

Package ‘BayesMultiMode’

April 3, 2023

Type Package

Title Bayesian Mode Inference

Version 0.5.0

Description

A bayesian approach for mode inference which works in two steps. 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](https://doi.org/10.1007/s11222-014-9500-2)>. The number of mixture components does not have to be specified; the size of the mixture is estimated endogenously through the SFM approach. Second, the modes of the estimated mixture at each MCMC draw are retrieved using algorithms specifically tailored for mode detection. These estimates are then used to construct posterior probabilities for the number of modes, their locations and uncertainties, providing a powerful tool for mode inference.

License GPL (>= 3)

Imports assertthat,
bayesplot,
dplyr,
ggplot2,
ggpubr,
gtools,
magrittr,
MCMCglmm,
mvtnorm,
posterior,
sn,
stringr,
tidyr,
Rdpack,
scales

Depends R (>= 3.4.0)

Suggests testthat (>= 3.0.0)

RdMacros Rdpack

Encoding UTF-8

LazyData true

URL <https://github.com/paullabonne/BayesMultiMode>

BugReports <https://github.com/paullabonne/BayesMultiMode/issues>

NeedsCompilation no

RoxygenNote 7.2.3

Config/testthat/edition 3

R topics documented:

bayes_estimation	<i>Bayesian estimation of mixture distributions</i>
------------------	---

Description

Gibbs samplers for sparse finite mixture Markov chain Monte Carlo (SFM MCMC) estimation.

Usage

```
bayes_estimation(
  data,
  K,
  dist,
  priors = list(),
  nb_iter = 2000,
  burnin = nb_iter/2,
  printing = TRUE
)
```

Arguments

data	Vector of observations
K	Maximum number of mixture components
dist	String indicating the distribution of the mixture components Currently supports "normal", "skew_normal", "poisson" and "shifted_poisson"
priors	List of priors; default is an empty list which implies the following priors : a0 = 1, A0 = 200, b0 = median(y), B0 = (max(y) - min(y))^2 (normal), D_xi = 1, D_psi = 1, (skew normal: B0 = diag(D_xi, D_psi)), c0 = 2.5, l0 = 1.1 (poisson), l0 = 5 (shifted poisson), L0 = 1.1/median(y), L0 = l0 - 1 (shifted poisson), g0 = 0.5, G0 = 100*g0/c0/B0 (normal), G0 = g0/(0.5*var(y)) (skew normal)
nb_iter	Number of MCMC iterations; default is 2000
burnin	Number of MCMC iterations used as burnin; default is nb_iter/2
printing	Showing MCMC progression ?

Value

A list of class ‘BayesMixture’ containing

- data - Same as argument
- dist_type - Type of the distribution (continuous or discrete)
- pars_names - Names of the mixture components’ parameters
- mcmc - Matrix of MCMC draws where the rows corresponding to burnin have been discarded
- mcmc_all - Original matrix of MCMC draws

References

Malsiner-Walli G, Frühwirth-Schnatter S, Grün 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](https://doi.org/10.1007/s1122201495002).

Frühwirth-Schnatter S, Pyne S (2010). “Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions.” *Biostatistics*, **11**(2), 317–336. ISSN 1465-4644, doi:[10.1093/biostatistics/kxp062](https://doi.org/10.1093/biostatistics/kxp062), [2023-02-22](https://doi.org/10.1093/biostatistics/kxp062).

Frühwirth-Schnatter S, Malsiner-Walli G (2019). “From here to infinity: sparse finite versus Dirichlet process mixtures in model-based clustering.” *Advances in Data Analysis and Classification*, **13**, 33–64.

Viallefont V, Richardson S, Peter J (2002). “Bayesian analysis of Poisson mixtures.” *Journal of Nonparametric Statistics*, **14**(1-2), 181–202.

