Package ‘qLRT’

July 26, 2013

Type Package

Title Testing for the number of states in hidden Markov models.

Version 1.1

Date 2013-07-26

Author Florian Schwaiger

Maintainer Florian Schwaiger <schwaige@mathematik.uni-marburg.de>

Description This package implements needed functions to perform the test presented in Holzmann, Schwaiger (2012) to test for the number of states in hidden Markov models.

License GPL-2

Depends methods, sn, rootSolve, quadprog, stats, MASS, parallel,moments, MixSim

R topics documented:

qLRT-package *Testing for the number of states in hidden Markov models.*

Description

This package implements needed functions to perform the test presented in Holzmann, Schwaiger (2013) to test for the number of states in several hidden Markov models.

The most important function is `qlrt` (see `qlrt-methods`), which tests for the number of states of the involved hidden Markov chain for a given dataset (the observable part of the HMM) and a selected model. Further, for a given model class `full_mle-methods` calculates the MLE and `simulate-methods` simulates a dataset from the given model and parameters. To select a specific HMM, e.g. a Skew-Normal HMM, classes are implemented, see `hiddenMarkovModel-class` or `select_model`. To find out which models are implemented you can also use the function `list_models`.

Details

Package:	qLRT
Type:	Package
Version:	1.1
Date:	2013-07-26
License:	GPL-2
Depends:	methods, sn, rootSolve, quadprog, stats, MASS, parallel, moments, MixSim

Note

I would like to thank Matthias Eulert for very helpful comments concerning the task of parallel computation and for writing the function `getNumberOfCores`.

Author(s)

Florian Schwaiger <schwaige@mathematik.uni-marburg.de>

References

Holzmann, H., Schwaiger, F. (2013). Testing for the number of states in hidden Markov models.
Zucchini and MacDonald (2009). Hidden Markov Models for Time Series: An Introduction Using R.

Examples

```
#check which models are available with list_models
list_models() #or help("hiddenMarkovModel-class")

#list parameters of model 4
list_necessary_parameters(4)
```

```
#then create a model of type 4 and use this for testing as below

#an example for log-returns
data(daxData)

#specify a skew normal HMM with switching variance
#selected parameters wil be used as starting parameters during estimation
tpm0=matrix(c(0.9,0.05,0.05,0.05,0.9,0.05,0.05,0.05,0.9),byrow=TRUE,ncol=3)
modelStart=select_model(4,sigma=c(1,2,4),tpm=tpm0,mu=0,gamma=0)

#testing
## Not run: qlrt(dataset=daxData,model=modelStart)

#second example: Two-State-Poisson HMM
m3 = select_model(3,tpm=matrix(c(0.8,0.2,0.1,0.9),byrow=TRUE,ncol=2),lambda=c(1,4))
## Not run: d3 = simulate(object=m3,nsim=2000)$data
qlrt(d3,m3)
full_mle(d3,m3)
## End(Not run)
```

calc_weights-methods

Calculating weights of asymptotic mixture.

Description

For models 1-5, this function can be used to calculate the mixture weights of the asymptotic distribution of the test statistic. Since this distribution depends on the true model, its parameters have to be entered. In applications these are estimated.

Usage

```
calc_weights(object)
```

Arguments

`object`	An object of type `hiddenMarkovModel-class`, see also `select_model`.

Examples

```
m4=select_model(4,sigma=c(1,3,7),
tpm=matrix(c(0.9,0.05,0.05,0.05,0.9,0.05,0.05,0.05,0.9),byrow=TRUE,ncol=3),
mu=1,gamma=0.4)

calc_weights(m4)
```

daxData *Daily log-returns.*

Description

Daily log-returns of the German stock index DAX (2005 - 2012).

Usage

```
data(daxData)
```

Format

A Vector with 1763 entries.

Examples

```
data(daxData)
```

full_mle-methods *Estimating the MLE.*

Description

This function estimates for a given model and a dataset the MLE. The model which is intended be used (e.g. normal HMM) has to be specified as an object, see `select_model`. Further, the parameters of the selected model will be used as starting parameters for the optimisation.
In case of the Poisson HMM we use the function of Zucchini and Macdonald.

Usage

```
full_mle(dataset,model)
```

Arguments

`dataset`	The dataset as a vector.
`model`	An object of type `hiddenMarkovModel-class`, see also `select_model`.

