[,,1] <- 0
Psi <- matrix(c(0.7,0.2,0.1,0.5,0.4,0.1), mb, k)
Psi <- array(Psi, c(mb, k, r))
out <- draw_lm_basic(piv, Pi, Psi, n = 1000)

## End(Not run)
```

`draw_lm_basic_cont` *Draw samples from the basic LM model for continuous outcomes*

Description

Function that draws samples from the basic LM model for continuous outcomes with specific parameters.

The function is no longer maintained. Please look at `drawLMbasiccont` **function**.

Usage

```
draw_lm_basic_cont(piv, Pi, Mu, Si, n)
```

Arguments

`piv`	vector of initial probabilities of the latent Markov chain
`Pi`	set of transition probabilities matrices (k x k x TT)
`Mu`	matrix of conditional means for the response variables (r x k)
`Si`	var-cov matrix common to all states (r x r)
`n`	sample size

Value

`Y`	array of continuous outcomes (n x TT x r)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:

# draw a sample for 1000 units and 3 response variable
n <- 1000
TT <- 5
k <- 2
r <- 3 #number of response variables

piv <- c(0.7,0.3)
Pi <- matrix(c(0.9,0.1,0.1,0.9), k, k)
Pi <- array(Pi, c(k, k, TT))
Pi[,,1] <- 0
Mu <- matrix(c(-2,-2,0,0,2,2), r, k)
Si <- diag(r)
out <- draw_lm_basic_cont(piv, Pi, Mu, Si, n)

## End(Not run)
```

---

draw_lm_cov_latent *Draw samples from LM model with covariaates in the latent model*

Description

Function that draws samples from the LM model with individual covariates with specific parameters.

The function is no longer maintained. Please look at `drawLMlatent` function.

Usage

```
draw_lm_cov_latent(X1, X2, param = "multilogit", Psi, Be, Ga, fort = TRUE)
```

Arguments

`X1`	desing matrix for the covariates on the initial probabilities (n x nc1)
`X2`	desing matrix for the covariates on the transition probabilities (n x TT-1 x nc2)
`param`	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
`Psi`	array of conditional response probabilities (mb x k x r)
`Be`	parameters affecting the logit for the initial probabilities
`Ga`	parametes affecting the logit for the transition probabilities
`fort`	to use fortran routine when possible (FALSE for not use fortran)

Value

Y	matrix of response configurations unit by unit (n x TT x r)
U	matrix containing the sequence of latent states (n x TT)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:

# draw a sample for 1000 units, 10 response variable and 2 covariates
n <- 1000
TT <- 5
k <- 2
nc <- 2 #number of covariates
r <- 10 #number of response variables
mb <- 2 #maximum number of response categories
fort <- TRUE


Psi <- matrix(c(0.9,0.1,0.1,0.9), mb, k)
Psi <- array(Psi, c(mb, k, r))
Ga <- matrix(c(-log(0.9/0.1),0.5,1), (nc+1)*(k-1), k)
Be <- array(c(0,0.5,1), (nc+1)*(k-1))
#Simulate covariates
X1 <- matrix(0, n, nc)
for(j in 1:nc) X1[,j] <- rnorm(n)
X2 <- array(0,c(n, TT-1, nc))
for (t in 1:(TT-1)) for(j in 1:nc){
if(t==1){
X2[,t,j] <- 0.5*X1[,j] + rnorm(n)
}else{
X2[,t,j] <- 0.5 *X2[,t-1,j] + rnorm(n)
}
}

out <- draw_lm_cov_latent(X1, X2, Psi = Psi, Be = Be, Ga = Ga, fort = fort)

## End(Not run)
```

---

draw_lm_cov_latent_cont

Draw samples from LM model for continuous outcomes with covariaates in the latent model

---

Description

Function that draws samples from the LM model for continuous outcomes with individual covariates with specific parameters.

The function is no longer maintained. Please look at `drawLMlatentcont` **function**.

Usage

```
draw_lm_cov_latent_cont(X1, X2, param = "multilogit", Mu, Si, Be, Ga, fort = TRUE)
```

Arguments

X1	desing matrix for the covariates on the initial probabilities (n x nc1)
X2	desing matrix for the covariates on the transition probabilities (n x TT-1 x nc2)
param	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
Mu	array of conditional means for the response variables (r x k)
Si	var-cov matrix common to all states (r x r)
Be	parameters affecting the logit for the initial probabilities
Ga	parametes affecting the logit for the transition probabilities
fort	to use fortran routine when possible (FALSE for not use fortran)

Value

Y	array of continuous outcomes (n x TT x r)
U	matrix containing the sequence of latent states (n x TT)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:
# draw a sample for 1000 units, 10 response variable and 2 covariates
n <- 1000
TT <- 5
k <- 2
nc <- 2 #number of covariates
r <- 3 #number of response variables
fort <- TRUE

Mu <- matrix(c(-2,-2,0,0,2,2), r, k)
Si <- diag(r)
Ga <- matrix(c(-log(0.9/0.1),0.5,1), (nc+1)*(k-1), k)
Be <- array(c(0,0.5,1), (nc+1)*(k-1))
```

```
#Simulate covariates
X1 <- matrix(0, n, nc)
for(j in 1:nc) X1[,j] <- rnorm(n)
X2 <- array(0, c(n,TT-1,nc))
for (t in 1:(TT-1)) for(j in 1:nc){
if(t==1){
X2[,t,j] <- 0.5*X1[,j] + rnorm(n)
}else{
X2[,t,j] <- 0.5*X2[,t-1,j] + rnorm(n)
}
}

out <- draw_lm_cov_latent_cont(X1, X2, param = "multilogit", Mu, Si, Be, Ga, fort = fort)

## End(Not run)
```

---

draw_lm_mixed *Draws samples from the mixed LM model*

Description

Function that draws samples from the mixed LM model with specific parameters.

The function is no longer maintained. Please look at drawLMmixed **function**.

Usage

```
draw_lm_mixed(la, Piv, Pi, Psi, n, TT)
```

Arguments

la	vector of mass probabilities for the first latent variable
Piv	matrix of initial probabilities of the latent Markov chain (k2 x k1)
Pi	set of transition matrices (k2 x k2 x k1)
Psi	array of conditional response probabitlies (mb x k2 x r)
n	sample size
TT	number of time occasions

Value

Y	matrix of response configurations unit by unit
S	matrix of distinct response configurations
yv	corresponding vector of frequencies

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:
# draw a sample for 1000 units and only one response variable and 5 time occasions
k1 <- 2
k2 <- 3
la <- rep(1/k1,k1)
Piv <- matrix(1/k2,k2,k1)
Pi <- array(0,c(k2,k2,k1))
Pi[,,1] <- diag(k2)
Pi[,,2] <- 1/k2
Psi <- cbind(c(0.6,0.3,0.1),c(0.1,0.3,0.6),c(0.3,0.6,0.1))
out <- draw_lm_mixed(la,Piv,Pi,Psi,n=1000,TT=5)

## End(Not run)
```

`est_lm_basic` *Estimate basic LM model*

Description

Main function for estimating the basic LM model.

The function is no longer maintained. Please look at `lmest` function.

Usage

```
est_lm_basic(S, yv, k, start = 0, mod = 0, tol = 10^-8, maxit = 1000,
                    out_se = FALSE, piv = NULL, Pi = NULL, Psi = NULL)
```

Arguments

`S`	array of available configurations (n x TT x r) with categories starting from 0 (use NA for missing responses)
`yv`	vector of frequencies of the available configurations
`k`	number of latent states
`start`	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
`mod`	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
`tol`	tolerance level for convergence
`maxit`	maximum number of iterations of the algorithm
`out_se`	to compute the information matrix and standard errors
`piv`	initial value of the initial probability vector (if start=2)
`Pi`	initial value of the transition probability matrices (k x k x TT) (if start=2)
`Psi`	initial value of the conditional response probabilities (mb x k x r) (if start=2)

Value

lk	maximum log-likelihood
piv	estimate of initial probability vector
Pi	estimate of transition probability matrices
Psi	estimate of conditional response probabilities
np	number of free parameters
aic	value of AIC for model selection
bic	value of BIC for model selection
lkv	log-likelihood trace at every step
V	array containing the posterior distribution of the latent states for each response configuration and time occasion
sepiv	standard errors for the initial probabilities
sePi	standard errors for the transition probabilities
sePsi	standard errors for the conditional response probabilities
call	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:
# Example of drug consumption data

# load data
data(data_drug)
data_drug <- as.matrix(data_drug)
S <- data_drug[,1:5]-1
yv <- data_drug[,6]

# fit of the Basic LM model
k <- 3
out <- est_lm_basic(S, yv, k, mod = 1)
summary(out)

# Example based on criminal data

# load criminal data
data(data_criminal_sim)
out <- long2wide(data_criminal_sim, "id" , "time" , "sex",
```

```
c("y1","y2","y3","y4","y5","y6","y7","y8","y9","y10"),aggr = T, full = 999)
XX <- out$XX
YY <- out$YY
freq <- out$freq

# fit basic LM model with increasing number of states to select the most suitable
Res0 <- vector("list", 7)
for(k in 1:7){
    Res0[[k]] <- est_lm_basic(YY, freq, k, mod = 1, tol = 10^-4)
    save(list <- ls(), file = "example_criminal_temp.RData")
}
out1 <- Res0[[6]]

## End(Not run)
```

---

est_lm_basic_cont *Estimate basic LM model for continuous outcomes*

Description

Main function for estimating the basic LM model for continuous outcomes.

The function is no longer maintained. Please look at `lmestCont` **function**.

