Package 'discretizeCtsDTMC'

June 20, 2020

Title Discretize a Continuous-valued Discrete-time Markov Chain

Version 0.0.0.9000

Description Package discretizeCtsDTMC creates a discrete approximation to a Markov process defined on a continuous state space in discrete time.
Once the state space is discretized, discretizeCtsDTMC provides tools to estimate the transition matrices and analyze the Markov process.
It is used to model a population of individuals, each following a continuous-state Markov process in discrete time.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

R topics documented:

`aaa-discretizeCtsDTMC` *Discretize a Continuous-state Discrete-time Markov Chain*

Description

`discretizeCtsDTMC` creates a discrete approximation to a Markov process defined on a continuous state space in discrete time. Once the state space is discretized, `discretizeCtsDTMC` provides tools to estimate the transition matrices and analyze the Markov process.

Note

We'll think of a better name later.

See Also

multinom function in nnet package for estimating the transition matrices. DTMCPack and markovchain for analyzing the discrete-time Markov model once the continuous state space is discretized.

---

cut_states *Discretize a Variable Defined on a Continuous State Space*

Description

cut_states transforms a variable defined on a continuous state space into a discrete variable.

Usage

```
cut_states(x_cts, breaks)
```

Arguments

x_cts	a numeric vector of observations from a continuous random variable with discontinuities in the distribution function.
breaks	a numeric vector of thresholds for allocating elements of the continuous state space to discrete state space.

Value

a categorical variable with states for each variable that correspond to elements in the continuous state space.

---

est_trans_mats *Estimate Transition Matrices*

Description

est_trans_mats estimates transition matrices from a discrete-state, discrete-time Markov process.

Usage

```
est_trans_mats(x, time_stamp, id, n_lags, Hessian = FALSE)
```

Arguments

x	a numeric vector of observations from a discrete random variable.
time_stamp	a vector of time stamps that correspond to the observations in x.
id	a vector of labels to identify different individuals in the population.
n_lags	an integer number of lags to define the order of the Markov process.
Hessian	an indicator to specify whether the Hessian matrices are returned for each column of the transition matrices (default is FALSE).

Value

trans_mats an array of transition matrices for the Markov process.

find_atoms *Find Points of Discontinuity in the CDF*

Description

find_atoms finds points of discontinuity in the CDF of a continuous random variable.

Usage

```
find_atoms(x)
```

Arguments

x	a numeric vector of observations from a continuous random variable with discontinuities in the distribution function.

Value

a numeric vector of points in the sample space at which there are discontinuities in the distribution function, which are sometimes referred to as "atoms".

forecast_distn *Forecast a Probability Distribution*

Description

forecast_distn calculates a forecast of a probability distribution for a population governed by a discrete-state, discrete-time Markov process.

Usage

```
forecast_distn(trans_mats, init_probs, n_ahead)
```

Arguments

trans_mats	an array of transition matrices for the Markov process.
init_probs	a numeric probability vector that defines the initial proportions of the population in each state.
n_ahead	an integer number of lags that defines the order of the Markov process.

Value

a n_ahead-row numeric matrix of probability vectors that define the forecasted proportion of the population in each state at each time.

sim_pop_MC *Simulate a Population of Individuals following a Markov Chain*

Description

sim_pop_MC draws a realization of a population of individuals, each following a discrete-time Markov process. It can be used for a parametric bootstrap procedure when the large sample properties may not hold.

Usage

```
sim_pop_MC(trans_mats, init_probs, n_ahead, n_ind)
```

Arguments

trans_mats	an array of transition matrices for the Markov process.
init_probs	a numeric probability vector that defines the initial proportions of the population in each state.
n_ahead	an integer number of lags that defines the order of the Markov process.
n_ind	an integer number of individuals in the cross-section.

Value

a data frame of realizations for individuals in the cross-section, including id labels for individuals and time stamps for each time period.

state_breaks *Divide a Continuous State Space into Intervals*

Description

state_breaks calculates a vector of thresholds for allocating elements of a continuous state space to a discrete state space

Usage

```
state_breaks(x)
```

Arguments

x	a numeric vector of observations from a continuous random variable with discontinuities in the distribution function.

Value

a numeric vector of thresholds for allocating elements of the continuous state space to a discrete state space.

test_fore_dev *Test for a Deviation from Forecasted Population*

Description

test_fore_dev tests for a deviation from a forecasted population by calculating the Kullback-Leibler divergence statistic and p-values from the quantiles of the chi-squared distribution.

Usage

```
test_fore_dev(x, time_stamp, id, out_probs)
```

Arguments

x	a numeric vector of observations of a discrete random variable from the out-of-sample forecast period.
time_stamp	a vector of time stamps that correspond to the observations in x.
id	a vector of labels to identify different individuals in the population.
out_probs	a n_ahead-row numeric matrix probability vector that defines the forecasted proportions of the population in each state.

Value

a data frame of Kullback-Leibler divergence statistics and p-values from the quantiles of the chi-squared distribution.

Index