Package: bayesmix (via r-universe)

August 22, 2024

Version 0.7-6

Date 2023-04-11

Title Bayesian Mixture Models with JAGS

Depends R (>= 2.10),

Imports graphics, stats, methods, rjags (>= 2.1.0), coda (>= 0.13)

Description Fits finite mixture models of univariate Gaussian distributions using JAGS within a Bayesian framework.

URL `https://statmath.wu.ac.at/~gruen/BayesMix/`

License GPL-2 | GPL-3

NeedsCompilation no

Author Bettina Gruen [aut, cre] (<`https://orcid.org/0000-0001-7265-4773`>), Martyn Plummer [ctb] (<`https://orcid.org/0000-0001-5130-6497`>)

Maintainer Bettina Gruen <`Bettina.Gruen@R-project.org`>

Repository CRAN

Date/Publication 2023-04-12 13:50:02 UTC

Contents

`BMMdiag` *Plot identifiability diagnostics for JAGSrun object*

Description

Two different plots are currently provided: a plot of different variables against each other and a plot of the same variable against its values in the other classes.

Usage

```
BMMdiag(object, which = 1:2, variables, ask = interactive(), fct1,
        fct2, xlim, ylim, auto.layout = TRUE, caption = NULL,
        main = "", ...)
```

Arguments

`object`	a `JAGSrun` object with model of class `BMMmodel`.
`which`	if only one of the plots is required, specify its number.
`variables`	if `variables` is missing, the names are taken from the `JAGSrun` object.
`ask`	prompt user before each page of plots
`fct1`	string: name of transformation function for variable on x-axis.
`fct2`	string: name of transformation function for variable on y-axis.
`xlim`	if no range for `xlim` is specified, a sensible range is taken.
`ylim`	if no range for `ylim` is specified, a sensible range is taken.
`auto.layout`	logical: if `TRUE` puts each of the two different plots on one figure.
`caption`	captions to appear above the plots.
`main`	title to each plot (in addition to the above 'caption').
`...`	further graphical parameters (see 'plot.xy' and 'par') may also be supplied as arguments.

Details

The plots help determining which variable will induce a unique labelling when taken for ordering of the segments and indicate if the model is overfitted by specifying too many segments.

Author(s)

Bettina Gruen

`BMMmodel` *Creates bugs model.*

Description

Creates the BUGS-model specification and return the values for the initialization, the prior specification and the observations.

Usage

```
BMMmodel(y, k, priors, inits = "initsFS", aprioriWeights = 1,
         no.empty.classes = FALSE, restrict = "none", ...)
```

Arguments

`y`	a numeric vector.
`k`	integer indicating the number of segments.
`priors`	specification of priors by a named list or a `BMMpriors` object.
`inits`	specification of initial values by a named list or string indicating the function to be called.
`aprioriWeights`	specification of prior of the a-priori weights. If `aprioriWeights` does not have length = k, there is an equal prior for the a-priori weights assumed.
`no.empty.classes`	logical: should it be prevented that empty classes arise during sampling.
`restrict`	one of "none", "mu", "tau".
`...`	further parameters for the function specified in `inits`.

Details

By default the function `initsFS` is called for generating initial values. Any other function specified by `inits` is assumed to have at least `x`, `k` and `restrict` as input parameters.

The parameter `restrict` indicates if a location-shift model ("tau"), a scale contaminated model ("mu") or a model where both variables vary over components shall be fitted.

If the logical `no.empty.classes` is `TRUE` there are observations added to the model that the classes are not empty. This signifies that the likelihood when sampling the class affiliations is changed thus that any data point which is sampled and is the last one in its class stays there.

Value

If `y` is specified an object of class `BMMmodel` is returned with components:

`inits`	named list.
`data`	named list.
`bugs`	text for .bug-file with prefix missing.

If `y` is missing an object of class `BMMsetup` is returned containing the parameter specifications. When `JAGScall` is called with this object as model argument, `BMMmodel` is called with `y` and the other parameters as input arguments before calling JAGS.