Usage

```
est_lm_basic_cont(Y, k, start = 0, mod = 0, tol = 10^-8, maxit = 1000,
                  out_se = FALSE, piv = NULL, Pi = NULL, Mu = NULL, Si = NULL)
```

Arguments

`Y`	array of continuous outcomes (n x TT x r)
`k`	number of latent states
`start`	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
`mod`	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
`tol`	tolerance level for convergence
`maxit`	maximum number of iterations of the algorithm
`out_se`	to compute the information matrix and standard errors
`piv`	initial value of the initial probability vector (if start=2)
`Pi`	initial value of the transition probability matrices (k x k x TT) (if start=2)
`Mu`	initial value of the conditional means (r x k) (if start=2)
`Si`	initial value of the var-cov matrix common to all states (r x r) (if start=2)

Value

lk	maximum log-likelihood
piv	estimate of initial probability vector
Pi	estimate of transition probability matrices
Mu	estimate of conditional means of the response variables
Si	estimate of var-cov matrix common to all states
np	number of free parameters
aic	value of AIC for model selection
bic	value of BIC for model selection
lkv	log-likelihood trace at every step
V	array containing the posterior distribution of the latent states for each units and time occasion
call	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:
# Example based on multivariate longitudinal continuous data

# load data
require(mmm)
data(multiLongGaussian)
res <- long2matrices(multiLongGaussian$ID,X=cbind(multiLongGaussian$X,multiLongGaussian$time),
      Y=cbind(multiLongGaussian$resp1, multiLongGaussian$resp2))
Y <- res$YY

# fit of the Basic LM model for continuous outcomes
k <- 3
out <- est_lm_basic_cont(Y, k, mod = 1, tol = 10^-5)
summary(out)

## End(Not run)
```

est_lm_cov_latent *Estimate LM model with covariates in the latent model*

Description

Main function for estimating the LM model with covariates in the latent model.

The function is no longer maintained. Please look at `lmest` **function**.

Usage

```
est_lm_cov_latent(S, X1=NULL, X2=NULL, yv = rep(1,nrow(S)), k, start = 0, tol = 10^-8,
                  maxit = 1000, param = "multilogit", Psi, Be, Ga, fort = TRUE,
                  output = FALSE, out_se = FALSE, fixPsi = FALSE)
```

Arguments

S	array of available configurations (n x TT x r) with categories starting from 0 (use NA for missing responses)
X1	matrix of covariates affecting the initial probabilities (n x nc1)
X2	array of covariates affecting the transition probabilities (n x TT-1 x nc2)
yv	vector of frequencies of the available configurations
k	number of latent states
start	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
tol	tolerance level for checking convergence of the algorithm
maxit	maximum number of iterations of the algorithm
param	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
Psi	intial value of the array of the conditional response probabilities (mb x k x r)
Be	intial value of the parameters affecting the logit for the initial probabilities (if start=2)
Ga	intial value of the parametes affecting the logit for the transition probabilities (if start=2)
fort	to use fortran routine when possible (FALSE for not use fortran)
output	to return additional output (V,PI,Piv,Ul)
out_se	to compute the information matrix and standard errors
fixPsi	TRUE if Psi is given in input and is not updated anymore

Value

lk	maximum log-likelihood
Be	estimated array of the parameters affecting the logit for the initial probabilities
Ga	estimated array of the parameters affecting the logit for the transition probabilities
Piv	estimate of initial probability matrix
PI	estimate of transition probability matrices
Psi	estimate of conditional response probabilities
np	number of free parameters
aic	value of AIC for model selection
bic	value of BIC for model selection
lkv	log-likelihood trace at every step
V	array containing the posterior distribution of the latent states for each response configuration and time occasion
Ul	matrix containing the predicted sequence of latent states by the local decoding method
sePsi	standard errors for the conditional response matrix
seBe	standard errors for Be
seGa	standard errors for Ga
call	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia, http://www.stat.unipg.it/bartolucci

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:
# Example based on self-rated health status (SRHS) data
# load SRHS data
data(data_SRHS_long)
dataSRHS = data_SRHS_long

TT <- 8
head(dataSRHS)
res <- long2matrices(dataSRHS$id, X = cbind(dataSRHS$gender-1,
dataSRHS$race == 2 | dataSRHS$race == 3, dataSRHS$education == 4,
dataSRHS$education == 5, dataSRHS$age-50, (dataSRHS$age-50)^2/100),
Y = dataSRHS$srhs)
```

```
# matrix of responses (with ordered categories from 0 to 4)
S <- 5-res$YY
n <- dim(S)[1]

# matrix of covariates (for the first and the following occasions)
# colums are: gender,race,educational level (2 columns),age,age^2)
X1 <- res$XX[,1,]
X2 <- res$XX[,2:TT,]

# estimate the model
est2f <- est_lm_cov_latent(S, X1, X2, k = 2, output = TRUE, out_se = TRUE)
summary(est2f)

# average transition probability matrix
PI <- round(apply(est2f$PI[,,,2:TT], c(1,2), mean), 4)

# Transition probability matrix for white females with high educational level
ind1 <- X1[,1] == 1 & X1[,2] == 0 & X1[,4] == 1)
PI1 <- round(apply(est2f$PI[,,ind1,2:TT], c(1,2), mean), 4)

# Transition probability matrix for non-white male, low educational level
ind2 <- (X1[,1] == 0 & X1[,2] == 1 & X1[,3] == 0 & X1[,4] == 0)
PI2 <- round(apply(est2f$PI[,,ind2,2:TT], c(1,2), mean), 4)

## End(Not run)
```

---

est_lm_cov_latent_cont — *Estimate LM model for continuous outcomes with covariates in the latent model*

Description

Main function for estimating the LM model for continuous outcomes with covariates in the latent model.

The function is no longer maintained. Please look at `lmestCont` **function.**

Usage

```
est_lm_cov_latent_cont(Y, X1 = NULL, X2 = NULL, yv = rep(1,nrow(Y)), k, start = 0,
                       tol = 10^-8, maxit = 1000, param = "multilogit",
                       Mu = NULL, Si = NULL, Be = NULL, Ga = NULL,
                       output = FALSE, out_se = FALSE)
```

Arguments

Y	array of continuous outcomes (n x TT x r)

`X1`	matrix of covariates affecting the initial probabilities (n x nc1)
`X2`	array of covariates affecting the transition probabilities (n x TT-1 x nc2)
`yv`	vector of frequencies of the available configurations
`k`	number of latent states
`start`	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
`tol`	tolerance level for checking convergence of the algorithm
`maxit`	maximum number of iterations of the algorithm
`param`	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
`Mu`	initial value of the conditional means (r x k) (if start=2)
`Si`	initial value of the var-cov matrix common to all states (r x r) (if start=2)
`Be`	intial value of the parameters affecting the logit for the initial probabilities (if start=2)
`Ga`	intial value of the parametes affecting the logit for the transition probabilities (if start=2)
`output`	to return additional output (V,PI,Piv,Ul)
`out_se`	to compute the information matrix and standard errors

Value

`lk`	maximum log-likelihood
`Be`	estimated array of the parameters affecting the logit for the initial probabilities
`Ga`	estimated array of the parameters affecting the logit for the transition probabilities
`Mu`	estimate of conditional means of the response variables
`Si`	estimate of var-cov matrix common to all states
`np`	number of free parameters
`aic`	value of AIC for model selection
`bic`	value of BIC for model selection
`lkv`	log-likelihood trace at every step
`Piv`	estimate of initial probability matrix
`PI`	estimate of transition probability matrices
`Ul`	matrix containing the predicted sequence of latent states by the local decoding method
`call`	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia, http://www.stat.unipg.it/bartolucci

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:
# Example based on multivariate longitudinal continuous data

# load data
require(mmm)
data(multiLongGaussian)
TT <- 4
res <- long2matrices(multiLongGaussian$ID, X = cbind(multiLongGaussian$X, multiLongGaussian$time),
      Y = cbind(multiLongGaussian$resp1, multiLongGaussian$resp2))
Y <- res$YY
X1 <- res$XX[,1,]
X2 <- res$XX[,2:TT,]

# estimate the model
est <- est_lm_cov_latent_cont(Y, X1, X2, k = 3, output = TRUE)
summary(est)

# average transition probability matrix
PI <- round(apply(est$PI[,,,2:TT], c(1,2), mean), 4)
PI

## End(Not run)
```

`est_lm_cov_manifest` *Estimate LM model with covariates in the measurement model*

Description

Main function for estimating LM model with covariates in the measurement model based on a global logit parameterization.

The function is no longer maintained. Please look at `lmest` **function**.

Usage

```
est_lm_cov_manifest(S, X, yv = rep(1,nrow(S)), k, q = NULL, mod = c("LM", "FM"),
                    tol = 10^-8, maxit = 1000, start = 0, mu = NULL, al = NULL,
                    be = NULL, si = NULL, rho = NULL, la = NULL, PI = NULL,
                    output = FALSE, out_se = FALSE)
```

Arguments

`S`	array of available configurations (n x TT) with categories starting from 0
`X`	array (n x TT x nc) of covariates with eventually includes lagged response (nc = number of covariates)
`yv`	vector of frequencies of the available configurations
`k`	number of latent states
`q`	number of support points for the AR(1) process
`mod`	model ("LM" = Latent Markov with stationary transition, "FM" = finite mixture)
`tol`	tolerance for the convergence (optional) and tolerance of conditional probability if tol>1 then return
`maxit`	maximum number of iterations of the algorithm
`start`	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
`mu`	starting value for mu (optional)
`al`	starting value for al (optional)
`be`	starting value for be (optional)
`si`	starting value for si when mod="FM" (optional)
`rho`	starting value for rho when mod="FM" (optional)
`la`	starting value for la (optional)
`PI`	starting value for PI (optional)
`output`	to return additional output (PRED0, PRED1)
`out_se`	TRUE for computing information matrix and standard errors

Value

`mu`	vector of cutpoints
`al`	support points for the latent states
`be`	estimate of the vector of regression parameters
`si`	sigma of the AR(1) process (mod = "FM")
`rho`	parameter vector for AR(1) process (mod = "FM")
`la`	vector of initial probabilities
`PI`	transition matrix
`lk`	maximum log-likelihood
`np`	number of parameters
`aic`	value of AIC index
`bic`	value of BIC index
`PRED0`	prediction of latent state
`PRED1`	prediction of the overall latent effect
`sebe`	standard errors for the regression parameters be
`selrho`	standard errors for logit type transformation of rho
`J1`	information matrix
`call`	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi - University of Perugia (IT)

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Bartolucci, F., Bacci, S. and Pennoni, F. (2014) Longitudinal analysis of the self-reported health status by mixture latent autoregressive models, *Journal of the Royal Statistical Society - series C*, **63**, pp. 267-288

Examples

```
## Not run:
# Example based on self-rated health status (SRHS) data

# load SRHS data
data(data_SRHS_long)
dataSRHS <- data_SRHS_long
head(dataSRHS)

res <- long2matrices(dataSRHS$id, X = cbind(dataSRHS$gender-1,
 dataSRHS$race == 2 | dataSRHS$race == 3, dataSRHS$education == 4,
dataSRHS$education == 5, dataSRHS$age-50, (dataSRHS$age-50)^2/100),
Y = dataSRHS$srhs)

