Package ‘GenHMM1d’

January 21, 2021

Type Package

Title Goodness-of-Fit for Univariate Hidden Markov Models

Version 0.1.0


License GPL-3

Encoding UTF-8

LazyData true

Imports actuar, EnvStats, extraDistr, ggplot2, matrixcalc, parallel, reshape2, rmutil, ssdtools, VaRES, VGAM

Depends doParallel, foreach, stats

Suggests gamlss.dist, GeneralizedHyperbolic, gld, GLDEX, sgt, skewt, sn, stabdist

RoxygenNote 7.1.1

NeedsCompilation no

Author Bouchra R. Nasri [aut, cre, cph], Mamadou Yamar Thioub [aut, cph]

Maintainer Bouchra R. Nasri <bouchra.nasri@umontreal.ca>

Repository CRAN

Date/Publication 2021-01-21 08:20:03 UTC

R topics documented:

CDF ................................................................. 2
distributions .................................................... 3
ES ................................................................. 3
EstHMMGen ....................................................... 4
ForecastHMMCdf ............................................... 6
ForecastHMMconfint .......................................... 7
CDF

Cumulative distribution function

Description

This function computes the cumulative distribution function (cdf) of a univariate distribution

Usage

CDF(family, y, param, size = 0)

Arguments

family
distribution name; run the function distributions() for help
y
observations
param
parameters of the distribution; (1 x p)
size
additional parameter for some discrete distributions; run the command distributions() for help

Value

f
cdf
distributions

The names and descriptions of the univariate distributions

Description

This function allows the users to find the details on the available distributions.

Usage

distributions()

Value

No returned value, allows the users to know the different distributions and parameters

ES

Expected shortfall function

Description

This function compute the expected shortfall of an univariate distribution

Usage

ES(family, p, param, size = 0, Nsim = 25000)

Arguments

- **family**
  - distribution name; run the function distributions() for help
- **p**
  - value (1 x 1) at which the expected shortfall needs to be computed; between 0 and 1; (e.g 0.01, 0.05)
- **param**
  - parameters of the distribution; (1 x p)
- **size**
  - additional parameter for some discrete distributions; run the command distributions() for help
- **Nsim**
  - number of simulations

Value

- **es**
  - expected shortfall
Examples

family = "gaussian"

theta = matrix(c(-1.5, 1.7),1,2) ;
es = ES(family, (0.01), theta)
print('Expected shortfall : ')
print(es$es)

EstHMMGen  

Estimation of univariate hidden Markov model

Description

This function estimates the parameters from a univariate hidden Markov model

Usage

EstHMMGen(
  y,
  reg,
  family,
  start = 0,
  max_iter = 10000,
  eps = 0.001,
  graph = 0,
  size = 0,
  theta0 = 0
)

Arguments

y  observations; (n x 1)
reg number of regimes
family distribution name; run the function distributions() for help
start starting parameters for the estimation; (1 x p)
max_iter maximum number of iterations of the EM algorithm; suggestion 10000
eps precision (stopping criteria); suggestion 0.001.
graph 1 for a graph, 0 otherwise (default); only for continuous distributions
size additional parameter for some discrete distributions; run the command distributions() for help
theta0 initial parameters for each regimes; (r x p)
Details

Value

theta estimated parameters; (r x p)
Q estimated transition matrix; (r x r)
eta conditional probabilities of being in regime k at time t given observations up to
time t; (n x r)
lambda conditional probabilities of being in regime k at time t given all observations; (n
x r)
U matrix of Rosenblatt transforms; (n x r)
cvm cramer-von-Mises statistic for goodness-of-fit
W pseudo-observations that should be uniformly distributed under the null hypoth-
esis
LL log-likelihood
nu stationary distribution
AIC Akaike information criterion
BIC Bayesian information criterion
CAIC consistent Akaike information criterion
AICcorrected Akaike information criterion corrected
HQC Hannan-Quinn information criterion
stats empirical means and standard deviation of each regimes using lambda
pred_l estimated regime using lambda
pred_e estimated regime using eta
runs_l estimated number of runs using lambda
runs_e estimated number of runs using eta

Examples

family = "gaussian"
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2); theta = matrix(c(-1.5, 1.7, 1, 1),2,2);
sim = SimHMMGen(Q, family, theta, 10)$SimData;
est = EstHMMGen(y=sim, reg=2, family=family)
ForecastHMMCdf

**Forecasted cumulative distribution function of a univariate HMM at times n+k1, n+k2,....**

**Description**

This function computes the forecasted cumulative distribution function of a univariate HMM for multiple horizons, given observations up to time n.