Examples

```
# Example with galaxy data =====
set.seed(123)

# retrieve galaxy data
y = galaxy

# estimation
bayesmix = bayes_estimation(data = y,
                           K = 5, #not many to run the example rapidly
                           dist = "normal",
                           nb_iter = 500, #not many to run the example rapidly
                           burnin = 100)

# plot estimated mixture
# plot(bayesmix, max_size = 200)

# Example with DNA data =====
set.seed(123)

# retrieve DNA data
y = d4z4

# estimation
bayesmix = bayes_estimation(data = y,
                           K = 5, #not many to run the example rapidly
```

```

dist = "shifted_poisson",
nb_iter = 500, #not many to run the example rapidly
burnin = 100)

# plot estimated mixture
# plot(bayesmix, max_size = 200)

```

bayes_mode

Bayesian mode inference

Description

Estimates modes for each mcmc draws which are then used to compute posterior probabilities for the number of modes and their locations. The fixed-point algorithm of Carreira-Perpinan (2000) is used for Gaussian mixtures while the Modal EM algorithm of Li et al. (2007) is used for other continuous mixtures.

Usage

```

bayes_mode(
  BayesMix,
  rd = 1,
  tol_x = sd(BayesMix$data)/10,
  show_plot = FALSE,
  nb_iter = NULL
)

```

Arguments

BayesMix	An object of class ‘BayesMixture’
rd	Rounding parameter
tol_x	Tolerance parameter for distance in-between modes; default is sd(data)/10 where data is an element of BayesMix. If two modes are closer than tol_x, only the first estimated mode is kept. Not needed for mixtures of discrete distributions.
show_plot	Show density with estimated mode as vertical bars ?
nb_iter	Number of draws on which the mode-finding algorithm is run; default is NULL which means the algorithm is run on all draws.

Value

A list of class ‘BayesMode’ containing

- data - from BayesMix argument
- dist - from BayesMix argument
- dist_type - from BayesMix argument
- pars_names - from BayesMix argument
- modes - Matrix with a row for each draw and columns showing modes
- p1 - Posterior probability of unimodality
- tb_nb_modes - Matrix showing posterior probabilities for the number of modes
- table_location - Matrix showing the posterior probabilities for location points being modes

References

Carreira-Perpinan MA (2000). “Mode-finding for mixtures of Gaussian distributions.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(11), 1318–1323. ISSN 1939-3539, doi:10.1109/34.888716, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Li J, Ray S, Lindsay BG (2007). “A Nonparametric Statistical Approach to Clustering via Mode Identification.” *Journal of Machine Learning Research*, **8**, 1687-1723.

Examples

```
# Example with galaxy data =====
set.seed(123)

# retrieve galaxy data
y = galaxy

# estimation
bayesmix = bayes_estimation(data = y,
                           K = 5, #not many to run the example rapidly
                           dist = "normal",
                           nb_iter = 500, #not many to run the example rapidly
                           burnin = 100)

# mode estimation
bayesmode = bayes_mode(bayesmix)

# plot
# plot(bayesmode, max_size = 200)

# summary
# summary(bayesmode)

# Example with DNA data =====
set.seed(123)

# retrieve galaxy data
y = d4z4

# estimation
bayesmix = bayes_estimation(data = y,
                           K = 5, #not many to run the example rapidly
                           dist = "shifted_poisson",
                           nb_iter = 500, #not many to run the example rapidly
                           burnin = 100)

# mode estimation
bayesmode = bayes_mode(bayesmix)

# plot
# plot(bayesmode, max_size = 200)

# summary
# summary(bayesmode)
```

bayes_trace	Trace plots
-------------	-------------

Description

This is wrapper around the ‘mcmc_trace()’ function from package bayesplot.

Usage

```
bayes_trace(BayesMix, mcmc_vars = NULL, with_burnin = FALSE, ...)
```

Arguments

BayesMix	An object of class BayesMixture
mcmc_vars	Variables to plot; default is all the variable in the MCMC output
with_burnin	Plot all draws ?
...	Additional arguments passed to function ‘mcmc_trace()’ from the package bayesplot.

Value

A trace plot.

