Package ‘BMRMM’

January 20, 2025

Title An Implementation of the Bayesian Markov (Renewal) Mixed Models

Version 1.0.1

Author Yutong Wu [aut, cre],
Abhra Sarkar [aut]

Maintainer Yutong Wu <`yutong.wu@utexas.edu`>

Description The Bayesian Markov renewal mixed models take sequentially observed categorical data with continuous duration times, being either state duration or inter-state duration. These models comprehensively analyze the stochastic dynamics of both state transitions and duration times under the influence of multiple exogenous factors and random individual effect. The default setting flexibly models the transition probabilities using Dirichlet mixtures and the duration times using gamma mixtures. It also provides the flexibility of modeling the categorical sequences using Bayesian Markov mixed models alone, either ignoring the duration times altogether or dividing duration time into multiples of an additional category in the sequence by a user-specific unit. The package allows extensive inference of the state transition probabilities and the duration times as well as relevant plots and graphs. It also includes a synthetic data set to demonstrate the desired format of input data set and the utility of various functions. Methods for Bayesian Markov renewal mixed models are as described in: Abhra Sarkar et al., (2018) <doi:10.1080/01621459.2018.1423986> and Yutong Wu et al., (2022) <doi:10.1093/biostatistics/kxac050>.

Imports fields, logOfGamma, MCMCpack, multicool, pracma

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2024-04-22 05:42:36 UTC

Contents

`BMRMM` *Bayesian Markov Renewal Mixed Models (BMRMMs)*

Description

Provides inference results of both transition probabilities and duration times using BMRMMs.

Usage

```
BMRMM(
  data,
  num.cov,
  cov.labels = NULL,
  state.labels = NULL,
  random.effect = TRUE,
  fixed.effect = TRUE,
  trans.cov.index = 1:num.cov,
  duration.cov.index = 1:num.cov,
  duration.distr = NULL,
  duration.incl.prev.state = TRUE,
  simsize = 10000,
  burnin = simsize/2
)
```

Arguments

`data`	a data frame containing – individual ID, covariate values, previous state, current state, duration times (if applicable), in that order.
`num.cov`	total number of covariates provided in `data`.
`cov.labels`	a list of vectors giving names of the covariate levels. Default is a list of numerical vectors.
`state.labels`	a vector giving names of the states. Default is a numerical vector.
`random.effect`	`TRUE` if population-level effects are considered. Default is `TRUE`.
`fixed.effect`	`TRUE` if individual-level effects are considered. Default is `TRUE`.
`trans.cov.index`	a numeric vector indicating the indices of covariates that are used for transition probabilities. Default is all of the covariates.

`duration.cov.index`	a numeric vector indicating the indices of covariates that are used for duration times. Default is all of the covariates.
`duration.distr`	a list of arguments indicating the distribution of duration times. Default is `NULL`, which is ignoring duration times.
`duration.incl.prev.state`	`TRUE` if the previous state is included in the inference of duration times. Default is `TRUE`.
`simsize`	total number of MCMC iterations. Default is 10000.
`burnin`	number of burn-ins for the MCMC iterations. Default is `simsize`/2.

Details

Users have the option to ignore duration times or model duration times as a discrete or continuous variable via defining `duration.distr`.

`duration.distr` can be one of the following:

- `NULL`: duration times are ignored. This is the default setting.
- `list('mixgamma', shape, rate)`: duration times are modeled as a mixture gamma variable. `shape` and `rate` must be numeric vectors of the same length. The length indicates the number of mixture components.
- `list('mixDirichlet', unit)`: duration times are modeled as a new state with discretization `unit`. The duration state is then analyzed along with the original states. For example, if an duration time entry is 20 and `unit` is 5, then the model will add 4 consecutive new states. If an duration time entry is 23.33 and `unit` is 5, then the model will still add 4 consecutive new states as the blocks are calculated with the floor operation.

Value

An object of class `BMRMM` consisting of `results.trans` and `results.duration` if duration times are analyzed as a continuous variable.

The field `results.trans` is a data frame giving the inference results of transition probabilities.