**Usage**

```
ForecastHMMCdf(y, family, theta, Q, eta, k = 1, graph = 0)
```

**Arguments**

- `y` points at which the cdf function is computed
- `family` distribution name; run the function distributions() for help
- `theta` parameters; (r x p)
- `Q` probability transition matrix; (r x r)
- `eta` vector of the estimated probability of each regime at time n; (1 x r)
- `k` times of prediction.
- `graph` (0 or else) produce plots

**Value**

```
cdf values of the cdf function
```

**Examples**

```
family = "gaussian"
lb = -6
ub = 6

theta = matrix(c(-1.5, 1.7, 1, 1),2,2)
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2)
eta = c(0.96091218, 0.03908782)

forecastedhmmcdf = ForecastHMMCdf(y=seq(from=-6, to=6, by=0.1), family=family, theta= theta, Q=Q, eta=eta, k=c(1,5,10,20), graph=1)
```
**ForecastHMMconfint**

*Forecasted confidence interval of a univariate HMM at times n+k1, n+k2,....*

**Description**

This function computes the forecasted confidence interval of a univariate HMM for multiple horizons, given observations up to time n.

**Usage**

```r
ForecastHMMconfint(U, family, theta, Q, eta, k = 1)
```

**Arguments**

- `U`: values between 0 and 1
- `family`: distribution name; run the function `distributions()` for help
- `theta`: parameters; (r x p)
- `Q`: probability transition matrix; (r x r)
- `eta`: vector of the estimated probability of each regime at time n; (1 x r)
- `k`: prediction times (may be a vector of integers).

**Value**

- `qlow`: lower bound of the forecasted confidence interval
- `qhigh`: upper bound of the forecasted confidence interval

**Examples**

```r
family = "gaussian"
theta = matrix(c(-1.5, 1.7, 1, 1),2,2)
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2)
eta = c(0.96091218, 0.03908782)
forecastedhmmconfint = ForecastHMMconfint(U=c(0.1, 0.9), family, theta=theta, Q=Q,
eta=eta, k=c(1,2,3,4,5))
print("Forecasted confidence interval : ")
print(forecastedhmmconfint)
```
Expected shortfall (ES) of a univariate HMM at time n+k1, n+k2, ...

Description

This function computes the ES of a univariate HMM for multiple horizons, given observations up to time n

Usage

ForecastHMMES(U, family, theta, Q, eta, k = 1)

Arguments

U
value (1 x 1) at which the expected shortfall needs to be computed; between 0 and 1; (e.g. 0.01, 0.05)

family
distribution name; run the function distributions() for help

theta
parameters; (r x p)

Q
probability transition matrix; (r x r)

eta
vector of the estimated probability of each regime at time n; (1 x r)

k
prediction times (may be a vector of integers).

Value

es
expected shortfall (1 x horizon)

Examples

family = "gaussian"

theta = matrix(c(-1.5, 1.7, 1, 1),2,2)
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2)
eta = c(0.96091218, 0.03908782)

forecastedES = ForecastHMMES(U=c(0.01), family, theta=theta, Q=Q, eta=eta, k=c(1,2,3,4,5))
print("Forecasted expected shortfall : ")
print(forecastedES)
ForecastHMMeta

Predicted probabilities of regimes of a univariate HMM given a new observation

Description

This function computes the predicted probabilities of the regimes for a new observation of a univariate HMM, given observations up to time n.