X <- res$XX
S <- 5-res$YY

# *** fit stationary LM model
res0 <- vector("list", 10)
tol <- 10^-6;
for(k in 1:10){
  res0[[k]] <- est_lm_cov_manifest(S, X, k, 1, mod = "LM", tol)
   save.image("example_SRHS.RData")
}

# *** fit the mixture latent auto-regressive model
tol <- 0.005
res <- vector("list",4)
k <- 1
q <- 51
res[[k]] <- est_lm_cov_manifest(S, X, k, q, mod = "FM", tol, output = TRUE)
for(k in 2:4) res[[k]] <- est_lm_cov_manifest(S, X, k, q = 61, mod = "FM", tol, output = TRUE)

## End(Not run)
```

est_lm_mixed *Estimate mixed LM model*

Description

Main function for estimating the mixed LM model with discrete random effect in the latent model.

The function is no longer maintained. Please look at `lmestMixed` function

Usage

```
est_lm_mixed(S, yv = rep(1,nrow(S)), k1, k2, start = 0, tol = 10^-8, maxit = 1000,
                    out_se = FALSE)
```

Arguments

`S`	array of available response configurations (n x TT x r) with categories starting from 0
`yv`	vector of frequencies of the available configurations
`k1`	number of latent classes
`k2`	number of latent states
`start`	type of starting values (0 = deterministic, 1 = random)
`tol`	tolerance level for convergence
`maxit`	maximum number of iterations of the algorithm
`out_se`	to compute standard errors

Value

`la`	estimate of the mass probability vector (distribution of the random effects)
`Piv`	estimate of initial probabilities
`Pi`	estimate of transition probability matrices
`Psi`	estimate of conditional response probabilities
`lk`	maximum log-likelihood
`W`	posterior probabilities of the random effect
`np`	number of free parameters
`bic`	value of BIC for model selection
`call`	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi - University of Perugia (IT)

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:
# Example based of criminal data

# load data
data(data_criminal_sim)
out <- long2wide(data_criminal_sim, "id", "time", "sex",
 c("y1","y2","y3","y4","y5","y6","y7","y8","y9","y10"), aggr = T, full = 999)

XX <- out$XX
YY <- out$YY
freq <- out$freq
n1 <- sum(freq[XX[,1] == 1])
n2 <- sum(freq[XX[,1] == 2])
n <- sum(freq)

# fit mixed LM model only for females
YY <- YY[XX[,1] == 2,,]
freq <- freq[XX[,1] == 2]
k1 <- 2
k2 <- 2
res <- est_lm_mixed(YY, freq, k1, k2, tol = 10^-8)
summary(res)

## End(Not run)
```

---

est_mc_basic *Estimate basic Markov chain (MC) model*

Description

Main function for estimating the basic MC model.

The function is no longer maintained. Please look at `lmestMc` **function**.

Usage

```
est_mc_basic(S, yv, mod = 0, tol = 10^-8, maxit = 1000, out_se = FALSE)
```

Arguments

S	matrix (n x TT) of available configurations of the response variable with categories starting from 0
yv	vector of frequencies of the available configurations

`mod`	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
`tol`	tolerance level for convergence
`maxit`	maximum number of iterations of the algorithm
`out_se`	to compute the information matrix and standard errors

Value

`lk`	maximum log-likelihood
`piv`	estimate of initial probability vector
`Pi`	estimate of transition probability matrices
`np`	number of free parameters
`aic`	value of AIC for model selection
`bic`	value of BIC for model selection
`Fy`	estimated marginal distribution of the response variable for each time occasion
`sepiv`	standard errors for the initial probabilities
`sePi`	standard errors for the transition probabilities
`call`	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
# Example of drug consumption data

# load data
data(data_drug)
data_drug <- as.matrix(data_drug)
S <- data_drug[,1:5]-1
yv <- data_drug[,6]

# fit of the Basic MC model
out <- est_mc_basic(S, yv, mod = 1, out_se = TRUE)
summary(out)
```

est_mc_cov *Estimate Markov chain (MC) model with covariates*

Description

Main function for estimating the MC model with covariates.

The function is no longer maintained. Please look at lmestMc **function**.

Usage

```
est_mc_cov(S, X1 = NULL, X2 = NULL, yv = rep(1,nrow(S)), start = 0, tol = 10^-8,
   maxit = 1000, out_se = FALSE, output = FALSE, fort = TRUE)
```

Arguments

S	matrix of available configurations of the response variable (n x TT) with categories starting from 0
X1	matrix of covariates affecting the initial probabilities (n x nc1)
X2	array of covariates affecting the transition probabilities (n x TT-1 x nc2)
yv	vector of frequencies of the available configurations
start	type of starting values (0 = deterministic, 1 = random)
tol	tolerance level for checking convergence of the algorithm
maxit	maximum number of iterations of the algorithm
out_se	to compute the information matrix and standard errors
output	to return additional output (PI,Piv)
fort	to use fortran routine when possible (FALSE for not use fortran)

Value

lk	maximum log-likelihood
Be	estimated array of the parameters affecting the logit for the initial probabilities
Ga	estimated array of the parameters affecting the logit for the transition probabilities
np	number of free parameters
aic	value of AIC for model selection
bic	value of BIC for model selection
seBe	standard errors for Be
seGa	standard errors for Ga
Piv	estimate of initial probability matrix
PI	estimate of transition probability matrices
call	command used to call the function

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia, http://www.stat.unipg.it/bartolucci

References

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:

# Example based on criminal data

# load criminal data
data(data_criminal_sim)

#We consider the response variable referring of crime of type 5

out <- long2wide(data_criminal_sim, "id", "time", "sex",
"y5", aggr = T, full = 999)
XX <- out$XX-1
YY <- out$YY
freq <- out$freq
TT <- 6

X1 <- as.matrix(XX[,1])
X2 <- as.matrix(XX[,2:TT])
# estimate the model
res <- est_mc_cov(S = YY, yv = freq, X1 = X1, X2 = X2, output = TRUE)
summary(res)

# Initial probability for female
Piv0 <- round(colMeans(res$Piv[X1 == 0,]), 4)

# Initial probability for male
Piv1 <- round(colMeans(res$Piv[X1 == 1,]), 4)


## End(Not run)
```

`LMbasic-class` *Class* `'LMbasic'`

Description

An S3 class object created by `lmest` function for basic Latent Markov (LM) model.

Value

`lk`	maximum log-likelihood at convergence of the EM algorithm
`piv`	estimate of initial probability vector
`Pi`	estimate of transition probability matrices (k x k x TT)
`Psi`	estimate of conditional response probabilities (mb x k x r)
`np`	number of free parameters
`k`	optimal number of latent states
`aic`	value of the Akaike Information Criterion for model selection
`bic`	value of the Bayesian Information Criterion for model selection
`lkv`	log-likelihood trace at every step
`V`	array containing the estimated posterior probabilities of the latent states for each response configuration and time occasion
`n`	number of observations in the data
`TT`	number of time occasions
`modBasic`	model on the transition probabilities: default 0 for time-heterogeneous transition matrices, 1 for time-homogeneous transition matrices, 2 for partial time homogeneity based on two transition matrices one from 2 to (TT-1) and the other for TT.
`sepiv`	standard errors for the initial probabilities
`sePi`	standard errors for the transition probabilities
`sePsi`	standard errors for the conditional response probabilities
`Lk`	vector containing the values of the log-likelihood of the LM model with each `k` (latent states)
`Bic`	vector containing the values of the BIC for each `k`
`Aic`	vector containing the values of the AIC for each `k`
`call`	command used to call the function
`data`	data.frame given in input

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

`lmest`

LMbasiccont-class *Class* 'LMbasiccont'

Description

An S3 class object created by `lmestCont` function for the latent Markov (LM) model for continuous responses in long format.

Value

`lk`	maximum log-likelihood
`piv`	estimate of initial probability vector
`Pi`	estimate of transition probability matrices (k x k x TT)
`Mu`	estimate of conditional means of the response variables (r x k)
`Si`	estimate of var-cov matrix common to all states (r x r)
`np`	number of free parameters
`k`	optimal number of latent states
`aic`	value of the Akaike Information Criterion for model selection
`bic`	value of the Bayesian Information Criterion for model selection
`lkv`	log-likelihood trace at every step
`V`	array containing the posterior distribution of the latent states for each units and time occasion
`n`	number of observations in the data
`TT`	number of time occasions
`modBasic`	model on the transition probabilities: default 0 for time-heterogeneous transition matrices, 1 for time-homogeneous transition matrices, 2 for partial time homogeneity based on two transition matrices one from 2 to (TT-1) and the other for TT
`sepiv`	standard errors for the initial probabilities
`sePi`	standard errors for the transition probabilities
`seMu`	standard errors for the conditional means
`seSi`	standard errors for the var-cov matrix
`Lk`	vector containing the values of the log-likelihood of the LM model with each `k` (latent states)
`Bic`	vector containing the values of the BIC of the LM model with each `k` (latent states)
`Aic`	vector containing the values of the AIC of the LM model with each `k` (latent states)
`call`	command used to call the function
`data`	data frame given in input

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

lmestCont

---

lmest *Estimate Latent Markov models*

Description

Main function for estimating Latent Markov (LM) models for categorical responses.

Usage

```
lmest(responsesFormula = NULL, latentFormula = NULL,
      data, index, k = 1:4, start = 0,
      modSel = c("BIC", "AIC"), modBasic = 0,
      modManifest = c("LM", "FM"),
      paramLatent = c("multilogit", "difflogit"),
      weights = NULL, tol = 10^-8, maxit = 1000,
      out_se = FALSE, q = NULL, output = FALSE,
      parInit = list(piv = NULL, Pi = NULL, Psi = NULL,
                     Be = NULL, Ga = NULL, mu = NULL,
                     al = NULL, be = NULL, si = NULL,
                     rho = NULL, la = NULL, PI = NULL,
                     fixPsi = FALSE),
      fort = TRUE, seed = NULL)
```

Arguments

responsesFormula	a symbolic description of the model to fit. A detailed description is given in the 'Details' section
latentFormula	a symbolic description of the model to fit. A detailed description is given in the 'Details' section
data	a data.frame in long format
index	a character vector with two elements, the first indicating the name of the unit identifier, and the second the time occasions
k	an integer vector specifying the number of latent states (default: 1:4)
start	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
modSel	a string indicating the model selection criteria: "BIC" for Bayesian Information Criterion and "AIC" for Akaike Information Criterion Criterion

`modBasic`	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
`modManifest`	model for manifest distribution ("LM" = Latent Markov with stationary transition, "FM" = finite mixture)
`paramLatent`	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
`weights`	an optional vector of frequencies of the available response configurations
`tol`	tolerance level for convergence
`maxit`	maximum number of iterations of the algorithm
`out_se`	to compute the information matrix and standard errors (By default is set to FALSE)
`q`	number of support points for the AR(1) process (if modManifest ="FM")
`output`	to return additional output: `PRED0`, `PRED1` for the LM model with covariates in the measurement model (`LMmanifest-class`) and `V`, `PI`, `Piv`, `Ul` for the LM with covariates on the latent model (`LMlatent-class`)
`parInit`	list of initial model parameters when `"start = 2"`. For the list of parameters look at `LMbasic-class`, `LMlatent-class` and `LMmanifest-class`
`fort`	to use fortran routines when possible (By default is set to TRUE)
`seed`	an integer value with the random number generator state

Details

The function `lmest` is a general function for estimating LM models for categorical responses. The function requires data in long format and two additional columns indicating the unit identifier and the time occasions.