`covs`	covariates levels for each row of the data.
`dpreds`	maximum level for each related covariate.
`MCMCparams`	MCMC parameters including simsize, burnin and thinning factor.
`tp.exgns.post.mean`	posterior mean of transition probabilities for different combinations of covariates.
`tp.exgns.post.std`	posterior standard deviation of transition probabilities for different combinations of covariates.
`tp.anmls.post.mean`	posterior mean of transition probabilities for different individuals.
`tp.anmls.post.std`	posterior standard deviation of transition probabilities for different individuals.
`tp.all.post.mean`	posterior mean of transition probabilities for different combinations of covariates AND different i
`tp.exgns.diffs.store`	difference in posterior mean of transition probabilities for every pair of covariate levels given level
`tp.exgns.all.itns`	population-level transition probabilities for every MCMC iteration.
`clusters`	number of clusters for each covariate for each MCMC iteration.
`cluster_labels`	the labels of the clusters for each covariate for each MCMC iteration.

`type`	a string identifier for results, which is "Transition Probabilities".
`cov.labels`	a list of string vectors giving labels of covariate levels.
`state.labels`	a list of strings giving labels of states.

The field `results.duration` is a data frame giving the inference results of duration times.

`covs`	covariates related to duration times.
`dpreds`	maximum level for each related covariate.
`MCMCparams`	MCMC parameters: simsize, burnin and thinning factor.
`duration.times`	duration times from the data set.
`comp.assignment`	mixture component assignment for each data point in the last MCMC iteration.
`duration.exgns.store`	posterior mean of mixture probabilities for different combinations of covariates of each MCMC ite
`marginal.prob`	estimated marginal mixture probabilities for each MCMC iteration.
`shape.samples`	estimated shape parameters for gamma mixtures for each MCMC iteration.
`rate.samples`	estimated rate parameters for gamma mixtures for each MCMC iteration.
`clusters`	number of clusters for each covariate for each MCMC iteration.
`cluster_labels`	the labels of the clusters for each covariate for each MCMC iteration.
`type`	a string identifier for results, which is "Duration Times".
`cov.labels`	a list of string vectors giving labels of covariate levels.

Author(s)

Yutong Wu, <`yutong.wu@utexas.edu`>

Examples

```
# In the examples, we use a shorted version of the foxp2 dataset, foxp2sm

# ignores duration times and only models transition probabilities using all three covariates
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 50)

# models duration times as a continuous variable with 3 gamma mixture components,
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 50,
                duration.distr = list('mixgamma', shape = rep(1,3), rate = rep(1,3)))

# models duration times as a discrete state with discretization 0.025 and
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 50,
                duration.distr = list('mixDirichlet', unit = 0.025))
```

`diag.BMRMM` *MCMC Diagnostic Plots for Transition Probabilities and Duration Times*

Description

Provides the traceplots and autocorrelation plots for (i) transition probabilities and (ii) mixture gamma shape and rate parameters.

Usage

```
diag.BMRMM(object, cov.combs = NULL, transitions = NULL, components = NULL)
```

Arguments

`object`	an object of class `BMRMM`
`cov.combs`	a list of covariate level combinations. Default is `NULL`, which is all possible combination of covariate levels.
`transitions`	a list of pairs denoting state transitions. Default is `NULL`, which is all possible state transitions.
`components`	a numeric vector denoting the mixture components of interest. Default is `NULL`, which is a list of all mixture components.

Value

None

Examples

```
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 80,
                 duration.distr = list('mixgamma',shape=rep(1,3),rate=rep(1,3)))
diag.BMRMM(results)
diag.BMRMM(results, cov.combs = list(c(1,1),c(1,2)),
           transitions = list(c(1,1)), components = c(3))
```

`foxp2` *Simulated FoxP2 Data Set.*

Description

A simulated data set of the original FoxP2 data set, which contains the sequences of syllables sung by male mice of different genotypes under various social contexts.

