Package 'HMMmlselect'

October 12, 2022

Type Package

Title Determine the Number of States in Hidden Markov Models via
Marginal Likelihood

Version 0.1.6

Author Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and S. C. Kou.

Maintainer Chu-Lan Michael Kao <chulankao@gmail.com>

Description Provide functions to make estimate the number of states for a hidden Markov
model (HMM) using marginal likelihood method proposed by the authors.
See the Manual.pdf file a detail description of all functions, and a detail tutorial.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports HiddenMarkov, mclust, mvtnorm, stats, MCMCpack, Rcpp (>=
0.12.10)

LinkingTo Rcpp

SystemRequirements C++11

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-05-03 05:00:02 UTC

R topics documented:

HMMmlselect-package   *Determine the number of states in hidden Markov models via marginal likelihood*

Description

(See the Manual.pdf file in "inst/extdata"" folder for a detail description of all functions, and a detail tutorial.)

This packages provides function to estimate the number of states in a Hidden Markov model (HMM) using the marginal likelihood method proposed by Chen, Fuh, Kao and Kou (2019+), which we would denoted as HMMml2017 afterward. HMMmlselect estimates the number of states, and PlotHMM plots. Other related functions are also provided.

Details

Package:	HMMmlselect
Type:	Package
Version:	1.0
Date:	2019-2-08
License:	GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Author(s)

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou

Maintainer: Chu-Lan Michael Kao <chulankao@gmail.com>

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

Examples

```
library(HMMmlselect)

# simulate a 25 observations HMM
obs = HMMsim ( n = 25 )$obs

# perform order selection and estimation
results = HMMmlselect ( y = obs, list(Kfits = c(2,3), boolUseMclust = FALSE) )

# visualize the results, see figure 1
PlotHMM ( y = obs, results )
```

---

check_para_validity   *check_para_validity*

Description

Auxiliary function that checks whether the parameter set is suitable for HMM/Gaussian mixture.

Usage

```
check_para_validity(parameters_in_matrix_form, bool_hmm)
```

Arguments

parameters_in_matrix_form
                 The parameters to be checked.

bool_hmm   1 for HMM. 0 for Gaussian mixture.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the logical value of whether the parameter set is suitable for HMM/Gaussian mixture.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

estimateNormalizingConst

estimateNormalizingConst

Description

Auxiliary function that approximate the normalizing constant. Used when approximating marginal likelihood.

Usage

```
estimateNormalizingConst(SampDens, boolHMM, dft, RIS, IS, NsmpResmp, llUn, Mclust = TRUE)
```

Arguments

SampDens	The sample and density
boolHMM	Number that controls the method.
dft	Degree of freedom used in the t-distribution.
RIS	RIS in the algorithm.
IS	IS in the algorithm.
NsmpResmp	Number of resampling used in the algorithm.
llUn	Unnormalized log likelihood.
Mclust	Auxiliary variable. Set to TRUE.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the approximated normalizing constant.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

---

find_importance_function

find_importance_function

Description

Auxiliary function that computes the importance weight used in approximate the normalizing constant.

Usage

```
find_importance_function(x, boolUseMclust)
```

Arguments

x	The data that the importance weight is computed on.
boolUseMclust	Auxiliary variable. Set to TRUE.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the importance weight used in approximated normalizing constant.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

---

find_index_local_region

find_index_local_region

Description

Auxiliary function that identify the local region for locally restricted importance sampling. Used in approximate the normalizing constant.

Usage

```
find_index_local_region(samples, EM_result, df_t)
```

Arguments

`samples`	The sample used to consruct region.
`EM_result`	The parameters.
`df_t`	Degree of freedom for the t-distrbution.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the local region for locally restricted importance sampling.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

HMMfit	*HMMfit*

Description

The following function performs (a) HMM fitting through the Expectation-Maximization al- gorithm (METHOD = 1), (b) HMM fitting through the Markov chain Monte Carlo algorithm (METHOD = 2), and (c) Gaussian mixture model fitting through the Markov chain Monte Carlo algorithm (METHOD = 3).

Usage

```
HMMfit(y, K, METHOD, optionalfit = list())
```

Arguments

`y`	The observed data.
`K`	The specified number of states of the underlying Markov chian.
`METHOD`	Integer value indicating the method of parameter estimation: (a) HMM fitting through the Expectation-Maximization al- gorithm (METHOD = 1), (b) HMM fitting through the Markov chain Monte Carlo algorithm (METHOD = 2), and (c) Gaussian mixture model fitting through the Markov chain Monte Carlo algo- rithm (METHOD = 3)
`optionalfit`	Optional variables as a list. Possible options include: • Ngibbs: Number of samples when using MCMC. Default is 5000. • Burnin: Length of burnin period when using MCMC. Default is 5000. • Thin: Thinning parameter when using MCMC. Default is 10.

- Nstart: Number of starting points. Default is 50.
- verbose: Logic variable indicating pritting details or not. Default is `FALSE`.
- priors: Prior when using MCMC. Default is flat.

Details

See Manual.pdf in "inst/extdata" folder.