Examples

```
# Example with galaxy data =====
set.seed(123)

# retrieve galaxy data
y = galaxy

# estimation
bayesmix = bayes_estimation(data = y,
                             K = 5, #not many to run the example rapidly
                             dist = "normal",
                             nb_iter = 500, #not many to run the example rapidly
                             burnin = 100)

# trace plot
bayes_trace(bayesmix)
```

ct47	X chromosomal macrosatellite repeats ct47
------	---

Description

Repeat units that encode for a cancer testis antigen.
Locus (hg18): Xq24
Unit (kb): 4.8
Restriction enzyme: EcoRI
Encoded product : cancer testis antigen 47

Usage

ct47

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](https://doi.org/10.1186/1471216414143).

cyclone*Tropical cyclones lifetime maximum intensity*

Description

Dataset constructed using the International Best Track Archive for Climate Stewardship (IBTrACS). The distribution of tropical cyclones lifetime maximum intensity across the globe is known to be bimodal which has important implications for climate modelling.

Usage

cyclone

Format

A dataset with three columns showing the identification of the cyclone, its year of occurrence and its lifetime maximum intensity (LMI). LMI is calculated as the maximum wind speed for each cyclone with unit ks.

Source

<https://www.ncei.noaa.gov/products/international-best-track-archive>

References

Knapp KR, Kruk MC, Levinson DH, Diamond HJ, Neumann CJ (2010). "The International Best Track Archive for Climate Stewardship (IBTrACS): Unifying Tropical Cyclone Data." *Bulletin of the American Meteorological Society*, **91**(3), 363–376. ISSN 0003-0007, 1520-0477, doi:[10.1175/2009BAMS2755.1](https://doi.org/10.1175/2009BAMS2755.1), Publisher: American Meteorological Society Section: Bulletin of the American Meteorological Society, 2023-03-29.

Knapp KR, Diamond HJ, J.P. K, Kruk MC, Schreck CJ (2018). "International Best Track Archive for Climate Stewardship (IBTrACS) Project, Version 4." *NOAA National Centers for Environmental Information*. doi:[10.1175/2009BAMS2755.1](https://doi.org/10.1175/2009BAMS2755.1).

d4z4	<i>Autosomal macrosatellite repeats d4z4</i>
------	--

Description

Macrosatellite repeats D4Z4 in the subtelomere of chromosome 4q.
 Locus (hg18): 4q35.2
 Unit (kb): 3.3
 Restriction enzyme: EcoRI + HindIII/EcoRI + BlnI/XapI
 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](https://doi.org/10.1186/1471216414143).

discrete_MF	<i>Mode-finding algorithm for mixture of discrete distributions</i>
-------------	---

Description

Mode-finding algorithm for mixture of discrete distributions

Usage

```
discrete_MF(
  mcmc,
  data,
  pars_names,
  dist = "NA",
  pmf_func = NULL,
  type = "all",
  show_plot = FALSE
)
```


Arguments

mcmc	Vector of estimated mixture parameters
data	Vector of observations used for estimating the mixture
pars_names	Names of the mixture parameters; first element should correspond to the mixture proportions.
dist	String indicating the distribution of the mixture components. Currently supports "poisson" and "shifted_poisson"; default is "NA"; only use this argument if you have used Poisson and shifted Poisson distributions identical to the one used in the package.
pmf_func	Pmf of the mixture components associated with the mcmc draws (if mcmc estimation has not been carried out with BayesMultiMode); default is null
type	Type of modes, either unique or all (the latter includes flat modes); default is "all"
show_plot	If true show the data and estimated modes; default is false

Value

Vector of estimated modes

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](https://doi.org/10.1186/1471216414143).