Usage

```
foxp2
```

Format

A data frame with 17391 rows and 6 variables:

Id Mouse Id

Genotype Genotype of the mouse, 1 = FoxP2 knocked out, 2 = wild type

Context Social context for the mouse, 1 = U (urine sample placed in the cage), 2 = L (living female mouse placed in the cage), 3 = A (an anesthetized female placed on the lid of the cage)

Prev_State The previous syllable, {1,2,3,4} = {d,m,s,u}

Cur_State The current syllable, {1,2,3,4} = {d,m,s,u}

Transformed_ISI Modified inter-syllable interval times, log(original ISI + 1)

References

Chabout, J., Sarkar, A., Patel, S. R., Radden, T., Dunson, D. B., Fisher, S. E., & Jarvis, E. D. (2016). A Foxp2 mutation implicated in human speech deficits alters sequencing of ultrasonic vocalizations in adult male mice. Frontiers in behavioral neuroscience, 10, 197.

Wu, Y., Jarvis E. D., & Sarkar, A. (2023). Bayesian semiparametric Markov renewal mixed models for vocalization syntax. Biostatistics, To appear.

`foxp2sm` *Shortened Simulated FoxP2 Data Set.*

Description

A shortened version of the `foxp2` data set for demonstrating R examples. See details of the `foxp2` data set by calling ?foxp2.

Usage

```
foxp2sm
```

Format

An object of class `data.frame` with 69 rows and 6 columns.

`hist.BMRMM` *Histogram of Duration Times*

Description

Plots the histogram of duration times in two ways as the users desire:

1. Histogram of all duration times superimposed the posterior mean mixture gamma distribution;
2. Histogram of a specified mixture component superimposed the gamma distribution with shape and rate parameters taken from the last MCMC iteration.

Usage

```
## S3 method for class 'BMRMM'
hist(
  x,
  comp = NULL,
  xlim = NULL,
  breaks = NULL,
  main = NULL,
  col = "gray",
  xlab = "Duration times",
  ylab = "Density",
  ...
)
```

Arguments

`x`	an object of class `BMRMM`.
`comp`	one of • `NULL`, which means the histogram for all duration times is plotted with the posterior mean mixture gamma distribution. Default option. • an integer specifying the mixture component for which the corresponding histogram is plotted with mixture gamma parameters taken from the last MCMC iteration.
`xlim`	a range of `x` values with sensible defaults. Default is `NULL`, which is to use `c(min(duration), max(duration))`.
`breaks`	an integer giving the number of cells for the histogram. Default is `NULL`, which is to use the Freedman-Diaconis rule, i.e., `(max(duration)-min(duration))*n^(1/3)/2/IQR(duration`
`main`	main title. Default is `NULL`, which is to use `"Histogram with Posterior Mean"` when `comp` is `NULL` and `"Component X"` if `comp` is specified.
`col`	color of the histogram bars. Default is `gray`.
`xlab`	x-axis label. Default is `"Duration times"`.
`ylab`	y-axis label. Default is `"Density"`.
`...`	further arguments for the hist function.

Value

An object of class `histogram`.

Examples

```
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 50,
                 duration.distr = list('mixgamma',shape=rep(1,3),rate=rep(1,3)))

# plot the histogram of all duration times superimposed with
# the posterior mixture gamma distribution
hist(results, xlim = c(0, 1), breaks = 50)

# plot the histogram for components 1 superimposed with
# the mixture gamma distribution of the last MCMC iteration
hist(results, components = 1)
```

`model.selection.scores` *Model Selection Scores for the Number of Components for Duration Times*

Description

Provides the LPML (Geisser and Eddy, 1979) and WAIC (Watanabe, 2010) scores of the Bayesian Markov renewal mixture models

Usage

```
model.selection.scores(object)
```

Arguments

`object`	An object of class BMRMM.

Details

The two scores can be used to compare different choices of isi_num_comp, i.e., the number of the mixture gamma components. Larger values of LPML and smaller values of WAIC indicate better model fits.

Value

a list consisting of LPML and WAIC scores for gamma mixture models.

References

Geisser, S. and Eddy, W. F. (1979). A predictive approach to model selection. Journal of the American Statistical Association, 74, 153–160.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. Journal of Machine Learning Research, 11, 3571–3594.