Value

This functions returns the fitting parameters of the observed data given the specified number of states.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

Examples

```
library(HMMmlselect)

# Example 1: use HMMfit to inference number of states
obs = HMMsim ( n = 200 )$obs
Nest = HMMfit( y = obs, K=3, METHOD = 1)
```

HMMfitting   *HMMfitting*

Description

Auxiliary function called from C++.

Usage

```
HMMfitting(tuningparameters)
```

Arguments

tuningparameters
         Carried values.

Value

Auxiliary function called from C++.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

HMMll *HMMll*

Description

Auxiliary function called from C++.

Usage

```
HMMll(tuningparameters)
```

Arguments

tuningparameters

Carried values.

Value

Auxiliary function called from C++.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

HMMmlestimate *HMMmlestimate*

Description

Auxiliary function that approximates the marginal likelihood.

Usage

```
HMMmlestimate(y, K, optionalfit = list())
```

Arguments

y	The data that the marginal likelihood is computed for.
K	Number of states.
optionalfit	Optional variables.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the approximated marginal likelihood.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

HMMmlselect   *HMMmlselect*

Description

This function computes the marginal likelihood of the HMM model with the observed data and various number of states, and choose the one with the highest marginal likelihood as the estimated number of states. The method in Chen et al. (2017) is used, which we will denote it as HMMml2017 afterward.

Usage

```
HMMmlselect(y, optionalfit = list())
```

Arguments

y   The observed data.

optionalfit   Optional variables as a list. Possible options include:

- Kfits: Possible numbers of states. The function will compute the marginal likelihood under each of these numbers, and choose the one with the highest values as the estimated number of states. Default is {2,3,...,6}
- RIS: Logical variable indicating whether to use the reciprocal importance sampling method in HMMml2017. Default is set to be `FALSE`.
- IS: Logical variable indicating whether to use the importance sampling method in HMMml2017. Default is set to be `TRUE`.
- RunParallel: Logical variable indicating whether to run using parallelization or not. Default is `TRUE`.
- boolUseMclust: Logical variable indicating whether to use the `Mclust` package. Default is set to be `FALSE`.
- priors: Lists of hyper parameters for the prior. Default is flat prior.
- boolHMM: Compute using marginal likelihood of HMM. Default is `TRUE`. If it is set to `FALSE`, then the program will use the marignal likelihood of mixture model instead.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns (1) the estimated number of hidden states using the marginal likelihood method, (2) the marginal likelihood values corresponding to 2, 3, ... number of hidden states, and (3) the fitted model parameters given the estimated number of hidden states.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

Examples

```
library(HMMmlselect)

# simulate a 25 observations HMM
obs = HMMsim ( n = 25 )$obs

# perform order selection and estimation
results = HMMmlselect ( y = obs, list(Kfits = c(2,3), boolUseMclust = FALSE) )

# visualize the results, see figure 1
PlotHMM ( y = obs, results )
```

HMMrepsim   *HMMrepsim*

Description

Auxiliary function called from C++.

Usage

```
HMMrepsim(tuningparameters)
```

Arguments

```
tuningparameters
```
                  Carried values.

Value

Auxiliary function called from C++.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

HMMsim   *HMMsim*

Description

This function simulates HMM with the observed data being conditionally Gaussian distributed given the underlying state.

Usage

```
HMMsim(n, optionalsim = list())
```

Arguments

n   The length of HMM to be simulated.

optionalsim   Optional variables as a list. Possible options include:

- Ksim: Number of states of the simulated HMM. Default is 3.
- P: The transition matrix of the underlying Markov chain. Default is a flat K-by-K matrix.
- mu: The mean of the observed data given each underlying state. Default is {1, 2, ..., K}.
- sigma: The standard deviation of the observed data given each underlying state. Default is {0.1, 0.1, ... 0.1}.
- pi: The distribution of the initial state. Default is an uniform distribution across all possible states.
- BoolWritetoFile: Logic variable indicating whether to write the result into file or not. Default is FALSE.
- Filenameoutput: The output file name for the simulated HMM. Default is "HMMtrace.txt".

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the sample of the simulated HMM.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

Examples

```
library(HMMmlselect)

# simulate a 25 observations HMM
obs = HMMsim ( n = 25 )$obs

# perform order selection and estimation
results = HMMmlselect ( y = obs, list(Kfits = c(2,3), boolUseMclust = FALSE) )

# visualize the results, see figure 1
PlotHMM ( y = obs, results )
```

HMMsimulate   *HMMsimulate*

Description

Auxiliary function called from C++.

Usage

```
HMMsimulate(tuningparameters)
```

Arguments

```
tuningparameters
                Carried values.
```

Value

Auxiliary function called from C++.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

```
ll_un_normalized_hmm_gm
```
ll_un_normalized_hmm_gm

Description

Auxiliary function that computes the unnormalized posterior density of HMM and Gaussian mixture.