Examples

```
# Example with the poisson distribution =====
lambda = c(0.1,10)
p = c(0.5,0.5)
params = c(eta = p, lambda = lambda)
pars_names = c("eta", "lambda")
dist = "poisson"

data = c(rpois(p[1]*1e3, lambda[1]),
         rpois(p[2]*1e3, lambda[2]))

modes = discrete_MF(params, data = data, pars_names = pars_names, dist = dist)

# Example with an arbitrary distribution =====
mu = c(20,5)
size = c(20,0.5)
p = c(0.5,0.5)
params = c(eta = p, mu = mu, size = size)
pars_names = c("eta", "mu", "size")

data = c(rnbinom(p[1]*1e3, mu = mu[1], size = size[1]),
         rnbinom(p[2]*1e3, mu = mu[2], size = size[2]))

pmf_func <- function(x, pars) {
  dnbinom(x, mu = pars["mu"], size = pars["size"])
}
```

```
modes = discrete_MF(params, data = data, pars_names = pars_names, pmf_func = pmf_func)
```

fixed_point

Modal fixed-point algorithm

Description

Algorithm for estimating modes in mixture of Normal distributions.

Usage

```
fixed_point(mcmc, data, pars_names, tol_x = sd(data)/10, show_plot = F)
```

Arguments

mcmc	Vector of estimated mixture parameters
data	Vector of observations used for estimating the mixture
pars_names	Names of the mixture parameters; first element should correspond to the mixture proportions; second to the mean; third to the standard deviation.
tol_x	Tolerance parameter for distance in-between modes; default is $\text{sd}(\text{data})/10$; if two modes are closer than tol_x , only the first estimated mode is kept.
show_plot	If true show the data and estimated modes; default is false

Value

Vector of estimated modes

References

Carreira-Perpinan MA (2000). "Mode-finding for mixtures of Gaussian distributions." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(11), 1318–1323. ISSN 1939-3539, doi:10.1109/34.888716, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Examples

```
mu = c(0,5)
sigma = c(1,2)
p = c(0.5,0.5)

data = c(rnorm(p[1]*100, mu[1], sigma[1]), rnorm(p[2]*100, mu[2], sigma[2]))
params = c(eta = p, mu = mu, sigma = sigma)
pars_names = c("eta", "mu", "sigma")
modes = fixed_point(params, data, pars_names)
```

galaxy

Galaxy series

Description

Velocity at which 82 galaxies in the Corona Borealis region are moving away from our galaxy, scaled by 1000.

Usage

```
galaxy
```

Format

An object of class `numeric` of length 82.

Source

<https://people.maths.bris.ac.uk/~mapjg/mixdata>

References

Richardson S, Green PJ (1997). “On Bayesian Analysis of Mixtures with an Unknown Number of Components.” *Journal of the Royal Statistical Society. Series B (Methodological)*, **59**(4), pp. 731–792. ISSN 00359246.

MEM

Modal EM algorithm (MEM)

Description

Algorithm to find modes in mixture of continuous distributions.

Usage

```
MEM(
  mcmc,
  data,
  pars_names,
  dist = "NA",
  pdf_func = NULL,
  tol_x = sd(data)/10,
  show_plot = FALSE
)
```

Arguments

mcmc	Vector of estimated mixture parameters
data	Vector of observations used for estimating the mixture
pars_names	Names of the mixture parameters; the first element of this vector should be the name of the mixture proportions. If you have used the skew normal of Azzalini, then the second element should correspond to the location, the third to the scale and the fourth to the shape.
dist	String indicating the distribution of the mixture components; default is "NA" Currently supports "normal" and "skew_normal"; not needed if pdf_func is provided
pdf_func	Pdf of the mixture components associated with the mcmc draws (if mcmc estimation has not been carried out with BayesMultiMode); default is null
tol_x	Tolerance parameter for distance in-between modes; default is sd(data)/10; if two modes are closer than tol_x, only the first estimated mode is kept.
show_plot	If true show the data and estimated modes; default is false

Value

Vector of estimated modes

References

Li J, Ray S, Lindsay BG (2007). "A Nonparametric Statistical Approach to Clustering via Mode Identification." *Journal of Machine Learning Research*, **8**, 1687-1723.