Covariates are allowed on manifest distribution (measurement model) or on the initial and transition probabilities (latent model). Two different formulas are employed to specify the different LM models, `responsesFormula` and `latentFormula`:

- `responsesFormula` is used to specify the measurament model:
  - `responsesFormula = x1 + x2 ~ NULL`
    the LM model without covariates and two responses (`x1` and `x2`) is specified;
  - `responsesFormula = NULL`
    all the columns in the data except the `"id"` and `"time"` columns are used to estimate the LM model without covariates;
  - `responsesFormula = x1 + x2 ~ x3 + x4`
    the LM model with two responses (`x1` and `x2`) and two covariates (`x3` and `x4`) in the measurement model is specified;
- `latentFormula` is used to specify the LM model with covariates in the latent model:
  - `responsesFormula = x1 + x2 ~ NULL`
    `latentFormula = ~ x3 + x4 | x5 + x6`
    the LM model with two responses (`x1` and `x2`) and two covariates affecting the initial probabilities (`x3` and `x4`) and other two affecting the transition probabilities (`x5` and `x6`) is specified;

- `responsesFormula = x1 + x2 ~ NULL`
  `latentFormula = ~ 1 | x3 + x4`
  (or `latentFormula = ~ NULL | x3 + x4`)
  the covariates affect only the transition probabilities and an intercept is specified for the intial probabilities;
- `responsesFormula = x1 + x2 ~ NULL`
  `latentFormula = ~ x3 + x4`
  the LM model with two covariates (`x3` and `x4`) affecting both the initial and transition probabilities is specified;
- `responsesFormula = x1 + x2 ~ NULL`
  `latentFormula = ~ NULL | NULL`
  (or `latentFormula = ~ 1 | 1`)
  the LM model with only an intercept on the initial and transition probabilities is specified.

The function also allows us to deal with missing responses, including drop-out and non-monotonic missingness, under the missing-at-random assumption. Missing values for the covariates are not allowed. The LM model with individual covariates in the measurement model is estimated only for complete univariate responses.

For continuous outcomes see the function `lmestCont`.

Value

Returns an object of class `'LMbasic'` for the model without covariates (see `LMbasic-class`), or an object of class `'LMmanifest'` for the model with covariates on the manifest model (see `LMmanifest-class`), or an object of class `'LMlatent'` for the model with covariates on the latent model (see `LMlatent-class`).

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

References

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.

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
### Basic LM model

data("data_SRHS_long")
SRHS <- data_SRHS_long[1:2400,]

# Categories rescaled to vary from 1 (“poor”) to 5 (“excellent”)

SRHS$srhs <- 5 - SRHS$srhs
```

```
out <- lmest(responsesFormula = srhs ~ NULL,
            index = c("id","t"),
            data = SRHS,
            k = 3,
            start = 1,
            modBasic = 1,
            seed = 123)
out
summary(out)




## Not run:

## Basic LM model with model selection using BIC

out1 <- lmest(responsesFormula = srhs ~ NULL,
            index = c("id","t"),
            data = SRHS,
            k = 1:5,
            tol = 1e-8,
            modBasic = 1,
            seed = 123)
out1
out1$Bic

# Basic LM model with model selection using AIC

out2 <- lmest(responsesFormula = srhs ~ NULL,
            index = c("id","t"),
            data = SRHS,
            k = 1:5,
            tol = 1e-8,
            modBasic = 1,
            modSel = "AIC",
            seed = 123)
out2
out2$Aic

# Criminal data

data(data_criminal_sim)
data_criminal_sim = data.frame(data_criminal_sim)

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

out3 <- lmest(responsesFormula = responsesFormula,
              index = c("id","time"),
              data = data_criminal_sim,
              k = 1:7,modBasic = 1,tol = 10^-4)
```

```
out3

# Example of drug consumption data

data("data_drug")
long <- data_drug[,-6]-1
long <- data.frame(id = 1:nrow(long),long)
long <- reshape(long,direction = "long",
                idvar = "id",
                varying = list(2:ncol(long)))

out4 <- lmest(index = c("id","time"),k = 3, data = long,
             weights = data_drug[,6],modBasic = 1)

out4
summary(out4)

### LM model with covariates in the latent model
# Covariates: gender, race, educational level (2 columns), age and age^2

out5 <- lmest(responsesFormula = srhs ~ NULL,
             latentFormula =  ~
             I(gender - 1) +
             I( 0 + (race == 2) + (race == 3)) +
             I(0 + (education == 4)) +
             I(0 + (education == 5)) +
             I(age - 50) + I((age-50)^2/100),
             index = c("id","t"),
             data = SRHS,
             k = 2,
             paramLatent = "multilogit",
             start = 0)

out5
summary(out5)

### LM model with the above covariates in the measurement model

out6 <- lmest(responsesFormula = srhs ~ -1 +
             I(gender - 1) +
             I( 0 + (race == 2) + (race == 3)) +
             I(0 + (education == 4)) +
             I(0 + (education == 5)) + I(age - 50) +
             I((age-50)^2/100),
             index = c("id","t"),
             data = SRHS,
             k = 2,
             modManifest = "LM",
             out_se = TRUE,
             tol = 1e-8,
             start = 1,
             seed = 123)
out6
```

```
summary(out6)

## End(Not run)
```

---

lmestCont *Estimate Latent Markov models for continuous responses*

Description

Main function for estimating Latent Markov (LM) models for continuous responses.

Usage

```
lmestCont(responsesFormula = NULL, latentFormula = NULL,
          data, index, k = 1:4, start = 0,
          modSel = c("BIC", "AIC"), modBasic = 0,
          paramLatent = c("multilogit", "difflogit"),
          weights = NULL, tol = 10^-10,
          maxit = 5000, out_se = FALSE, output = FALSE,
          parInit = list(piv = NULL, Pi = NULL,
                         Mu = NULL, Si = NULL,
                         Be = NULL, Ga = NULL),
          fort = TRUE, seed = NULL)
```

Arguments

`responsesFormula`	a symbolic description of the model to be fitted. A detailed description is given in the 'Details' section
`latentFormula`	a symbolic description of the model to be fitted. A detailed description is given in the 'Details' section
`data`	a `data.frame` in long format
`index`	a character vector with two elements, the first indicating the name of the unit identifier, and the second the time occasions
`k`	an integer vector specifying the number of latent states (default: `1:4`)
`start`	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
`modSel`	a string indicating the model selection criteria: "BIC" for Bayesian Information Criterion and "AIC" for Akaike Information Criterion Criterion
`modBasic`	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
`paramLatent`	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
`weights`	an optional vector of frequencies of the available response configurations

`tol`	tolerance level for convergence
`maxit`	maximum number of iterations of the algorithm
`out_se`	to compute the information matrix and standard errors (By default is set to FALSE)
`output`	to return additional output (`V`, `PI`, `Piv`, `Ul`) (`LMlatentcont-class`)
`parInit`	list of initial model parameters when `"start = 2"`. For the list of parameters look at `LMbasiccont-class` and `LMlatentcont-class`
`fort`	to use fortran routines when possible (By default is set to TRUE)
`seed`	an integer value with the random number generator state

Details

The function `lmestCont` is a general function for estimating LM models for continuous responses. The function requires data in long format and two additional columns indicating the unit identifier and the time occasions.

Covariates are allowed on the initial and transition probabilities (latent model). Two different formulas are employed to specify the different LM models, `responsesFormula` and `latentFormula`:

- `responsesFormula` is used to specify the measurement model:
  - `responsesFormula = x1 + x2 ~ NULL`
    the LM model without covariates and two responses (`x1` and `x2`) is specified;
  - `responsesFormula = NULL`
    all the columns in the data except the `"id"` and `"time"` columns are used to estimate the LM model without covariates;
- `latentFormula` is used to specify the LM model with covariates in the latent model:
  - `responsesFormula = x1 + x2 ~ NULL`
    `latentFormula = ~ x3 + x4 | x5 + x6`
    the LM model with two responses (`x1` and `x2`) and two covariates affecting the initial probabilities (`x3` and `x4`) and other two affecting the transition probabilities (`x5` and `x6`) is specified;
  - `responsesFormula = x1 + x2 ~ NULL`
    `latentFormula = ~ 1 | x3 + x4`
    (or `latentFormula = ~ NULL | x3 + x4`)
    the covariates affect only the transition probabilities and an intercept is specified for the intial probabilities;
  - `responsesFormula = x1 + x2 ~ NULL`
    `latentFormula = ~ x3 + x4`
    the LM model with two covariates (`x3` and `x4`) affecting both the initial and transition probabilities is specified;
  - `responsesFormula = x1 + x2 ~ NULL`
    `latentFormula = ~ NULL | NULL`
    (or `latentFormula = ~ 1 | 1`)
    the LM model with only an intercept on the initial and transition probabilities is specified.

The function also allows us to deal with missing responses using the `mix` package for imputing the missing values. Missing values for the covariates are not allowed.

For categorical outcomes see the function `lmest`.

Value

Returns an object of class 'LMbasiccont' for the model without covariates (see LMbasiccont-class), or an object of class 'LMlatentcont' for the model with covariates on the latent model (see LMlatentcont-class).

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

References

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.

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

See Also

lmestFormula

Examples

```
## Not run:
require(mmm)
data(multiLongGaussian)

t <- rep(1:4, times  = max(multiLongGaussian$ID))
multiLongGaussian <- data.frame(t = t, multiLongGaussian)

# Basic LM model

out <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                  index = c("ID", "t"),
                  data = multiLongGaussian,
                  k =3,
                  modBasic=1,
                  tol=10^-5)

out
summary(out)