Author(s)

Bettina Gruen

See Also

`JAGSrun`, `initsFS`

Examples

```
data("fish", package = "bayesmix")
model <- BMMmodel(fish, k = 4, priors = list(kind = "independence",
                  parameter = "priorsFish", hierarchical = "tau"),
                  initialValues = list(S0 = 2))
model
```

`BMMposteriori` *Plot aposteriori probabilities of data points*

Description

Given a `JAGSrun` object with model of class `BMMmodel` the aposteriori probabilities are determined. If `plot=TRUE`, the resulting object of class `BMMposteriori` is plotted.

Usage

```
BMMposteriori(object, class, caption = NULL, plot = TRUE,
              auto.layout = TRUE, ...)
```

Arguments

`object`	a `JAGSrun` object with model of class `BMMmodel`.
`class`	a vector of integers indicating for which classes the posterior probabilities shall be plotted. The default is all.
`caption`	captions to appear above the plots.
`plot`	logical indicating if a plot shall be made.
`auto.layout`	logical: if `TRUE` puts all classes in the posterior probabilities plot on the same figure.
`...`	further graphical parameters may also be supplied as arguments.

Details

Given a `JAGSrun` object with model of class `BMMmodel` the a posteriori probabilities are calculated for the unique data points with respect to the components specified by `class`.

Value

There is a `BMMposteriori` object returned which is a list including the following components

`data`	vector of unique data points.
`post`	a matrix including the posteriori probability of the data points for each class.

Author(s)

Bettina Gruen

See Also

`plot.BMMposteriori`

`BMMpriors` *Create a 'BMMpriors' object*

Description

This function enables a comfortable creation of `BMMpriors` objects, which can be used for specifying the priors of a `BMMmodel`.

Usage

```
BMMpriors(specification, y, eps = 10^-16)
```

Arguments

`specification`	named list including `kind`, `parameter`, `hierarchical` and `mod`.
`y`	a numeric vector.
`eps`	a numeric value indicating the smallest value for flat priors.

Details

In `specification` `kind` can be used for specifying if an "independent" or a "conditionallyconjugate" prior shall be used. `parameter` can be a named list of values for the prior variables or a function name (e.g., "priorsUncertain", "priorsFish", "priorsRaftery"). `hierarchical` can be `NULL` or "tau" if a hierarchical prior shall be taken for τ. `mod` is a named list which provides the possibility to override the values from `parameter`.

Value

There is an object `BMMpriors` returned with components

`name`	vector indicating which kind of prior is specified and if it is an hierarchical prior and if appropriate with respect to which variable.
`var`	list of variables in the prior and their specified values.

Author(s)

Bettina Gruen

Examples

```
data("fish", package = "bayesmix")
priors <- BMMpriors(y = fish)
```

`darwin` *Differences in heights between plants*

Description

A numeric vector containing 15 observations of differences in heights between pairs of self-fertilized and cross-fertilized plants grown in the same condition.

Usage

```
data(darwin)
```

Format

A numeric vector of length 15.

Details

Darwin's data set contains two extremely small values. Therefore, this data set can be used for outlier modelling.

Source

Abraham, B. and G. Box (1978) Linear models and spurious observations. *Applied Statistics*, **27**, 131–8.

Examples

```
data("darwin", package = "bayesmix")
## Estimated sample density
plot(density(darwin[[1]]), ylim = c(0, 0.02), main = "Outlier modelling")
ss <- seq(-100, 100, by = 1)
## Normal density with estimated mean and sd of whole sample
lines(ss, dnorm(ss, mean = mean(darwin[[1]]), sd = sd(darwin[[1]])), col = "red")
## Normal density with estimated mean and sd of sample, where the 2
## extremely small values are removed
lines(ss, dnorm(ss, mean = mean(darwin[-c(1:2),1]),
      sd = sd(darwin[-c(1:2),1])), col = "green")
```

`fish` *Fish length data*

Description

A numeric vector containing 256 observations of fish lengths.

Usage

```
data(fish)
```

Format

A numeric vector of length 256.

Details

This data set can be used for modeling unobserved heterogeneity, as it can be assumed that underlying categories present in the data are the age groups to which the fish belong.

Source

D. M. Titterington, A. F. M. Smith and U.E. Makov (1985) *Statistical Analysis of Finite Mixture Distributions.* Wiley.

Examples

```
data("fish", package = "bayesmix")
ss <- seq(-3, 13, by = 0.01)
hist(fish[[1]], 20, freq = FALSE, main = "Fish data")
lines(ss, dnorm(ss, mean(fish[[1]]), sd(fish[[1]])), col = "red")
```

`initsFS` *Create initial values*

Description

Initial values for nodes are created after the suggestion in Sylvia Fruehwirth-Schnatter's book.