Usage

ForecastHMMeta(ynew, family, theta, Q, eta)

Arguments

- `ynew`: the new observations
- `family`: distribution name; run the function distributions() for help
- `theta`: parameters; (r x p)
- `Q`: probability transition matrix; (r x r)
- `eta`: vector of the estimated probability of each regime at time n; (1 x r)

Value

- `etanew`: predicted probabilities of the regimes

Examples

```r
family = "gaussian"
theta = matrix(c(-1.5, 1.7, 1, 1),2,2)
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2)
eta = c(0.96091218, 0.03908782)
forecastedhmmeta = ForecastHMMeta(c(1.5), family, theta=theta, Q=Q, eta=eta)
print("Forecasted regime probabilities : ")
print(forecastedhmmeta)
```
ForecastHMMPdf

**Forecasted density function of a univariate HMM at time** \( n+k_1, n+k_2, \ldots \)

**Description**

This function computes the probability forecasted density function of a univariate HMM for multiple horizons, given observations up to time \( n \).

**Usage**

\[
\text{ForecastHMMPdf}(y, \text{family, theta, Q, eta, k = 1, graph = 0})
\]

**Arguments**

- \( y \): points at which the pdf function is computed
- \textit{family}: distribution name; run the function \text{distributions}() for help
- \textit{theta}: parameters; \((r \times p)\)
- \textit{Q}: probability transition matrix; \((r \times r)\)
- \textit{eta}: vector of the estimated probability of each regime at time \( n \); \((1 \times r)\)
- \textit{k}: prediction times (may be a vector of integers).
- \textit{graph}: \((0 \text{ or else})\) produce plots

**Value**

- \textit{pdf}: values of the pdf function

**Examples**

```r
family = "gaussian"

lb = -6
ub = 6

theta = matrix(c(-1.5, 1.7, 1, 1),2,2)
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2)
eta = c(0.96091218, 0.03908782)

forecastedhmmpdf = ForecastHMMPdf(y=seq(from=lb, to=ub, by=0.1), family=family, theta=theta, Q=Q, eta=eta, k=c(1,5,10,20), graph=1)
```

**ForecastHMMTCM**

**Tail conditional median (TCM) of a univariate HMM at time n+k1, n+k2, ...**

**Description**

This function computes the tail conditional median of a univariate HMM at multiple times, given observations up to time n.

**Usage**

`ForecastHMMTCM(U, family, theta, Q, eta, k = 1)`

**Arguments**

- **U**: value (1 x 1) between 0 and 1
- **family**: distribution name; run the function `distributions()` for help
- **theta**: parameters; (r x p)
- **Q**: probability transition matrix; (r x r)
- **eta**: vector of the estimated probability of each regime at time n; (1 x r)
- **k**: prediction times (may be a vector of integers).

**Value**

- **tcm**: tail conditional median (1 x horizon)

**Examples**

```r
family = "gaussian"
theta = matrix(c(-1.5, 1.7, 1, 1),2,2)
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2)
eta = c(0.96091218, 0.03908782)
fq=ForecastHMMTCM(U=c(0.01), family, theta=theta, Q=Q, eta=eta, k=c(1,2,3,4,5))
print("Forecasted tail conditional mean : ",fq)
```


### ForecastHMMVAR

**Value at risk (VAR) of a univariate HMM at time n+k1, n+k2, ...**

**Description**

This function computes the VAR of a univariate HMM for multiple horizons, given observations up to time n.

**Usage**

```
ForecastHMMVAR(U, family, theta, Q, eta, k = 1)
```

**Arguments**

- **U**: value (1 x 1) between 0 and 1
- **family**: distribution name; run the function `distributions()` for help
- **theta**: parameters; (r x p)
- **Q**: probability transition matrix; (r x r)
- **eta**: vector of the estimated probability of each regime at time n; (1 x r)
- **k**: prediction times (may be a vector of integers).

**Value**

- **var**: values at risk (1 x horizon)

**Examples**

```r
family = "gaussian"
theta = matrix(c(-1.5, 1.7, 1, 1),2,2)
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2)
eta = c(0.96091218, 0.03908782)
forecastedVAR = ForecastHMMVAR(U=c(0.01), family, theta, Q, eta, k=c(1,2,3,4,5))
print('Forecasted VAR : ')
print(forecastedVAR)
```
**GofHMMGen**