References

Zucchini and MacDonald (2009). Hidden Markov Models for Time Series: An Introduction Using R.

Examples

```
list_necessary_parameters(1)
m1=select_model(1,sigma=1,tpm=
matrix(c(0.9,0.05,0.05,0.05,0.9,0.05,0.05,0.05,0.9),byrow=TRUE,ncol=3),
mu=c(1,4,8))

## Not run:
s1=simulate(m1,1000)
full_mle(dataset=s1$data,model=m1)

## End(Not run)
```

`getNumberOfCores` *Calculating the number of available cores of the CPU*

Description

This function can be used to find out how many cores are available. This is helpful, since the function `qlrt-methods` is parallelized.

Author(s)

Matthias Eulert

`hiddenMarkovModel-class` *Class for hiddenMarkovModels.*

Description

Hidden Markov Model class.

Objects from the Class

This class has six sub-classes, where each specifies another parametric family of a hidden Markov model. The following families/models are implemented (i.e. HMMs with the according state dependent distributions): Poisson, Normal with switching mu, Normal with switching sigma, Skew-Normal with switching sigma, t with switching location and Normal with both parameters switching. More details of each model (especially needed parameters) can be found below.

We recommend to create models (classes) using the function `select_model`.

Slots

Each model has the following three slots. Besides, the models have specific additional slots listed further below.

- `tpm`**:** Square matrix, represents the transition probability matrix.
- `p`**:** Vector of probabilities, which represents the stationary distribution of the tpm. If the function `select_model` is used to create an object of any provided HMM, p is automatically set as the stationary distribution of the given tpm.
- `m`**:** Numeric, represents the number of states of the HMM.

Methods

For each model the following generic functions are provided:

qlrt: quasi-likelihood ratio test (for a given dataset and model)

simulate: simulates a dataset of the given model

full_mle: calculates the MLE (for a given dataset and model)

calc_weights: calculates weights of the asymptotic mixture weights of the test statistic, for models 1-5 only

Extends

The classes

`pois_hmm`

`normal_hmm_switching_mu`

`t_hmm_switching_mu`

`normal_hmm_switching_sigma`

`skew_normal_hmm_switching_sigma`

extend the class `hiddenMarkovModel`.

Normal HMM switching mu:

The standard deviation is assumed to be equal in all states. Only the mean parameter of the Normal is state dependent. Additional slots are

`mu`**:** Vector of state dependent mean parameters.

`sigma`**:** Standard deviation of each state dependent distribution, i.e. one positive value has to be entered here. The name of the class is `normal_hmm_switching_mu`.

Normal HMM switching sigma:

The mean is assumed to be equal in all states. Only the scale parameter of the Normal is state dependent. Additional slots are

`mu`**:** Mean of each state dependent distribution.

`sigma`**:** Vector of state dependent standard deviations.

The name of the class is `normal_hmm_switching_sigma`.

Poisson HMM:

Additional slot is `lambda` specifiying the state dependent rate parameters of the Poisson distribution. The name of the class is `pois_hmm`.

Skew-Normal HMM switching sigma:

The mean and the skewness are assumed to be equal in all states. Only the scale parameter of the Skew-Normal is state dependent. We used a reparametrized version of the Skew-Normal, where the location, scale and skewness parameter are infact the mean, standard deviation and skewness of the distribution, see Azzalini (1985) 'A class of distributions which includes the normal ones'. Additional slots are

`mu`: Mean of each state dependent distribution.

`sigma`: Vector of state dependent standard deviations.

`gamma`: Skewness of each state dependent distribution.

The name of the class is `skew_normal_hmm_switching_sigma`.

t HMM switching loation:

The scale parameter and the degrees of freedom are assumed to be equal in all states. Only the location parameter of the t-distribution is state dependent. Additional slots are

`mu`: Vector of state dependent mean parameters.

`sigma`: Scale parameter of each state dependent distribution, i.e. one positive value has to be entered here.

`df`: Degrees of freedom of each state dependent distribution.

The name of the class is `t_hmm_switching_mu`.

Normal HMM:

Mean and scale are both state-dependent in this model.

`mu`: Vector of state dependent means.

`sigma`: Vector of state dependent standard deviations.

The name of the class is `normal_hmm_switching_both`.