# Basic LM model with model selection using BIC

out1 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                  index = c("ID", "t"),
                  data = multiLongGaussian,
                  k = 1:5,
                  modBasic=1,
                  tol=10^-5)
```

```
out1
out1$Bic

# Basic LM model with model selection using AIC

out2 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                  index = c("ID", "t"),
                  data = multiLongGaussian,
                  k = 1:5,
                  modBasic=1,
                  modSel = "AIC",
                  tol=10^-5)
out2
out2$Aic


# LM model with covariates in the latent model

out3 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                   latentFormula = ~ X + time,
                   index = c("ID", "t"),
                   data = multiLongGaussian,
                   k = 3,
                   output=TRUE)

out3
summary(out3)

out4 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                   latentFormula = ~ X + time | X + time,
                   index = c("ID", "t"),
                   data = multiLongGaussian,
                   k = 3,
                   output=TRUE)

out4
summary(out4)


## End(Not run)
```

`lmestData` *Data for* `LMest` *functions*

Description

An object of class `lmestData` containing data in long format, some necessary information on the data structure and objects for the estimation functions.

Usage

```
lmestData(data, id = NULL, time = NULL,
          idAsFactor = TRUE, timeAsFactor = TRUE,
          responsesFormula = NULL, latentFormula = NULL,
          na.rm = FALSE, check.names = FALSE)
```

Arguments

`data`	a matrix or data frame in long format of observation
`id`	a numeric vector or a string indicating the column with the unit identifier. If NULL, the first column is considered
`time`	a numeric vector or a string indicating the column with the time occasions. If NULL, the second column is considered, and if the `id` is not NULL, the function will automatically add the column with the time occasions
`idAsFactor`	a logical value indicating whether or not the column with the ids is converted to a factor. (By default is set to TRUE)
`timeAsFactor`	a logical value indicating whether or not the column with the time occasions is converted in a factor. (By default is set to TRUE)
`responsesFormula`	A detailed description is given in `lmest`, `lmestCont`
`latentFormula`	A detailed description is given in `lmest`, `lmestCont`
`na.rm`	a logical value indicating whether or not the observation with at least a missing value is removed (By default is set to FALSE)
`check.names`	a logical value indicating whether or not the names of the variables are syntactically valid, and adjusted if necessary. (By default is set to FALSE)

Value

An object of class `'lmestData'` with the following objects:

`data`	a data.frame object to use in the estimation functions
`id`	a integer vector with the unit identifier
`time`	a integer vector with the time occasions
`n`	the number of observation
`TT`	an integer value indicating number of time occasions
`d`	an interger value indicating the number of variables (columns except id and time)
`Y`	the response variables
`Xmanifest`	the variables affecting the measurement model if specified in `responsesFormula`
`Xinitial`	the variables affecting the initial probabilities of the latent model if specified in `latentFormula`
`Xtrans`	the variables affecting the transition probabilities of the latent model if specified in `latentFormula`

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
require(mmm)
data(multiLongGaussian)
str(multiLongGaussian)

## Data with continous resposes

dt <- lmestData(data = multiLongGaussian, id = "ID")
str(dt)

## Summary of each variable and for each time

summary(dt)

## Summary of each variable

summary(dt, type = "cross")

## Summary of each variable by time

summary(dt, type = "year")

plot(dt)
plot(dt, typePlot = "sh")

#######################

## Not run:

data("data_criminal_sim")

dt1 <- lmestData(data = data_criminal_sim, id = "id", time = "time")
str(dt1)

summary(dt1, varType = rep("d",ncol(dt1$Y)))

dt2 <- lmestData(data = data_criminal_sim, id = "id", time = "time",
                 responsesFormula = y1 + y2 ~ y3, latentFormula = ~ y7 + y8 | y9 + y10)
str(dt2)

## Summary for responses, covariates on the manifest distribution,
## covariates on intial and transition probabilities

summary(dt2, dataSummary = "responses",varType = rep("d",ncol(dt2$Y)))
summary(dt2, dataSummary = "manifest",varType = rep("d",ncol(dt2$Xmanifest)))
summary(dt2, dataSummary = "initial",varType = rep("d",ncol(dt2$Xinitial)))
summary(dt2, dataSummary = "transition",varType = rep("d",ncol(dt2$Xtrans)))
```

```
## End(Not run)
```

`lmestDecoding` *Perform local and global decoding*

Description

Function that performs local and global decoding (Viterbi algorithm) from the output of `lmest`, `lmestCont`, and `lmestMixed`.

Usage

```
lmestDecoding(est, sequence = NULL, fort = TRUE, ...)
## S3 method for class 'LMbasic'
lmestDecoding(est, sequence = NULL,fort = TRUE, ...)
## S3 method for class 'LMmanifest'
lmestDecoding(est, sequence = NULL, fort = TRUE, ...)
## S3 method for class 'LMlatent'
lmestDecoding(est, sequence = NULL, fort = TRUE,...)
## S3 method for class 'LMbasiccont'
lmestDecoding(est, sequence = NULL, fort = TRUE,...)
## S3 method for class 'LMmixed'
lmestDecoding(est, sequence = NULL, fort = TRUE,...)
```

Arguments

`est`	an object obtained from a call to `lmest`, `lmestCont`, and `lmestMixed`
`sequence`	an integer vector indicating the units for the decoding. If `NULL` the whole observations are considered. (By default is set to NULL)
`fort`	to use fortran routines when possible (By default is set to TRUE)
`...`	further arguments

Value

`Ul`	matrix of local decoded states corresponding to each row of Y
`Ug`	matrix of global decoded states corresponding to each row of Y

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

References

Viterbi A. (1967) Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, **13**, 260-269.

Juan B., Rabiner L. (1991) Hidden Markov Models for Speech Recognition. *Technometrics*, **33**, 251-272.

Examples

```
# Decoding for basic LM model

data("data_drug")
long <- data_drug[,-6]-1
long <- data.frame(id = 1:nrow(long),long)
long <- reshape(long,direction = "long",
                idvar = "id",
                varying = list(2:ncol(long)))

est <- lmest(index = c("id","time"),k = 3, data = long,
              weights = data_drug[,6], modBasic = 1)

# Decoding for a single sequence

out1 <- lmestDecoding(est, sequence = 1)

out2 <- lmestDecoding(est, sequence = 1:4)

# Decoding for all sequences

out3 <- lmestDecoding(est)

## Not run:
# Decoding for LM model  with covariates on the initial and transition probabilities

data("data_SRHS_long")
SRHS <- data_SRHS_long[1:2400,]

# Categories rescaled to vary from 1 (“poor”) to 5 (“excellent”)

SRHS$srhs <- 5 - SRHS$srhs

est2 <- lmest(responsesFormula = srhs ~ NULL,
             latentFormula =  ~
             I(gender - 1) +
             I( 0 + (race == 2) + (race == 3)) +
             I(0 + (education == 4)) +
             I(0 + (education == 5)) +
             I(age - 50) + I((age-50)^2/100),
             index = c("id","t"),
             data = SRHS,
             k = 2,
```

```
              paramLatent = "difflogit",
              output = TRUE)

# Decoding for a single sequence

out3 <- lmestDecoding(est2, sequence = 1)

# Decoding for the first three sequences

out4 <- lmestDecoding(est2, sequence = 1:3)

# Decoding for all sequences

out5 <- lmestDecoding(est2)

## End(Not run)
```

---

lmestFormula *Formulas for* LMest *functions*

Description

Bulding formulas for lmest, lmestCont, lmestMixed, and lmestMc.

Usage

```
lmestFormula(data,
                response, manifest = NULL,
                LatentInitial = NULL, LatentTransition = NULL,
                AddInterceptManifest = FALSE,
                AddInterceptInitial = TRUE,
                AddInterceptTransition = TRUE, responseStart = TRUE,
                manifestStart = TRUE, LatentInitialStart = TRUE,
                LatentTransitionStart = TRUE)
```

Arguments

data	a data.frame or a matrix of data
response	a numeric or character vector indicating the column indices or the names for the response variables
manifest	a numeric or character vector indicating the column indices or the names for the covariates affecting the measurement model
LatentInitial	a numeric or character vector indicating the column indices or the names for the covariates affecting the initial probabilities
LatentTransition	a numeric or character vector indicating the column indices or the names for the covariates affecting the transition probabilities

AddInterceptManifest
: a logical value indicating whether the intercept is added to the covariates affecting the measurement model

AddInterceptInitial
: a logical value indicating whether the intercept is added to covariates affecting the initial probabilities

AddInterceptTransition
: a logical value indicating whether the intercept is added to covariates affecting the transition probabilities

responseStart
: a logical value indicating whether the response variables names start with response argument

manifestStart
: a logical value indicating whether the covariates names start with manifest argument

LatentInitialStart
: a logical value indicating whether the covariates names start with LatentInitial argument

LatentTransitionStart
: a logical value indicating whether the covariates names start with LatentTransition argument

Details

Generates formulas for responsesFormula and latentFormula to use in lmest, lmestCont, lmestMixed, and lmestMc.

Value

Returns a list with responsesFormula and latentFormula objects.

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
data(data_SRHS_long)
names(data_SRHS_long)
# Formula with response srhs and covariates for both initail and transition
# gender,race,educational,age.

