Package ‘BayesMultiMode’

October 12, 2022

Type Package

Title Testing and Detecting Multimodality using Bayesian Methods

Version 0.1.1

Description The testing approach works in two stages. First, a mixture distribution is fitted on the data using a Sparse Finite Mixture (SFM) Markov chain Monte Carlo (MCMC) algorithm following Malsiner-Walli, Frühwirth-Schnatter and Grün (2016) <doi:10.1007/s11222-014-9500-2>). The number of mixture components does not have to be specified; it is estimated simultaneously with the mixture weights and components through the SFM approach. Second, the resulting MCMC output is used to calculate the number of modes and their locations following Basturk, Hoogerheide and van Dijk (2021) <doi:10.2139/ssrn.3783351>. Posterior probabilities are retrieved for both of these quantities providing a powerful tool for mode inference. Currently the package supports a flexible mixture of shifted Poisson distributions (see Basturk, Hoogerheide and van Dijk (2021) <doi:10.2139/ssrn.3783351>).

License GPL (>= 3)

Imports ggpubr, dplyr, tidyr, ggplot2, stringr, ggh4x, magrittr, gtools, Rdpack

RdMacros Rdpack

Encoding UTF-8

LazyData true

URL `https://github.com/paullabonne/BayesMultiMode`

BugReports `https://github.com/paullabonne/BayesMultiMode/issues`

NeedsCompilation no

RoxygenNote 7.1.2

Author Nalan Baştürk [aut],
Jamie Cross [aut],
Peter de Knijff [aut],
Lennart Hoogerheide [aut],
Paul Labonne [aut, cre],
Herman van Dijk [aut]

Maintainer Paul Labonne <`paul.labonne@bi.no`>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2022-10-12 08:32:59 UTC

R topics documented:

`bayes_mode` *Mode inference using post-processed SFM MCMC draws.*

Description

Computes the number of modes, their locations and posterior probabilities.

Usage

```
bayes_mode(theta_draws, y, mixt = "shifted_poisson")
```

Arguments

`theta_draws`	a (M x 3xJb) matrix. Output of 'sfm_mcmc_spmix()' giving MCMC parameter draws after burn-in and discarding empty components.
`y`	(a vector of integers) Observations used to fit the model.
`mixt`	(a string) giving the mixture distribution. Default is "shifted_poisson".

Value

A list containing:

- Prob_unimod : Posterior probability of unimodality. (1-Prob_unimod) is equal to the posterior probability of multimodality.
- table_nb_modes : Possible number of modes and posterior probability for each of those.
- table_locations : Possible locations of modes and posterior probability for each of those.
- A list of graphs showing :
  - 1: The posterior probability of multimodality;
  - 2: Possible number of modes and posterior probability for each of those;
  - 3: Possible locations of modes and posterior probability for each of those;

References

Basturk N, Hoogerheide LF, Dijk Hv (2021). "Bayes estimates of multimodal density features using DNA and Economic Data." *SSRN Electronic Journal*. ISSN 1556-5068, doi: 10.2139/ssrn.3783351, `https://www.ssrn.com/abstract=3783351`.

Examples

```
# Example with simulated data ================================================
#set seed for random number generation
set.seed(1)

# Set the parameters for drawing from a two-component shifted Poisson mixture
p1 = 0.3
p2 = 1-p1
kap1 = 3
kap2 = 0
lam1 = 1
lam2 = 0.5
length_data = 70

# Generate data
y <- c(rpois(length_data*p1, lam1)+kap1, rpois(length_data*p2, lam2)+kap2)

# Set parameters for the SFM MCMC estimation
M = 1000 # Number of MCMC iterations
Jmax = 4 # Maximum number of mixture components

# Estimation with SFM MCMC
sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)

# Proportion of draws burned in
S = 0.5

# Post processing
post_sfm_mcmc = post_sfm_mcmc(sfm_mcmc, S=S)

# Mode inference
bayes_mode(post_sfm_mcmc$theta_draws_slim,y)

# Example with DNA data =====================================================

y = d4z4
M = 5000 # Number of MCMC iterations
Jmax = 10 # Maximum number of mixture components

# Estimation with SFM MCMC

sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)
# Proportion of draws burned in
S = 0.5

# Mode inference
```

```
bayes_mode(post_sfm_mcmc$theta_draws_slim,y)
```

`d4z4` *DNA data*

Description

Macrosatellite repeat D4Z4 (4q). Locus (hg18): 4q35.2; Unit (kb): 3.3; Encoded product: DUX4.