**Goodness-of-fit of univariate hidden Markov model**

**Description**

This function performs goodness-of-fit test of an univariate hidden Markov model

**Usage**

```r
GofHMMGen(
  y,
  reg,
  family,
  start = 0,
  max_iter = 10000,
  eps = 0.001,
  graph = 0,
  size = 0,
  n_sample = 100,
  n_cores = 1,
  useFest = 1
)
```

**Arguments**

- `y` observations
- `reg` number of regimes
- `family` distribution name; run the function `distributions()` for help
- `start` starting parameter for the estimation
- `max_iter` maximum number of iterations of the EM algorithm; suggestion 10000
- `eps` precision (stopping criteria); suggestion 0.0001.
- `graph` 1 for a graph, 0 otherwise (default); only for continuous distributions
- `size` additional parameter for some discrete distributions; run the command `distributions()` for help
- `n_sample` number of bootstrap samples; suggestion 1000
- `n_cores` number of cores to use in the parallel computing
- `useFest` 1 (default) to use the first estimated parameters as starting value for the bootstrap, 0 otherwise

**Value**

- `pvalue` pvalue of the Cramer-von Mises statistic in percent
- `theta` Estimated parameters; (r x p)
### Description

This function shows the graphs resulting from the estimation of a HMM model.

### Usage

`graphEstim(y, reg, theta, family, pred_l, pred_e)`

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>y</code></td>
<td>observations</td>
</tr>
<tr>
<td><code>reg</code></td>
<td>number of regimes</td>
</tr>
<tr>
<td><code>theta</code></td>
<td>estimated parameters; (r x p)</td>
</tr>
<tr>
<td><code>family</code></td>
<td>distribution name; run the function <code>distributions()</code> for help</td>
</tr>
<tr>
<td><code>pred_l</code></td>
<td>estimated regime using lambda</td>
</tr>
<tr>
<td><code>pred_e</code></td>
<td>estimated regime using eta</td>
</tr>
</tbody>
</table>

### Value

No returned value; produces figures of interest for the HMM model.
### Description

This function performs a gridsearch to find a good starting value for the EM algorithm. A good starting value for the EM algorithm is one for which all observations have strictly positive density (the higher the better).

### Usage

GridSearchS0(family, y, params, lbpdf = 0)

### Arguments

- **family**: distribution name; run the function distributions() for help
- **y**: observations
- **params**: list of six vectors named (p1, p2, p3, p4, p5, p6). Each corresponding to a parameter of the distribution (additionnal parameters will be ignored). For example:
  - params = list(p1=c(0.5, 5, 0.5), p2=c(1, 5, 1), p3=c(0.1, 0.9, 0.1), p4=c(1,1,1), p5=c(1,1,1), p6=c(1,1,1)) where p1 is the grid of value for the first parameter.
- **lbpdf**: minimal acceptable value of the density; (should be >= 0)

### Value

- **goodStart**: accepted parameter set

### Examples

```r
family = "gaussian"
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2) ; theta = matrix(c(-1.5, 1.7, 1, 1),2,2) ;
sim = SimHMMGen(Q, "gaussian", theta, 500, graph=0)$SimData ;
params = list(p1=c(-2, 2, 0.5), p2=c(1, 5, 1), p3=c(1, 1, 1), p4=c(1,1,1), p5=c(1,1,1), p6=c(1,1,1))
accepted_params = GridSearchS0(family, sim, params, 0)
```

```r
family = "gaussian"
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2) ; theta = matrix(c(-1.5, 1.7, 1, 1),2,2) ;
sim = SimHMMGen(Q, "gaussian", theta, 500, graph=0)$SimData ;
params = list(p1=c(-2, 2, 0.5), p2=c(1, 5, 1), p3=c(1, 1, 1), p4=c(1,1,1), p5=c(1,1,1), p6=c(1,1,1))
accepted_params = GridSearchS0(family, sim, params, 0.1)
```
### PDF  
*Probability density function*

**Description**
This function computes the probability density function (pdf) of a univariate distribution

**Usage**
PDF(family, y, param)

**Arguments**
- **family**: distribution name; run the function distributions() for help
- **y**: observations
- **param**: parameters of the distribution; (1 x p)