See Also

`select_model`

Examples

```
tpm0 = matrix(c(0.9,0.05,0.05,0.05,0.9,0.05,0.05,0.05,0.9),byrow=TRUE,ncol=3)
select_model(1,sigma=1,tpm=tpm0,mu=c(1,4,8))
select_model(2,sigma=c(1,2,6),tpm=tpm0,mu=4)
select_model(3,lambda=c(1,5,12),tpm=tpm0)
select_model(4,sigma=c(1,3,7),tpm=tpm0,mu=1,gamma=0.4)
select_model(5,sigma=2,df=4,tpm=tpm0,mu=c(3,5,10))
select_model(6,sigma=c(1,0.5,1),tpm=tpm0,mu=c(-3,1,2.5))
```

`list_models` *Showing all available models.*

Description

This function can be used to find out which models are implemented for testing (see `qlrt-methods`), simulation (see `simulate-methods`) or estimation (see `full_mle-methods`).

Usage

```
list_models()
```

`list_necessary_parameters` *Showing all necessary parameters of a model.*

Description

This function explains shortly on the console which parameter of a special model have to be specified.

Usage

```
list_necessary_parameters(model_no)
```

Arguments

`model_no`	The number of the requested model, see `select_model`.

`price2logreturn` *Calculates log-returns.*

Description

This function calculates the corresponding logarithmic returns either for a vector of stock prices (use variable prices) or for a link directing to a csv file with prices in a column called Adj.Close (e.g. Yahoo!Finance).

Usage

```
price2logreturn(link = NA, prices = NA)
```

Arguments

`link`	A link (as a string) directing to csv file with prices in a column called Adj.Close with values from new to old.
`prices`	A vector of prices with values from old to new.

Value

The function returns a vector of log-returns in percent.

qlrt-methods *Performing the qLRT.*

Description

This function performs for a given model and dataset the test for the number of states of the HMM. The model to be used has to be specified as an object, see `hiddenMarkovModel-class`, `select_model` and `list_necessary_parameters`. Further, the parameters of the selected model will be used as starting parameters for the optimization.

For models with state dependent scale parameter the (quasi) likelihood can be replaced by a penalized version, see optional arguments penaltyConstant0 and penaltyConstant1 (see below for default values).

Usage

```
qlrt(dataset,model,...)
```

Arguments

`dataset` The observed dataset.

`model` An object, which represents the selected type of hidden Markov model, see `hiddenMarkovModel-class` and `select_model`. Its parameters are moreover used as starting points for the optimization.

`...` Several optional arguments:

lower An optional argument, which only can be used for models 2 and 4. When used it is the lower bound for the smallest scale parameter during estimation. This restriction can avoid the known unboundedness of the likelihood.

cores The computation is performed parallel. With this argument the number of used cores can be specified - if not specified all cores are used.

a0.in If the true weights of the asymptotic mixture are known (and in case of model 1-5), they can be entered here. Otherwise the weights are calculated with respect to the estimation under the null hypothesis.

penaltyConstant0 Optionally the (quasi) likelihood can be penalized for small scale paramter values to avoid unboundedness of the likelihood. This positive constant determines how strong the penalty is for the estimation under the null. The default value is zero (no penalization) for models 2 and 4 and 1/n for model 6. If a penalty is requested, we recommend using the constant 1/n, where n is the length of the dataset.

penaltyConstant1 This constant determines how strong the penalization is under the alternative (for model 2 and 4 default is 0, i.e. no penalization; for model 6 the default value is >0 and depends on the estimated hypothesis parameter). Here, allways two scale parameters under the alternative are around one estimate under the null. A deviation away from the estimate under the null is penalized (if requested).

Value

An object of type htest.

References

Holzmann, H., Schwaiger, F. (2013). Testing for the number of states in hidden Markov models.