## With intercept
fm <- lmestFormula(data = data_SRHS_long,
                   response = "srhs",
                   LatentInitial = 3:6, LatentTransition = 3:6)
fm

## Without intercept

fm <- lmestFormula(data = data_SRHS_long,
```

```
                   response = "srhs",
                   LatentInitial = 3:6, LatentTransition = 3:6,
                   AddInterceptInitial = FALSE,AddInterceptTransition = FALSE)
fm

######

data(data_criminal_sim)
str(data_criminal_sim)

# Formula with only the responses from y1 to y10

fm <- lmestFormula(data = data_criminal_sim,response = "y")$responsesFormula
fm

# Formula with only the responses from y1 to y10 and intercept for manifest

fm <- lmestFormula(data = data_criminal_sim,
                  response = "y",AddInterceptManifest = TRUE)$responsesFormula
fm


#####

require(mmm)
data(multiLongGaussian)
names(multiLongGaussian)

# Formula with response resp1, resp2, covariate for manifest,
# X covariates for initail and time covariate for transition

fm <- lmestFormula(data = multiLongGaussian,
                    response = c("resp"),
                    LatentInitial = "X",LatentTransition = "time")
fm
## Wrong model because two variable start with X.
## Check the starts arguments. For the right model:

fm <- lmestFormula(data = multiLongGaussian,
                    response = c("resp"),
                    LatentInitial = "X",LatentTransition = "time",
                    LatentInitialStart = FALSE)
fm
## or
fm <- lmestFormula(data = multiLongGaussian,
                    response = c("resp"),
                    LatentInitial = 4,LatentTransition = "time",
                    LatentInitialStart = FALSE)
fm

## Not run:
data(data_criminal_sim)
data_criminal_sim <- data.frame(data_criminal_sim)
```

```
# Estimate mixed LM model for females

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

out <- lmest(responsesFormula = responsesFormula,
             index = c("id","time"),
             data = data_criminal_sim,
             k = 2)

## End(Not run)
```

`lmestMc` *Estimate Markov Chain models*

Description

Main function for estimating Markov Chain (MC) models for categorical responses with or without covariates.

Usage

```
lmestMc(responsesFormula = NULL,
        data, index, start = 0,
        modBasic = 0, weights = NULL,
        tol = 10^-8, maxit = 1000,
        out_se = FALSE, output = FALSE, fort = TRUE, seed = NULL)
```

Arguments

`responsesFormula`	a symbolic description of the model to fit. A detailed description is given in the 'Details' section
`data`	a `data.frame` in long format
`index`	a character vector with two elements, the first indicating the name of the unit identifier, and the second the time occasions
`start`	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
`modBasic`	model on the transition probabilities (0 for time-heter., 1 for time-homog., from 2 to (TT-1) partial homog. of that order)
`weights`	an optional vector of frequencies of the available response configurations
`tol`	tolerance level for convergence
`maxit`	maximum number of iterations of the algorithm
`out_se`	to compute the information matrix and standard errors (FALSE is the default option)
`output`	to return additional output (`PI`,`Piv`) (`MCcov-class`)
`fort`	to use fortran routines when possible (By default is set to TRUE)
`seed`	An integer value with the random number generator state.

Details

The function `lmestMc` estimates the basic MC model and the MC model with covariates for categorical responses. The function requires data in long format and two additional column indicating the unit identifier and the time occasions.

`responsesFormula` is used to specify the basic MC models and the model with covariates:

- `responsesFormula = x1 + x2 ~ NULL`
  the MC model without covariates and two responses (`x1` and `x2`) is specified;
- `responsesFormula = NULL`
  all the columns in the data except the `"id"` and `"time"` columns are used to estimate MC without covariates;
- `responsesFormula = x1 ~ x2 + x3 | x4 + x5`
  the MC model with one response (`x1`), two covariates affecting the initial probabilities (`x2` and `x3`) and other two different covariates affecting the transition probabilities (`x4` and `x5`) is specified;
- `responsesFormula = x1 ~ x2 + x3`
  the MC model with one response (`x1`) and two covariates (`x2` and `x3`) affecting both the initial and transition probabilities is specified.

Missing responses are not allowed.

Value

Returns an object of class `'MCbasic'` for the basic model without covariates (see `MCbasic-class`), or an object of class `'MCcov'` for the model with covariates (see `MCcov-class`).

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

References

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.

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:
# Basic Markov Chain  model

data("RLMSlong")

# Categories  rescaled from 1 “absolutely unsatisfied” to  5 “absolutely satisfied”

RLMSlong$value <- 5 - RLMSlong$value

out <- lmestMc(responsesFormula = value ~ NULL,
```

```
                index = c("id","time"),
                modBasic = 1,
                data = RLMSlong)

out
summary(out)



# Example of drug consumption data

data("data_drug")
long <- data_drug[,-6]
long <- data.frame(id = 1:nrow(long),long)
long <- reshape(long,direction = "long",
                idvar = "id",
                varying = list(2:ncol(long)))

out1 <- lmestMc(index = c("id","time"), data = long,
                weights = data_drug[,6], modBasic = 1, out_se = TRUE)

out1

### MC model with covariates
### Covariates: gender, race, educational level (2 columns), age and age^2

data("data_SRHS_long")
SRHS <- data_SRHS_long[1:2400,]

# Categories of the responses rescaled from 1 “poor” to 5 “excellent”

SRHS$srhs <- 5 - SRHS$srhs


out2 <- lmestMc(responsesFormula = srhs ~
                         I( 0 + (race==2) + (race == 3)) +
                         I(0 + (education == 4)) +
                         I(0 + (education == 5)) +
                         I(age - 50) +
                         I((age-50)^2/100),
                 index = c("id","t"),
                 data = SRHS)
out2
summary(out2)

# Criminal data

data(data_criminal_sim)
data_criminal_sim = data.frame(data_criminal_sim)

out3 <- lmestMc(responsesFormula = y5~sex,
              index = c("id","time"), data = data_criminal_sim, output = TRUE)
```

```
out3

## End(Not run)
```

---

lmestMixed *Estimate mixed Latent Markov models*

Description

Main function for estimating the mixed latent Markov (LM) models for categorical responses with discrete random effects in the latent model.

Usage

```
lmestMixed(responsesFormula = NULL,
           data, index, k1, k2, start = 0,
           weights = NULL, tol = 10^-8, maxit = 1000,
           out_se = FALSE, seed = NULL)
```

Arguments

responsesFormula	a symbolic description of the model to fit. A detailed description is given in the 'Details' section
data	a data.frame in long format
index	a character vector with two elements, the first indicating the name of the unit identifier, and the second the time occasions
k1	number of latent classes
k2	number of latent states
start	type of starting values (0 = deterministic, 1 = random, 2 = initial values in input)
weights	an optional vector of frequencies of the available response configurations
tol	tolerance level for convergence
maxit	maximum number of iterations of the algorithm
out_se	to compute the information matrix and standard errors (FALSE is the default option)
seed	an integer value with the random number generator state

Details

The function lmestMixed estimates the mixed LM for categorical data. The function requires data in long format and two additional columns indicating the unit identifier and the time occasions.

responsesFormula is used to specify the responses of the mixed LM model:

- `responsesFormula = x1 + x2 ~ NULL`
  the mixed LM model with two categorical responses (`x1` and `x2`) is specified;
- `responsesFormula = NULL`
  all the columns in the data except the `"id"` and `"time"` columns are used to estimate the mixed LM.

Missing responses are not allowed.

Value

Returns an object of class `'LMmixed'` (see `LMmixed-class`).

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

References

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.

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
## Not run:
# Example based on criminal data

data(data_criminal_sim)
data_criminal_sim <- data.frame(data_criminal_sim)
# Estimate mixed LM model for females

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

# fit mixed LM model only for females
out <- lmestMixed(responsesFormula = responsesFormula,
                  index = c("id","time"),
                  k1 = 2,
                  k2 = 2,
                  data = data_criminal_sim[data_criminal_sim$sex == 2,])
out
summary(out)

## End(Not run)
```

`lmestSearch` *Search for the global maximum of the log-likelihood*

Description

Function that searches for the global maximum of the log-likelihood of different models and selects the optimal number of states.

Usage

```
lmestSearch(responsesFormula = NULL, latentFormula = NULL,
            data, index, k,
            version = c("categorical", "continuous"),
            weights = NULL, nrep = 2, tol1 = 10^-5,
            tol2 = 10^-10, out_se = FALSE, seed = NULL, ...)
```

Arguments

`responsesFormula`	a symbolic description of the model to fit. A detailed description is given in the 'Details' section of `lmest`
`latentFormula`	a symbolic description of the model to fit. A detailed description is given in the 'Details' section of `lmest`
`data`	a `data.frame` in long format
`index`	a character vector with two elements, the first indicating the name of the unit identifier, and the second the time occasions
`k`	a vector of integer values for the number of latent states
`weights`	an optional vector of frequencies of the available response configurations
`version`	type of responses for the LM model: "categorical" and "continuous"
`nrep`	number of repetitions of each random initialization
`tol1`	tolerance level for checking convergence of the algorithm in the random initializations
`tol2`	tolerance level for checking convergence of the algorithm in the last deterministic initialization
`out_se`	to compute the information matrix and standard errors (FALSE is the default option)
`seed`	an integer value with the random number generator
`...`	additional arguments to be passed to functions `lmest` or `lmestCont`

Details

The function combines deterministic and random initializations strategy to reach the global maximum of the model log-likelihood. It uses one deterministic initialization (`start=0`) and a number of random initializations (`start=1`) proportional to the number of latent states. The tolerance level is set equal to 10^-5. Starting from the best solution obtained in this way, a final run is performed (`start=2`) with a default tolerance level equal to 10^-10.

Missing responses are allowed according to the model to be estimated.

Value

Returns an object of class `'LMsearch'` with the following components:

`out.single`	Output of every LM model estimated for each number of latent states given in input
`Aic`	Values the Akaike Information Criterion for each number of latent states given in input
`Bic`	Values of the Bayesian Information Criterion for each number of latent states given in input
`lkv`	Values of log-likelihood for each number of latent states given in input.

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

References

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.

Bartolucci, F., Farcomeni, A. and Pennoni, F. (2013) *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC press.

Examples

```
### Example with data on drug use in wide format

data("data_drug")
long <- data_drug[,-6]

# add labels referred to the identifier

long <- data.frame(id = 1:nrow(long),long)

# reshape data from the wide to the long format

long <- reshape(long,direction = "long",
                idvar = "id",
                varying = list(2:ncol(long)))
```

```
out <- lmestSearch(data = long,
                   index = c("id","time"),
                   version = "categorical",
                   k = 1:3,
                   weights = data_drug[,6],
                   modBasic = 1,
                   seed = 123)

out
summary(out$out.single[[3]])

## Not run:

### Example with data on self rated health

# LM model with covariates in the measurement model

data("data_SRHS_long")
SRHS <- data_SRHS_long[1:1000,]

# Categories rescaled to vary from 1 ("poor") to 5 ("excellent")

SRHS$srhs <- 5 - SRHS$srhs

out1 <- lmestSearch(data = SRHS,
                    index = c("id","t"),
              version = "categorical",
             responsesFormula = srhs ~ -1 +
             I(gender - 1) +
             I( 0 + (race == 2) + (race == 3)) +
             I(0 + (education == 4)) +
             I(0 + (education == 5)) + I(age - 50) +
             I((age-50)^2/100),
                   k = 1:2,
                   out_se = TRUE,
                   seed = 123)
summary(out1)
summary(out1$out.single[[2]])