Azzalini A (1985). "A Class of Distributions Which Includes the Normal Ones." *Scandinavian Journal of Statistics*, **12**(2), 171–178. ISSN 0303-6898, Publisher: [Board of the Foundation of the Scandinavian Journal of Statistics, Wiley], [2023-03-13](#).

Examples

```
# Example with the skew normal =====
xi = c(0,6)
omega = c(1,2)
alpha = c(0,0)
p = c(0.8,0.2)
params = c(eta = p, xi = xi, omega = omega, alpha = alpha)
pars_names = c("eta", "xi", "omega", "alpha")
dist = "skew_normal"

data = c(su::rsn(p[1]*100, xi[1], omega[1], alpha[1]),
         su::rsn(p[2]*100, xi[2], omega[2], alpha[2]))

modes = MEM(params, data = data, pars_names = pars_names, dist = dist)

# Example with an arbitrary distribution =====
xi = c(0,6)
omega = c(1,2)
alpha = c(0,0)
nu = c(3,100)
p = c(0.8,0.2)
params = c(eta = p, mu = xi, sigma = omega, xi = alpha, nu = nu)
pars_names = c("eta", "mu", "sigma", "xi", "nu")
```

```
pdf_func <- function(x, pars) {
  sn::dst(x, pars["mu"], pars["sigma"], pars["xi"], pars["nu"])
}

data = c(sn::rst(p[1]*100, xi[1], omega[1], alpha = alpha[1], nu = nu[1]),
         sn::rst(p[2]*100, xi[2], omega[2], alpha = alpha[2], nu = nu[2]))

modes = MEM(params, pars_names = pars_names, data = data, pdf_func = pdf_func)
```

new_BayesMixture

*Creating a S3 object of class 'BayesMixture'***Description**

Function for creating an object of class 'bayes_estimation()' which can subsequently be used as argument in 'bayes_mode()'.

Usage

```
new_BayesMixture(
  mcmc,
  data,
  K,
  burnin,
  dist = "NA",
  pars_names,
  pdf_func = NULL,
  dist_type
)
```

Arguments

mcmc	A matrix of MCMC draws
data	A vector containing the data used for estimating the model and generating the MCMC draws
K	Number of mixture components
burnin	Number of draws to discard as burnin
dist	Distribution family of the mixture components supported by the package (e.g. "normal", "student", "skew_normal" or "shifted_poisson")
pars_names	Names of the mixture parameters; first element should correspond to the mixture proportions.
pdf_func	Pdf or pmf of the mixture components; this input is used only if dist_name is invalid or NULL.
dist_type	Either "continuous" or "discrete"

Value

A list of class 'BayesMixture' containing :

- data - Same as argument
- dist_type - Same as argument
- pars_names - Same as argument
- mcmc - Matrix of MCMC draws where the rows corresponding to burnin have been discarded
- mcmc_all - Original matrix of MCMC draws

plot.BayesMixture	<i>Plot an estimated mixture for a given number of draws with an histogram of the data.</i>
-------------------	---

Description

Plot an estimated mixture for a given number of draws with an histogram of the data.

Usage

```
## S3 method for class 'BayesMixture'
plot(x, max_size = 250, transparency = 0.1, ...)
```

Arguments

x	An object of class BayesMixture.
max_size	The number of MCMC draws to plot.
transparency	transparency of the density lines. Default is 0.1. Should be greater than 0 and below or equal to 1.
...	Not used.

plot.BayesMode	<i>Plot Bayesian mode estimates.</i>
----------------	--------------------------------------

Description

Plot Bayesian mode estimates.

Usage

```
## S3 method for class 'BayesMode'
plot(x, graphs = c("p1", "number", "loc"), ...)
```

Arguments

x	An object of class BayesMode.
graphs	which plot to show ? Default is all three c("p1", "number", "loc").
...	Not used.

summary.BayesMode	<i>Summary of Bayesian mode estimates.</i>
-------------------	--

Description

Summary of Bayesian mode estimates.

Usage

```
## S3 method for class 'BayesMode'  
summary(object, ...)
```

Arguments

object	An object of class BayesMode.
...	Not used.