Usage

```
d4z4
```

Format

A vector of counts with 410 elements.

References

Schaap M, Lemmers RJ, Maassen R, van der Vliet PJ, Hoogerheide LF, van Dijk HK, Basturk N, de Knijff P, van der Maarel SM (2013). "Genome-wide analysis of macrosatellite repeat copy number variation in worldwide populations: evidence for differences and commonalities in size distributions and size restrictions." *BMC Genomics*, **14**(1), 143. ISSN 1471-2164, doi: 10.1186/1471216414143.

`dspmix` *Density of a mixture of shifted Poisson distributions*

Description

Density function of a mixture of shifted Poisson distributions.

Usage

```
dspmix(x, p, lambda, kappa)
```

Arguments

`x`	(an integer) Observation at which the density is evaluated.
`p`	(a vector) Mixture weights.
`lambda`	(a vector) Lambda parameter for each component.
`kappa`	(a vector of integers) Kappa parameter for each component.

Value

Returns the density evaluated at the observation.

References

Basturk N, Hoogerheide LF, Dijk Hv (2021). “Bayes estimates of multimodal density features using DNA and Economic Data.” *SSRN Electronic Journal*. ISSN 1556-5068, doi: 10.2139/ssrn.3783351, `https://www.ssrn.com/abstract=3783351`.

Examples

```
# a three-component mixture
p = c(0.1,0.5,0.4)
lambda = c(1,2,3)
kappa= c(0,5,1)
dspmix(4,p,lambda,kappa)
dspmix(2,p,lambda,kappa)
dspmix(10,p,lambda,kappa)
```

`plots_mcmc` *Plots SFM MCMC output.*

Description

Show plots of the MCMC estimation.

Usage

```
plots_mcmc(sfm_mcmc, S)
```

Arguments

`sfm_mcmc`	a list. Output of ‘sfm_mcmc_spmix()‘ containing the parameter draws from the posterior distribution at each MCMC iteration.
`S`	(a number between 0 and 1) The first S*M draws will be discarded as a burn-in. M is the total number of MCMC iterations.

Value

A list showing :

- 1: Trace plots of pre-processed draws;
- 2: The posterior distribution of each parameter after post-processing;
- 3: The posterior probability of the number of components after post-processing;
- 4: Estimated mixture density after post-processing at each iteration and histogram of the data.

Examples

```
# Example with simulated data ================================================
#set seed for random number generation
set.seed(1)

# Set the parameters for drawing from a two-component shifted Poisson mixture
p1 = 0.3
p2 = 1-p1
kap1 = 3
kap2 = 0
lam1 = 1
lam2 = 0.5
length_data = 70

# Generate data
y <- c(rpois(length_data*p1, lam1)+kap1, rpois(length_data*p2, lam2)+kap2)

# Set parameters for the SFM MCMC estimation
M = 1000 # Number of MCMC iterations
Jmax = 4 # Maximum number of mixture components

# Estimation with SFM MCMC
sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)

# Generating plots

# Proportion of draws burned in
S = 0.5

plots_mcmc(sfm_mcmc, S=S)

# Example with DNA data =====================================================

y = d4z4
M = 5000 # Number of MCMC iterations
Jmax = 10 # Maximum number of mixture components

# Estimation with SFM MCMC
sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)

# Generating plots

# Proportion of draws burned in
S = 0.5

plots_mcmc(sfm_mcmc, S=S)
```

`post_sfm_mcmc` *Simple post-processing of SFM MCMC output.*

Description

Gives a matrix of MCMC parameters after burn-in and discarding empty components.

Usage

```
post_sfm_mcmc(sfm_mcmc, S)
```

Arguments

`sfm_mcmc`	a list. Output of ‘sfm_mcmc_spmix()‘ containing the parameter draws from the posterior distribution at each MCMC iteration.
`S`	(a number between 0 and 1) The first S*M draws will be discarded as a burn-in. M is the total number of MCMC iterations.