Usage

```
ll_un_normalized_hmm_gm(paras, yobs, bool_hmm, priors)
```

Arguments

paras	The parameter for the HMM.
yobs	The observed data.
bool_hmm	1 if it's HMM. 0 if it's Gaussian mixture.
priors	Prior distribution.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the unnormalized posterior density of HMM and Gaussian mixture.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

```
logmvNormdensity
```
logmvNormdensity

Description

Auxiliary function that computes log-likelihood based on multivariate t-distribution. Used when approximating marginal likelihood.

Usage

```
logmvNormdensity(x, mu, sqrt_inv_sigma, lgsqrt_det_sigma, d, logconstnormal)
```

Arguments

x	Data to compute the log likelihood.
mu	Mean of the normal distribution
sqrt_inv_sigma	Corresponds to the standard deviation of the normal distribution.
lgsqrt_det_sigma	
	Corresponds to the determinant of the variance-covariance matrix.
d	Dimension of the normal distribution.
logconstnormal	Leading constant for normal density function.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the log-likelihood of the multivariate student-t distribution.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

---

logmvTdensity	*logmvTdensity*

Description

Auxiliary function that computes log-likelihood based on multivariate t-distribution. Used when approximating marginal likelihood.

Usage

```
logmvTdensity(x, mu, sqrt_inv_sigma, lgsqrt_det_sigma, df, d, logconstT)
```

Arguments

x	Data to compute the log likelihood.
mu	Mean of the t-distribution
sqrt_inv_sigma	Corresponds to the standard deviation of the t-distribution.
lgsqrt_det_sigma	
	Corresponds to the determinant of the variance-covariance matrix.
df	Degree of freedom of the t-distribution.
d	Dimension of the t-distribution.
logconstT	Leading constant for t-distribution density function.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the log-likelihood of the multivariate student-t distribution.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

---

multivariate_mixture_density

multivariate_mixture_density

Description

Auxiliary function that computes probability density of mixture normal/t-distribution. Used when approximating marginal likelihood.

Usage

```
multivariate_mixture_density(x, EM_result, df_t, logconstnormal, logconstT)
```

Arguments

x	Data to compute the log likelihood.
EM_result	Parameter as a dataset, including $p for probability weight, $Mu for the mean of each mixture component, and $Sigma for the standard deviation of each mixture component.
df_t	Degree of freedom of the t-distribution. If df_t==0, then it is treated as a normal distribution.
logconstnormal	Leading constant for normal density function. Used when df_t==0.
logconstT	Leading constant for t-distribution density function. Used when df_t>0.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the likelihood of the mixture normal/student-t distribution.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

PlotHMM	*PlotHMM*

Description

This function visualizes the state inference result using `HMMmlselect`. See the Manual.pdf under data folder for a figure example.

Usage

```
PlotHMM(y, results)
```

Arguments

y	The observed data.
results	The resulting output of state inference using `HMMmlselect`.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the graph with the original data and the inferenced states.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

Examples

```
library(HMMmlselect)

# simulate a 25 observations HMM
obs = HMMsim ( n = 25 )$obs

# perform order selection and estimation
results = HMMmlselect ( y = obs, list(Kfits = c(2,3), boolUseMclust = FALSE) )

# visualize the results, see figure 1
PlotHMM ( y = obs, results )
```

RobustBIC   *RobustBIC*

Description

This function estimates the number of states of the given HMM data using robust BIC criteria.

Usage

```
RobustBIC(y, optionalbic = list())
```

Arguments

y   The observed data.

optionalbic   Optional variables as a list. Possible options include:

- Kfits: Possible numbers of states. The function will compute the marginal likelihood under each of these numbers, and choose the one with the highest values as the estimated number of states. Default is {2,3,...,6}
- Nstart: The number of starting points for the robust BIC. Default is 50.
- verbose: Logic variable indicating pritting details or not. Default is FALSE.

Details

See Manual.pdf in "inst/extdata" folder.

Value

This function returns the estimated number of hidden states through minimizing the BIC, the BIC values of all the possible number of hidden states, and the fitted model parameters under the estimated number of hidden states under the BIC method.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2017+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

Examples

```
library(HMMmlselect)

# Example 1: use robust BIC to determine the order of HMM
obs = HMMsim ( n = 200 )$obs
resultsBIC = RobustBIC ( y = obs )
```

---

sample_mixture   *sample_mixture*

Description

Auxiliary function that simulates mixture normal/t-distribution. Used when approximating marginal likelihood.

Usage

```
sample_mixture(N, list_paras, df_t)
```

Arguments

N	The number of sample to be drawn.
list_paras	Parameter as a dataset, including $p for probability weight, $Mu for the mean of each mixture component, and $Sigma for the standard deviation of each mixture component.
df_t	Degree of freedom of the t-distribution. If df_t==0, then it is treated as a normal distribution.

Details

See Manual.pdf in "inst/extdata" folder.

Value

It returns the sample of the simulated mixture normal/student-t distribution.

References

Yang Chen, Cheng-Der Fuh, Chu-Lan Kao, and Samuel Kou (2019+) "Determine the number of states in hidden markov models via marginal likelihood." Submitted.

Index