Examples

```
#check which models are available with list_models
list_models() #or help("hiddenMarkovModel-class")

#list parameters of model 4
list_necessary_parameters(4)

#then create a model of type 4 and use this for testing as below

#an example for log-returns
data(daxData)

#specify a skew normal HMM with switching variance
#selected parameters wil be used as starting parameters during estimation
tpm0=matrix(c(0.9,0.05,0.05,0.05,0.9,0.05,0.05,0.05,0.9),byrow=TRUE,ncol=3)
modelStart=select_model(4,sigma=c(1,2,4),tpm=tpm0,mu=0,gamma=0)

#testing
## Not run: qlrt(dataset=daxData,model=modelStart)

#second example: Two-State-Poisson HMM
m3 = select_model(3,tpm=matrix(c(0.8,0.2,0.1,0.9),byrow=TRUE,ncol=2),lambda=c(1,4))
## Not run: d3 = simulate(object=m3,nsim=2000)$data
qlrt(d3,m3)
full_mle(d3,m3)
## End(Not run)
```

select_model *Selecting a model.*

Description

This function is needed to select a specific model and its according parameters. First of all each model corresponds to a number 1-6, which has to be selected via the variable `model_no`. To find out which models are available use the function `list_models`. Further, not all parameters are needed for each model, e.g. for a Poisson HMM one only needs the tpm and lambda, whereas for a skew normal HMM one needs the tpm, sigma, mu and gamma. For more details, which parameters are sufficient for which model use the function `list_necessary_parameters` or see `hiddenMarkovModel-class`.

Usage

```
select_model(model_no, tpm, p, sigma, mu, gamma, lambda, df)
```

Arguments

model_no
: A value 1-6.

  This specifies the type of model, see hiddenMarkovModel-class.

tpm
: Transition probability matrix, needed for simulation, but redundant for testing (see next argument).

p
: When the t.p.m. is entered, p is set automatically as the according stationary distribution. For testing the t.p.m. is not urgently needed. The parameter p is needed and enough (and used as starting parameter). Thus, for simulation one has to specify the t.p.m. (p is set automatically) and for testing one only has to specify p.

sigma
: State dependent scale parameters, dimension depends on the type of model, see hiddenMarkovModel-class or list_necessary_parameters.

mu
: For some models the location parameter, see hiddenMarkovModel-class for details.

gamma
: In skewed-model the skewness parameter, has to be in the interval (-0.92,0.92).

lambda
: State dependent parameters for Poisson HMMs.

df
: Degrees of freedom for t HMM.

Value

An object of a class corresponding to the selected model.

Examples

```
list_models()
tpm0 = matrix(c(0.9,0.05,0.05,0.05,0.9,0.05,0.05,0.05,0.9),byrow=TRUE,ncol=3)
select_model(1,sigma=1,tpm=tpm0,mu=c(1,4,8))
select_model(2,sigma=c(1,2,6),tpm=tpm0,mu=4)
select_model(3,lambda=c(1,5,12),tpm=tpm0)
select_model(4,sigma=c(1,3,7),tpm=tpm0,mu=1,gamma=0.4)
select_model(5,sigma=2,df=4,tpm=tpm0,mu=c(3,5,10))
select_model(6,sigma=c(1,0.5,1),tpm=tpm0,mu=c(-3,1,2.5))
```

---

show-methods *Output of HMMs on console.*

Description

Provides for each HMM an output on the console.

simulate-methods *Simulating Datasets.*

Description

In order to simulate datasets from a specific model (see hiddenMarkovModel-class or list_models) this function can be used. At first select a model with list_models and then simulate with this function a dataset.

Usage

```
simulate(object,nsim,seed = NULL, ...)
```

Arguments

object	An object of type hiddenMarkovModel-class, see also select_model.
nsim	Number of requested samples.
seed	Currently it is not implemented to set a seed, thus this argument has no functionality.
...	No further arguments.

Value

A list with 2 entries:

data	the simulated dataset
s	the (hidden) states of the Markov chain

Examples

```
#Poisson HMM
p1 = select_model(3,lambda=c(1,10,30),
tpm=matrix(c(0.9,0.05,0.05,0.05,0.9,0.05,0.05,0.05,0.9),byrow=TRUE,ncol=3))
## Not run: simulate(p1,1000)
```

stat_distr *Calculating the stationary distribution of a MC*

Description

This function can be used to calculate the stationary distribution of a given Markov chain (parameterized by a transition probability matrix mat). This function is not written by us (see reference), but very useful when working with the present type of models, and therefore added to the package.

Usage

```
stat_distr(mat)
```

Arguments

`mat`	The tpm of a Markov chain.

Value

A vector of probabilities, which is the stationary distribution of the Markov chain.

Author(s)

Zucchini and MacDonald (2009)

References

Zucchini and MacDonald (2009). Hidden Markov Models for Time Series: An Introduction Using R.

Index