Value

A list containing:

- A (M x 3xJb) matrix. Returns theta_draws after burn-in (discarding) the S*M rows. M is the number of rows of theta_draws (number of MCMC iterations). Jb is the number of components which are non-empty in at least one of the draws.
- J_ne (an integer). The number of non-empty components.

Examples

```
# Example with simulated data ================================================
#set seed for random number generation
set.seed(1)

# Set the parameters for drawing from a two-component shifted Poisson mixture
p1 = 0.3
p2 = 1-p1
kap1 = 3
kap2 = 0
lam1 = 1
lam2 = 0.5
length_data = 70

# Generate data
y <- c(rpois(length_data*p1, lam1)+kap1, rpois(length_data*p2, lam2)+kap2)

# Set parameters for the SFM MCMC estimation
M = 1000 # Number of MCMC iterations
Jmax = 4 # Maximum number of mixture components

# Estimation with SFM MCMC
sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)

# Proportion of draws burned in
S = 0.5
```

```
# Post processing
post_sfm_mcmc = post_sfm_mcmc(sfm_mcmc, S=S)

# Example with DNA data =====================================================

y = d4z4
M = 5000 # Number of MCMC iterations
Jmax = 10 # Maximum number of mixture components

# Estimation with SFM MCMC

sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)
# Proportion of draws burned in
S = 0.5

# Post processing
post_sfm_mcmc = post_sfm_mcmc(sfm_mcmc, S=S)
```

`sfm_mcmc_spmix` *Estimation of a mixture of shifted Poisson distributions.*

Description

Bayesian estimation of a mixture of shifted Poisson distributions using a Sparse Finite Mixture MCMC algorithm.

Usage

```
sfm_mcmc_spmix(y, Jmax, M, prt = TRUE)
```

Arguments

`y`	(a vector of integers) Observations used to fit the model.
`Jmax`	(an integer) Maximum number of mixture components.
`M`	(an integer) Number of MCMC iterations.
`prt`	print intermediate of the MCMC estimation ? default = TRUE.

Value

A list containing:

- theta_draws : a (M x 3xJmax) matrix. Parameter draws from the posterior distribution at each MCMC iteration.
- p_draws : a (M x Jmax) matrix. Posterior draw of mixture weights at each MCMC iteration.
- kappa_draws : a (M x Jmax) matrix. Posterior draws of kappa at each MCMC iteration.
- lambda_draws : a (M x Jmax) matrix. Posterior draws of lambda at each MCMC iteration.

- snj : a (M x Jmax) matrix. Number of observations allocated to each components at each MCMC iteration.
- se0 : a vector of size M. Concentration parameter from (symmetric) Dirichlet distribution at each MCMC iteration.
- dist : a string indicating the distribution used in the mixture.
- y, Jmax and M, given as input.

References

Malsiner-Walli G, Fruhwirth-Schnatter S, Grun B (2016). "Model-based clustering based on sparse finite Gaussian mixtures." *Statistics and Computing*, **26**(1), 303–324. ISSN 1573-1375, doi: 10.1007/s1122201495002.

Basturk N, Hoogerheide LF, Dijk Hv (2021). "Bayes estimates of multimodal density features using DNA and Economic Data." *SSRN Electronic Journal*. ISSN 1556-5068, doi: 10.2139/ssrn.3783351, `https://www.ssrn.com/abstract=3783351`.

Examples

```
# Example with simulated data ================================================
#set seed for random number generation
set.seed(1)

# Set the parameters for drawing from a two-component shifted Poisson mixture
p1 = 0.3
p2 = 1-p1
kap1 = 3
kap2 = 0
lam1 = 1
lam2 = 0.5
length_data = 70

# Generate data
y <- c(rpois(length_data*p1, lam1)+kap1, rpois(length_data*p2, lam2)+kap2)

# Set parameters for the SFM MCMC estimation
M = 1000 # Number of MCMC iterations
Jmax = 4 # Maximum number of mixture components

# Estimation with SFM MCMC
sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)

# Example with DNA data =====================================================

y = d4z4
M = 5000 # Number of MCMC iterations
Jmax = 10 # Maximum number of mixture components

# Estimation with SFM MCMC
sfm_mcmc = sfm_mcmc_spmix(y=y,Jmax=Jmax, M=M)
```

Index