Usage

```
initsFS(x, k, restrict, initialValues = list())
```

Arguments

`x`	a numeric vector.
`k`	number of segments.
`initialValues`	additional initial values specifications.
`restrict`	one of "none", "mu", "tau".

Details

The initial values for μ are determined by the quantiles of the data, those for η give equal weight on each segment and those for τ are equal for all segments and estimated by the inverse of the IQR of the data divided by 1.34 and squared.

Value

A list with initial values for the parameter indicated by the name of the respective list element is returned.

Author(s)

Bettina Gruen

`JAGScall` *Call jags*

Description

Using functionality from package rjags the JAGS model is created and posterior draws are sampled.

Usage

```
JAGScall(model, y, prefix, control, ...)
```

Arguments

`model`	`JAGSmodel` object or output from `BMMmodel`.
`y`	a numeric vector.
`prefix`	character: prefix for .bug-file.
`control`	named list or `JAGScontrol` object.
`...`	additional parameters handed over to `BMMmodel`.

Value

Returns a `"JAGSrun"` object.

Author(s)

Bettina Gruen

See Also

`JAGSrun`

`JAGScontrol` *Control parameters for the sampling.*

Description

The information on the number of burn-in draws and monitored draws is specified. Furthermore, it includes the information which variables shall be monitored and possibly a seed and a random number generator (RNG).

Usage

```
JAGScontrol(variables, n.iter = 1000, thin = 1, burn.in = 0, seed,
            rng = c("base::Wichmann-Hill",
                    "base::Marsaglia-Multicarry",
                    "base::Super-Duper",
                    "base::Mersenne-Twister"))
```

Arguments

`variables`	names of variables which shall be monitored.
`n.iter`	number of monitored draws.
`thin`	thinning interval.
`burn.in`	number of discarded burn-in draws.
`seed`	integer setting the seed for the RNG.
`rng`	specification of RNG.

Value

An object of class `JAGScontrol` is returned which is a list containing the specified information.

Author(s)

Bettina Gruen

See Also

`JAGSrun`

Examples

```
control <- JAGScontrol(variables = "mu")
control
```

`JAGSrun` *MCMC sampling of Bayesian models*

Description

Calls jags for MCMC sampling.

Usage

```
JAGSrun(y, prefix = yname, model = BMMmodel(k = 2),
        control = JAGScontrol(variables = c("mu", "tau", "eta")),
        tmp = TRUE, cleanup = TRUE, ...)
```

Arguments

`y`	a numeric vector.
`prefix`	character: prefix for .bug-file.
`model`	object of class `JAGSmodel` or output from `BMMmodel`.
`control`	specification of control by a `JAGScontrol` object.
`tmp`	logical: shall the files be written in a temporary directory.
`cleanup`	logical: shall the created files be removed.
`yname`	a character string with the actual `y` argument name.
`...`	further parameters handed over to `BMMmodel` where it is used for the function specifying the initial values, e.g., `initsFS`.

Details

This function is a wrapper calling `JAGScall`.

Value

Returns a `JAGSrun` object with components

`call`	the matched call.
`results`	MCMC results obtained by calling functionality from package **rjags**.
`model`	a `JAGSmodel` object.
`variables`	vector containing the names of the monitored variables.
`data`	a numeric vector.

Author(s)

Bettina Gruen

See Also

`JAGScall`, `BMMmodel`, `initsFS`

Examples

```
data("fish", package = "bayesmix")
prefix <- "fish"
variables <- c("mu","tau","eta")
k <- 3
modelFish <- BMMmodel(k = k, priors = list(kind = "independence",
                      parameter = "priorsFish", hierarchical = "tau"))
controlFish <- JAGScontrol(variables = c(variables, "S"), n.iter = 100)
z1 <- JAGSrun(fish, prefix, model = modelFish, initialValues = list(S0 = 2),
              control = controlFish, cleanup = TRUE, tmp = FALSE)
zSort <- Sort(z1, "mu")
BMMposteriori(zSort)

data("darwin", package = "bayesmix")
prefix <- "darwin"
k <- 2
modelDarwin <- BMMmodel(k = k, priors = list(kind = "independence",
                        parameter = "priorsUncertain"),
                        aprioriWeights = c(1, 15),
                        no.empty.classes = TRUE, restrict = "tau")
z2 <- JAGSrun(darwin, prefix, model = modelDarwin, control =
              JAGScontrol(variables = variables, n.iter = 3000,
              burn.in = 1000), cleanup = TRUE, tmp = FALSE)
plot(z2, variables = "mu")
```