## End(Not run)
```

`LMlatent-class` *Class* 'LMlatent'

Description

An S3 class object created by `lmest` for Latent Markov (LM) model with covariates in the latent model.

Value

lk	maximum log-likelihood
Be	estimated array of the parameters affecting the logit for the initial probabilities
Ga	estimated array of the parameters affecting the logit for the transition probabilities
Piv	estimate of initial probability matrix. The first state is used as reference category when param = "multilogit"
PI	estimate of transition probability matrices. State u is used as reference category when paramLatent = "multilogit"
Psi	estimate of conditional response probabilities (mb x k x r)
np	number of free parameters
k	optimal number of latent states
aic	value of the Akaike Information Criterion for model selection
bic	value of the Bayesian Information Criterion for model selection
lkv	log-likelihood trace at every step of the EM algorithm
n	number of observations in the data
TT	number of time occasions
paramLatent	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
V	array containing the posterior distribution of the latent states for each response configuration and time occasion
Ul	matrix containing the predicted sequence of latent states by the local decoding method
sePsi	standard errors for the conditional response matrix
seBe	standard errors for Be
seGa	standard errors for Ga
Lk	vector containing the values of the log-likelihood of the LM model with each k (latent states)
Bic	vector containing the values of the BIC for each k
Aic	vector containing the values of the AIC for each k
call	command used to call the function
data	Data frame given in input

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

lmest

LMlatentcont-class *Class* 'LMlatentcont'

Description

An S3 class object created by lmestCont for the Latent Markov (LM) model for continuous responses in long format with covariates in the latent model.

Value

lk	maximum log-likelihood
Be	estimated array of the parameters affecting the logit for the initial probabilities
Ga	estimated array of the parameters affecting the logit for the transition probabilities
Mu	estimate of conditional means of the response variables
Si	estimate of var-cov matrix common to all states
np	number of free parameters
k	optimal number of latent states
aic	value of the Akaike Information Criterion for model selection
bic	value of the Bayesian Information Criterion for model selection
lkv	log-likelihood trace at every step
n	number of observations in the data
TT	number of time occasions
paramLatent	type of parametrization for the transition probabilities ("multilogit" = standard multinomial logit for every row of the transition matrix, "difflogit" = multinomial logit based on the difference between two sets of parameters)
seMu	standard errors for the conditional means
seSi	standard errors for the var-cov matrix
seBe	standard errors for Be
seGa	standard errors for Ga
PI	estimate of transition probability matrices
Piv	estimate of initial probability matrix
Ul	matrix containing the predicted sequence of latent states by the local decoding method
Lk	vector containing the values of the log-likelihood of the LM model with each k (latent states)
Bic	vector containing the values of the BIC of the LM model with each k (latent states)
Aic	vector containing the values of the AIC of the LM model with each k (latent states)
call	command used to call the function
data	data frame given in input

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

`lmestCont`

`LMmanifest-class` *Class* `'LMmanifest'`

Description

An S3 class object created by `lmest` for Latent Markov (LM) model with covariates in the measurement model.

Value

`mu`	vector of cut-points
`al`	support points for the latent states
`be`	estimate of the vector of regression parameters
`si`	sigma of the AR(1) process (mod = "FM")
`rho`	parameter vector for AR(1) process (mod = "FM")
`la`	vector of initial probabilities
`PI`	transition matrix
`lk`	maximum log-likelihood
`np`	number of parameters
`k`	optimal number of latent states
`aic`	value of the Akaike Information Criterion
`bic`	value of Bayesian Information Criterion
`n`	number of observations in the data
`TT`	number of time occasions
`modManifest`	for LM model with covariates on the manifest model: "LM" = Latent Markov with stationary transition, "FM" = finite mixture model where a mixture of AR(1) processes is estimated with common variance and specific correlation coefficients
`sebe`	standard errors for the regression parameters be
`selrho`	standard errors for logit type transformation of rho
`J1`	information matrix
`PRED0`	prediction of latent state
`PRED1`	prediction of the overall latent effect

Lk	vector containing the values of the log-likelihood of the LM model with each k (latent states)
Bic	vector containing the values of the BIC for each k
Aic	vector containing the values of the AIC for each k
call	command used to call the function
data	data frame given in input

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

lmest

LMmixed-class *Class* 'LMmixed'

Description

An S3 class object created by lmestMixed for the mixed latent Markov (LM) models for categorical data in long format.

Value

la	estimate of the mass probability vector (distribution of the random effects)
Piv	estimate of initial probabilities
Pi	estimate of transition probability matrices
Psi	estimate of conditional response probabilities
lk	maximum log-likelihood
W	posterior probabilities of the random effect
np	number of free parameters
k1	number of support points (latent classes) of the latent variable defining the unobserved clusters
k2	number of support points (latent states) of the latent variable defining the first-order Markov process
bic	value of the Akaike Information Criterion for model selection
aic	value of the Akaike Information Criterion for model selection
n	number of observations in the data
TT	number of time occasions
sela	standard errors for la
sePiv	estimate of initial probability matrix

sePi	standard errors for the transition probabilities
sePsi	standard errors for the conditional response matrix
call	command used to call the function
data	the input data

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

lmestMixed

---

long2matrices *From data in the long format to data in array format*

Description

Function that transforms data in the long format to data in array format.

Usage

```
long2matrices(id, time = NULL, X = NULL, Y)
```

Arguments

id	vector of subjects id
time	vector of time occasions
X	matrix of covariates in long format
Y	matrix of responses in long format

Value

XX	array of covariates (n x TT x nc)
YY	array of responses (n x TT x r)

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
# Example based on SRHS data

# load SRHS data
data(data_SRHS_long)
dataSRHS <- data_SRHS_long[1:1600,]
head(dataSRHS)
X <- cbind(dataSRHS$gender-1, dataSRHS$race == 2 | dataSRHS$race == 3,
dataSRHS$education == 4,dataSRHS$education == 5, dataSRHS$age-50,
(dataSRHS$age-50)^2/100)
Y <- dataSRHS$srhs
res <- long2matrices(dataSRHS$id, X = X, Y = Y)
```

`long2wide` *From data in the long format to data in the wide format*

Description

Function that transforms data in the long format to data in the wide format.

Usage

```
long2wide(data, nameid, namet, colx, coly, aggr = T, full = 999)
```

Arguments

`data`	matrix of data
`nameid`	name of the id column
`namet`	name of the t column
`colx`	vector of the names of the columns of the covariates
`coly`	vector of the names of the columns of the responses
`aggr`	if wide aggregated format is required
`full`	number to use for missing data

Value

`listid`	list of id for every unit
`listt`	list of the time occasions
`data_wide`	data in wide format
`XX`	array of the covariates
`YY`	array of the responses
`freq`	vector of the corresponding frequencies

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
# Example based on criminal data
# load criminal data
data(data_criminal_sim)
# consider only the first 1000 records to shorten time
out <- long2wide(data_criminal_sim[1:1000,], "id", "time", "sex",
c("y1","y2","y3","y4","y5","y6","y7","y8","y9","y10"), aggr = TRUE, full = 999)
```

---

matrices2long *From data in array format to data in long format*

Description

Function to convert data with array format in data with long format.

Usage

```
matrices2long(Y, X1 = NULL, X2 = NULL)
```

Arguments

Y	array of responses (n x TT x r)
X1	array of covariates (n x TT x nc1)
X2	array of covariates (n x TT x nc2)

Details

Y, X1 and X2 must have the same number of observations.

Value

Returns a data.frame with data in long format. The first column indicates the name of the unit identifier, and the second column indicates the time occasions.

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
### Example with data on self rated health

data(data_SRHS_long)
SRHS <- data_SRHS_long[1:1600,]

# Covariates

X <- cbind(SRHS$gender-1,
           SRHS$race == 2 | SRHS$race == 3,
           SRHS$education == 4,
           SRHS$education == 5,
           SRHS$age-50,
           (SRHS$age-50)^2/100)

# Responses

Y <- SRHS$srhs


res <- long2matrices(SRHS$id, X = X, Y = Y)

long <- matrices2long(Y = res$YY, X1 = res$XX)
```

MCbasic-class *Class* 'MCbasic'

Description

An S3 class object created by lmestMc function for the Markov chain (MC) model without covariates.

Value

lk	maximum log-likelihood
piv	estimate of initial probability vector
Pi	estimate of transition probability matrices
np	number of free parameters
aic	value of the Akaike Information Criterion for model selection
bic	value of the Bayesian Information Criterion for model selection
Fy	estimated marginal distribution of the response variable ats each time occasion
n	number of observations in the data
TT	number of time occasions

`modBasic`	model on the transition probabilities: default 0 for time-heterogeneous transition matrices, 1 for time-homogeneous transition matrices, 2 for partial time homogeneity based on two transition matrices one from 2 to (TT-1) and the other for TT
`sepiv`	standard errors for the initial probabilities
`sePi`	standard errors for the transition probabilities
`call`	command used to call the function
`data`	data frame given in input

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

`lmestMc`

`MCcov-class` *Class* `'MCcov'`

Description

An S3 class object created by `lmestMc` function for Markov chain (MC) model for categorical responses in long format with covariates.

Value

`lk`	maximum log-likelihood
`Be`	estimated array of the parameters affecting the logit for the initial probabilities
`Ga`	estimated array of the parameters affecting the logit for the transition probabilities
`np`	number of free parameters
`aic`	value of the Akaike Information Criterion (AIC) for model selection
`bic`	value of the Bayesian Information Criterion (BIC) for model selection
`n`	number of observations in the data
`TT`	number of time occasions
`seBe`	standard errors for `Be`
`seGa`	standard errors for `Ga`
`Piv`	estimate of initial probability matrix
`PI`	estimate of transition probability matrices
`call`	command used to call the function
`data`	data frame given in input

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

See Also

lmestMc

---

print *Print the output*

Description

Given the output, it is written in a readable form

Usage

```
## S3 method for class 'LMbasic'
print(x, ...)
## S3 method for class 'LMbasiccont'
print(x, ...)
## S3 method for class 'LMlatent'
print(x, ...)
## S3 method for class 'LMlatentcont'
print(x, ...)
## S3 method for class 'LMmanifest'
print(x, ...)
## S3 method for class 'LMmixed'
print(x, ...)
## S3 method for class 'MCbasic'
print(x, ...)
## S3 method for class 'MCcov'
print(x, ...)
## S3 method for class 'LMsearch'
print(x, modSel = "BIC",...)
```

Arguments

x	output from lmest,lmestCont,lmestMixed, and lmestMc
modSel	a string indicating the model selection criteria: "BIC" (default) for Bayesian Information Criterion and "AIC" for Akaike Information Criterion Criterion
...	further arguments passed to or from other methods

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

RLMSdat *Dataset about job satisfaction*

Description

Longitudinal dataset deriving from the Russia Longitudinal Monitoring Survey (RLMS) about job satisfaction measured by an ordinal variable at seven different occasions with five categories, 1 for "absolutely satisfied", 2 for "mostly satisfied", 3 for "neutral", 4 for "not very satisfied", and 5 for "absolutely unsatisfied".