Examples

```
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 50,
                 duration.distr = list('mixgamma',shape=rep(1,3),rate=rep(1,3)))
model.selection.scores(results)
```

`plot.BMRMMsummary` *Plot Method for Visualizing BMRMM Summary*

Description

Visualization of a specified field of a `BMRMMsummary` object.

Usage

```
## S3 method for class 'BMRMMsummary'
plot(x, type, xlab = NULL, ylab = NULL, main = NULL, col = NULL, ...)
```

Arguments

`x`	an object of class `BMRMMsummary`.
`type`	a string indicating the plot(s) to draw. Must be named after a field of `object`.
`xlab`	x-axis label. Default is NULL.
`ylab`	y-axis label. Default is NULL.
`main`	main title. Default is NULL.
`col`	color of the plot. Default is NULL.
`...`	further arguments for the plot function.

Value

None

See Also

`summary.BMRMM()`

Examples

```
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 50,
                 cov.labels = list(c("F", "W"), c("U", "L", "A")),
                 duration.distr = list('mixgamma',shape=rep(1,3),rate=rep(1,3)))
fit.summary <- summary(results)
plot(fit.summary, 'trans.probs.mean')
plot(fit.summary, 'dur.mix.probs')
```

`summary.BMRMM` *Summary Method for Objects of Class* `BMRMM`

Description

Summarizing an object of class `BMRMM`, including results for transition probabilities and duration times, if applicable.

Usage

```
## S3 method for class 'BMRMM'
summary(object, delta = 0.02, digits = 2, ...)
```

Arguments

`object`	an object of class `BMRMM`.
`delta`	threshold for the null hypothesis for the local tests of transition probabilities (see Details). Default is 0.02.
`digits`	integer used for number formatting. Default is 2.
`...`	further arguments for the summary function.

Details

We give more explanation for the global tests and local tests results.

- Global tests (for both transition probabilities and duration times)

  Global tests are presented as a matrix, where the row denote the number of clusters and the column represents covariates. For each row `i` and column `j`, the matrix entry is the percentage of the number of the clusters within the stored MCMC samples for this covariate, i.e., an estimation for `Pr(# clusters for covariate j == i)`. We note that the probability `Pr(# clusters for covariate j > 1)` would be the probability for the null hypothesis that the covariate `j` is significant.
- Local tests (for transition probabilities only)

  Local tests focus on a particular covariate and compare the influence among its levels when the other covariates values are fixed.
  Given a pair of levels of covariate `j`, say `j_1` and `j_2`, and given the levels of other covariates, the null hypothesis is that the difference between `j_1` and `j_2` is not significant for transition

probabilities. It is calculated as the percentage of the samples with absolute difference less than `delta`.

The local tests provide two matrices of size `d0` x `d0` where `d0` is the number of states:

1. `mean.diff` – the mean of the absolute difference in each transition type between levels `j_1` and `j_2`;
2. `null.test` – the probability of the null hypothesis that `j_1` and `j_2` have the same significance for each transition type.

Value

An object of class `BMRMMsummary` with the following elements:

`trans.global`	global test results for transition probabilities (see Details).
`trans.probs.mean`	mean for the posterior transition probabilities.
`trans.probs.sd`	standard deviation for the posterior transition probabilities.
`trans.local.mean.diff`	the absolute difference in transition probabilities for a pair of covariate levels (see Details).
`trans.local.null.test`	probability for the null hypothesis that the difference between two covariate levels is not significa
`dur.global`	global test results for duration times (see Details).
`dur.mix.params`	mixture parameters taken from the last MCMC iteration if duration times follow a mixture gamm
`dur.mix.probs`	mixture probabilities for each covariate taken from the last MCMC iteration if duration times foll

See Also

plot.BMRMMsummary for plotting the summary results.

Examples

```
results <- BMRMM(foxp2sm, num.cov = 2, simsize = 50,
                 cov.labels = list(c(”F”, ”W”), c(”U”, ”L”, ”A”)),
                 duration.distr = list('mixgamma',shape=rep(1,3),rate=rep(1,3)))
sm <- summary(results)
sm
```

Index