`plot.BMMposteriori` *Plot a posteriori probabilities of data points*

Description

Plot method for object of class `BMMposteriori`, typically called by `BMMposteriori`.

Usage

```
## S3 method for class 'BMMposteriori'
plot(x, caption, main = "", ...)
```

Arguments

`x`	a `BMMposteriori` object.
`caption`	captions to appear above the plots.
`main`	title to each plot-in addition to the above `caption`.
`...`	further graphical parameters may also be supplied as arguments.

Details

This function is called by `BMMposteriori` if `plot = TRUE`.

Author(s)

Bettina Gruen

See Also

`BMMposteriori`

`plot.JAGSrun` *Plot JAGSrun object*

Description

Plots `mcmc` chains of a `JAGSrun` object.

Usage

```
## S3 method for class 'JAGSrun'
plot(x, variables = NULL, trace = TRUE, density = TRUE,
     smooth = TRUE, bwf, num, xlim, auto.layout = TRUE,
     ask = interactive(), ...)
```

Arguments

`x`	a `JAGSrun` object.
`variables`	names of variables which shall be plotted. Default are all names of results except those with a column dimension larger than the number of classes `k`.
`trace`	plot trace of each variable.
`density`	plot density estimate of each variable.
`smooth`	draw a smooth line through trace plots.
`bwf`	bandwidth function for density plots.
`num`	if not all classes of a variable shall be plotted, a subset can be specified.
`xlim`	if not specified, the range of each variable over all classes is taken as default.
`auto.layout`	automatically generate output format.
`ask`	prompt user before each page of plots.
`...`	further arguments for `densityplot`.

Details

Adapted from `plot.mcmc`.

Currently only implemented for `JAGSrun` objects with model of class `BMMmodel`. Otherwise the default plot method for the `results` of the `JAGSrun` object is called (`plot.mcmc`).

Author(s)

Bettina Gruen

See Also

`plot.mcmc`, `BMMdiag`, `BMMposteriori`

`priors` *Create list of prior specifications*

Description

Given the data values for the priors are determined.

Usage

```
priorsFish(y, eps = 10^-16)
priorsRaftery(y)
priorsUncertain(y, eps = 10^-16)
```

Arguments

`y`	a numeric vector.
`eps`	a numeric value indicating the smallest value for flat priors.

Details

Values for the prior parameter `b0`, `B0`, `nu0` and `S0` are determined.

Value

There is a list returned with named components of the prior parameters.

Author(s)

Bettina Gruen

`randomPermutation` *Randomly permute segments for MCMC draws*

Description

Random permutation of segment labels for each draw in order to get a better estimate of the unrestricted likelihood.

Usage

```
randomPermutation(x)
```

Arguments

`x`	a `JAGSrun` object with model of class `BMMmodel`.

Details

The draws are permuted with respect to the different classes `k`.

Value

The input object with permuted results for each draw is returned.

Warning

Any variables where there are neither `k` different chains nor only one chain observed are dropped.

Author(s)

Bettina Gruen

`Sort` *Sort MCMC chains according to certain variables*

Description

Ascending sorting of results of `JAGSrun` object with model of class `BMMmodel` with respect to a given variable.

Usage

```
Sort(x, by = NULL)
```

Arguments

`x`	a `JAGSrun` object with model of class `BMMmodel`.
`by`	variable name according to which the segments shall be ordered.

Details

If `by` is not specified, the first variable in the corresponding vector of the `JAGSrun` object is taken.

Value

The input object with results sorted in ascending order according to the variable given in `by` is returned.

Warning

If there arise problems, the original object is returned with a warning.

Author(s)

Bettina Gruen

Index