Usage

```
data(RLMSdat)
```

Format

A data frame with 1718 observations on the following 7 variables.

- `IKSJQ` reported job satisfaction at the 1st occasion
- `IKSJR` reported job satisfaction at the 2nd occasion
- `IKSJS` reported job satisfaction at the 3rd occasion
- `IKSJT` reported job satisfaction at the 4th occasion
- `IKSJU` reported job satisfaction at the 5th occasion
- `IKSJV` reported job satisfaction at the 6th occasion
- `IKSJW` reported job satisfaction at the 7th occasion

Source

http://www.cpc.unc.edu/projects/rlms-hse, http://www.hse.ru/org/hse/rlms

References

Russia Longitudinal Monitoring survey, RLMS-HSE, conducted by Higher School of Economics and ZAO "Demoscope" together with Carolina Population Center, University of North Carolina at Chapel Hill and the Institute of Sociology RAS

Examples

```
data(RLMSdat)
```

RLMSlong *Dataset about job satisfaction*

Description

Longitudinal dataset in long format deriving from the Russia Longitudinal Monitoring Survey (RLMS) about job satisfaction measured by an ordinal variable at seven different occasions with five categories, 1 for "absolutely satisfied", 2 for "mostly satisfied", 3 for "neutral", 4 for "not very satisfied", and 5 for "absolutely unsatisfied".

Usage

```
data(RLMSlong)
```

Format

A data frame with 1718 observations on the following 7 variables.

`time` occasion of observation.

`id` subject id.

`rlms` see `RLMSdat`.

`value` reported job satisfaction at different time occasions coded as 1 for "absolutely satisfied", 2 for "mostly satisfied", 3 for "neutral", 4 for "not very satisfied", 5 for "absolutely unsatisfied".

Source

http://www.cpc.unc.edu/projects/rlms-hse, http://www.hse.ru/org/hse/rlms

References

Russia Longitudinal Monitoring survey, RLMS-HSE, conducted by Higher School of Economics and ZAO "Demoscope" together with Carolina Population Center, University of North Carolina at Chapel Hill and the Institute of Sociology RAS

Examples

```
data(RLMSlong)
```

se — *Standard errors*

Description

Function to compute standard errors for the parameter estimates.

Usage

```
se(est, ...)
## S3 method for class 'LMbasic'
se(est, ...)
## S3 method for class 'LMbasiccont'
se(est, ...)
## S3 method for class 'LMlatent'
se(est, ...)
## S3 method for class 'LMlatentcont'
se(est, ...)
```

Arguments

`est`	an object obtained from a call to `lmest` and `lmestCont`
`...`	further arguments

Value

Standard errors for estimates in `est` object.

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Examples

```
## Not run:

data("data_SRHS_long")
SRHS <- data_SRHS_long[1:2400,]

# Categories rescaled to vary from 1 (“poor”) to 5 (“excellent”)

SRHS$srhs <- 5 - SRHS$srhs

out <- lmest(responsesFormula = srhs ~ NULL,
            index = c("id","t"),
            data = SRHS,
            k = 3,
            modBasic = 1,
            out_se = FALSE)
```

```
out.se <- se(out)

out1 <- lmest(responsesFormula = srhs ~ NULL,
            index = c("id","t"),
            data = SRHS,
            k = 3,
            modBasic = 1,
            out_se = TRUE)
out1.se <- se(out1)

out2 <- lmest(responsesFormula = srhs ~ NULL,
             latentFormula =  ~
             I(gender - 1) +
             I( 0 + (race == 2) + (race == 3)) +
             I(0 + (education == 4)) +
             I(0 + (education == 5)) +
             I(age - 50) + I((age-50)^2/100),
             index = c("id","t"),
             data = SRHS,
             k = 2,
             paramLatent = "multilogit",
             start = 0)
out2.se <- se(out2)

require(mmm)
data(multiLongGaussian)

t <- rep(1:4, times  = max(multiLongGaussian$ID))
multiLongGaussian <- data.frame(t = t, multiLongGaussian)

out3 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                  index = c("ID", "t"),
                  data = multiLongGaussian,
                  k =3,
                  modBasic = 1,
                  tol = 10^-5)
out3.se <- se(out3)

out4 <- lmestCont(responsesFormula = resp1 + resp2 ~ NULL,
                   latentFormula = ~ X + time | X + time,
                   index = c("ID", "t"),
                   data = multiLongGaussian,
                   k = 3,
                   output = TRUE)
out4.se <- se(out4)

## End(Not run)
```

`search.model.LM` *Search for the global maximum of the log-likelihood*

Description

Function that searches for the global maximum of the log-likelihood of different models given a vector of possible number of states to try for.

The function is no longer maintained. Please look at lmestSearch **function**.

Usage

```
search.model.LM(version = c("basic","latent","manifest","basic.cont", "latent.cont"),
                kv, ..., nrep = 2, tol1 = 10^-5, tol2 = 10^-10,out_se = FALSE)
```

Arguments

version	model to be estimated ("basic" = basic LM model (est_lm_basic function); "latent" = LM model with covariates in the distribution of the latent process (est_lm_cov_latent function); "manifest" = LM model with covariates in the measurement model (est_lm_cov_maifest function),"basic.cont" = basic LM model for continuous outcomes (est_lm_basic_cont function); "latent.cont" = LM model for continuous outcomes with covariates in the distribution of the latent process (est_lm_cov_latent_cont function))
kv	vector of possible number of latent states
...	additional arguments to be passed based on the model to be estimated (see details)
nrep	number of repetitions of each random initialization
tol1	tolerance level for checking convergence of the algorithm in the random initializations
tol2	tolerance level for checking convergence of the algorithm in the last deterministic initialization
out_se	TRUE for computing information matrix and standard errors

Details

The function combines deterministic and random initializations strategy to reach the global maximum of the model log-likelihood. It uses one deterministic initialization (start=0) and a number of random initializations (start=1) proportional to the number of latent states. The tolerance level is set equal to 10^-5. Starting from the best solution obtained in this way, a final run is performed (start=2) with a default tolerance level equal to 10^-10.

Arguments in ... depend on the model to be estimated. They match the arguments to be passed to functions `est_lm_basic`, `est_lm_cov_latent`, `est_lm_cov_manifest`, `est_lm_basic_cont`, or `est_lm_cov_latent_cont`.

Value

out.single	output of each single model (as from `est_lm_basic`, `est_lm_cov_latent` or `est_lm_cov_manifest`) for each k in kv
aicv	value of AIC index for each k in kv

bicv	value of BIC index for each k in kv
lkv	value of log-likelihood for each k in kv

Author(s)

Francesco Bartolucci, Silvia Pandolfi, University of Perugia (IT), http://www.stat.unipg.it/bartolucci

Examples

```
## Not run:

# example for est_lm_basic
data(data_drug)
data_drug <- as.matrix(data_drug)
S <- data_drug[,1:5]-1
yv <- data_drug[,6]
n <- sum(yv)
# Search Basic LM model

res <- search.model.LM("basic", kv = 1:4, S, yv, mod = 1)
summary(res)


## End(Not run)
```

---

summary *Summary of LM fits*

Description

Summary methods

Usage

```
## S3 method for class 'LMbasic'
summary(object, ...)
## S3 method for class 'LMbasiccont'
summary(object, ...)
## S3 method for class 'LMlatent'
summary(object, ...)
## S3 method for class 'LMlatentcont'
summary(object, ...)
## S3 method for class 'LMmanifest'
summary(object, ...)
## S3 method for class 'LMmixed'
summary(object, ...)
## S3 method for class 'MCbasic'
summary(object, ...)
```

```
## S3 method for class 'MCcov'
summary(object, ...)
## S3 method for class 'LMsearch'
summary(object,...)
```

Arguments

`object`	output from `lmest`,`lmestCont`,`lmestMixed`, and `lmestMc`
`...`	further arguments passed to or from other methods

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

---

`summary.lmestData` *Summary and plot of* `lmestData`

Description

Methods for `lmestData` object providing basic descriptive statistics (summary) and plots.

Usage

```
## S3 method for class 'lmestData'
summary(object, type = c("all","cross", "year"),
       dataSummary = c("all", "responses", "manifest", "initial", "transition"),
        varType = rep("c", x$d), digits = getOption("digits"),
        maxsum = 10, maxobs = 20, ...)
## S3 method for class 'lmestData'
plot(x, typePlot = c("s", "sh"),
     dataPlots = c("all", "responses", "manifest", "initial", "transition"),
        ...)
## S3 method for class 'lmestData'
print(x, ...)
```

Arguments

`object`	an object of class `lmestData`
`x`	an object of class `lmestData`
`type`	type of summary to print. `all` prints a summary for each varaible, and a summary for each variables by time. `cross` prints a summary for each variable. `year` prints a summary for each variable by time. The summary is adapted according to `varType` (By default is set to `all`)

`dataSummary`	a string indicating whether summary is returned: `all` for the entire data, `responses` for the responses, `manifest` for covariates on the manifest distribution, `initial` for the covariate affecting the initial probabilities, and `transition` for the covariates affecting the transition probabilities. (By default is set to `all`)
`varType`	a string vector of lengh equal to the number of variables, "c" for continuous and "d" for discrete, indicating wich varaibles are continuos and which are discrete
`digits`	the number of significant digits
`maxsum`	an integer value indicating the maximum number of levels to print
`maxobs`	an integer value indicating the maximun number of observation in which the summary statistics are reported for each observation
`typePlot`	a string indicating the type of plot. "s" plots a scaterplot matrix. "sh" plots a scatterplot matrix with the histogram for each variable in the diagonal
`dataPlots`	a string indicating whether the plot is returned: `all` for the entire data, `responses` for the responses, `manifest` for covariates on the manifest distribution, `initial` for the covariate affecting the initial probabilities, `transition` for the covariates affecting the transition probabilities. (By default is set to all)
`...`	further arguments

Author(s)

Francesco Bartolucci, Silvia Pandolfi, Fulvia Pennoni, Alessio Farcomeni, Alessio Serafini

Index