**Value**
- **f**: pdf

---

### QUANTILE  
*Quantile function*

**Description**
This function computes the quantile function of a univariate distribution

**Usage**
QUANTILE(family, p, param, size = 0)

**Arguments**
- **family**: distribution name; run the function distributions() for help
- **p**: values at which the quantile needs to be computed; between 0 and 1; (e.g. 0.01, 0.05)
- **param**: parameters of the distribution; (1 x p)
- **size**: additional parameter for some discrete distributions; run the command distributions() for help

**Value**
- **q**: quantile/V AR
Examples

```r
family = "gaussian"

Q = 1 ; theta = matrix(c(-1.5, 1.7),1,2) ;
quantile = QUANTILE(family, (0.01), theta)
print('Quantile : ')
print(quantile)
```

---

**SimHMMGen**

*Simulation of univariate hidden Markov model*

**Description**

This function simulates observation from a univariate hidden Markov model

**Usage**

```r
SimHMMGen(Q, family, theta, n, graph = 0)
```

**Arguments**

- **Q** transition probability matrix; (r x r)
- **family** distribution name; run the function distributions() for help
- **theta** parameters; (r x p)
- **n** number of simulated observations
- **graph** 1 for a graph, 0 otherwise (default); only for continuous distributions

**Details**

HMM observations simulation

**Value**

- **SimData** Simulated data
- **MC** Simulated Markov chain
- **Sim** Simulated Data for each regime

**Examples**

```r
family = "gaussian"
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2) ; theta = matrix(c(-1.5, 1.7, 1, 1), 2, 2) ;
sim = SimHMMGen(Q, family, theta, 500, 0)

family = "binomial"
```

```r
family = "gaussian"
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2) ; theta = matrix(c(-1.5, 1.7, 1, 1), 2, 2) ;
sim = SimHMMGen(Q, family, theta, 500, 0)

family = "binomial"
```
size = 5
Q = matrix(c(0.8, 0.3, 0.2, 0.7), 2, 2) ; thetaB = matrix(c(size, size, 0.2, 0.7), 2, 2) ;
simB = SimHMMGen(Q, family, thetaB, 500, graph=0)$SimData

SimMarkovChain  Markov chain simulation

Description
This function generates a Markov chain \( X(1), \ldots, X(n) \) with transition matrix \( Q \), starting from a state \( \eta_0 \) or the uniform distribution on \( 1, \ldots, r \).

Usage
SimMarkovChain(Q, n, eta0)

Arguments
- \( Q \): transition probability matrix
- \( n \): number of simulated vectors
- \( \eta_0 \): initial value in \( 1, \ldots, r \).

Value
- \( x \): Generated Markov chain

Snd1  Cramer-von Mises statistic for the goodness-of-fit test of the null hypothesis of a univariate uniform distribution over \([0,1]\)

Description
This function computes the Cramer-von Mises statistic \( S_n \) for goodness-of-fit of the null hypothesis of a univariate uniform distribution over \([0,1]\).

Usage
Snd1(U)

Arguments
- \( U \): vector of pseudos-observations (approximating uniform)

Value
- \( sta \): Cramer-von Mises statistic
Description

This function computes the tail conditional median of a univariate distribution

Usage

TCM(family, p, param, size = 0, Nsim = 25000)

Arguments

- **family**: distribution name; run the command distributions() for help
- **p**: (1 x 1) values between 0 and 1
- **param**: parameters of the distribution
- **size**: additional parameter for some discrete distributions
- **Nsim**: number of simulations

Value

tcm tail conditional median

Examples

```r
family = "gaussian"
Q = 1 ; theta = matrix(c(-1.5, 1.7),1,2) ;
tcm = TCM(family, (0.01), theta)
print('Tail conditional mean : ')
print(tcm$tcm)
```
Index

CDF, 2

distributions, 3

ES, 3
EstHMMGen, 4

ForecastHMMcdf, 6
ForecastHMMconfint, 7
ForecastHMMES, 8
ForecastHMMeta, 9
ForecastHMMpdf, 10
ForecastHMMTCM, 11
ForecastHMMVAR, 12

GofHMMGen, 13
graphEstim, 14
GridSearchS0, 15

PDF, 16

QUANTILE, 16

SimHMMGen, 17
SimMarkovChain, 18
Snd1, 18

TCM, 19