

Package ‘BMRMM’

November 15, 2021

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

Version 0.0.1

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](https://doi.org/10.1080/01621459.2018.1423986)> and Yutong Wu et al., (2021) <[arXiv:2107.07648](https://arxiv.org/abs/2107.07648)>.

Imports fields, graphics, grDevices, gsubfn, logOfGamma, MCMCpack, mhsmm, multicool, pracma, stats, utils

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

NeedsCompilation no

Author Yutong Wu [aut, cre]

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

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2021-11-15 19:30:05 UTC

R topics documented:

BMRMM	2
-------	---

foxp2	5
foxp2_sm	5
get_duration_diagnostic_plots	6
get_estimated_post_mean_and_sd	6
get_global_test_results	8
get_heatmap_by_cov	9
get_heatmap_mix_param	10
get_histogram_by_component	10
get_histogram_with_post_mean	11
get_lpml_waic_scores	12
get_tp_diagnostic_plots	13
get_tp_local_test_results	14

Index	16
--------------	-----------

BMRMM	<i>Bayesian Markov Renewal Mixed Models (BMRMMs)</i>
-------	--

Description

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

Usage

```
BMRMM(
  data,
  num_cov,
  switch_off_random = FALSE,
  trans_cov_index = 1:num_cov,
  duration_type = "Continuous",
  duration_cov_index = 1:num_cov,
  duration_excl_prev_state = FALSE,
  duration_unit = NULL,
  duration_num_comp = 4,
  duration_init_shape = rep(1, duration_num_comp),
  duration_init_rate = rep(1, duration_num_comp),
  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'
switch_off_random	TRUE if only population-level effects are considered, default is FALSE

<code>trans_cov_index</code>	indices of covariates that are used for transition probabilities, default is all of the covariates
<code>duration_type</code>	one of 'None', 'Discrete', 'Continuous', default is 'Continuous'
<code>duration_cov_index</code>	indices of covariates that are used for duration times, default is all of the covariates plus the previous state
<code>duration_excl_prev_state</code>	TRUE if the previous state is excluded from duration times inference, default is FALSE
<code>duration_unit</code>	the discretization of duration times, only used when 'duration_type' is 'Discrete'
<code>duration_num_comp</code>	number of gamma mixture components for duration times, default is 4
<code>duration_init_shape</code>	initialization of mixture gamma shape parameters, default is a vector of 1 of size 'duration_num_comp'
<code>duration_init_rate</code>	initialization of mixture gamma rate parameters, default is a vector of 1 of size 'duration_num_comp'
<code>simsiz</code>	total number of MCMC iterations, default is 10000
<code>burnin</code>	number of burn-ins for the MCMC iterations, default is <code>simsiz/2</code>

Details

Returns `list(trans_results, duration_results)`

(1) Users have the option to ignore duration times or model duration times as a discrete or continuous variable via defining 'duration_type':

"None": ignores duration times

"Continuous": treat duration times as a continuous variable (DEFAULT)

"Discrete": treat duration times as a new state with discretization 'duration_unit'. If 'Discrete' is used, duration times becomes a new state type. 'duration_unit' must be specified if 'duration_type' is 'Discrete'. For example, if an duration time entry is 20 and 'duration_unit' is 5, then the model will add 4 consecutive new states. If an duration time entry is 23.33 and 'duration_unit' is 5, then the model will still add 4 consecutive new states as the blocks are calculated with the floor operation

(2) 'trans_results' contains the results of the transition probabilities:

- "Num_States": total number of states

- "Xexgns": covariates related to transition probabilities

- "dpreds": maximum level for each related covariate

- "MCMCparams": MCMC parameters: `simsiz`, `burnin` and thinning factor

- "TP_Exgns_Post_Mean": posterior mean of transition probabilities for different combinations of exogenous predictors

- "TP_Exgns_Post_Std": posterior standard deviation of transition probabilities for different combinations of exogenous predictors

- "TP_Anmls_Post_Mean": posterior mean of transition distribution components for different individuals
- "TP_All_Post_Mean": posterior mean of transition distribution components for different combinations of exogenous predictors AND different individuals
- "TP_Anmls_Post_Std": standard deviation of transition probabilities among different mice
- "TP_Exgns_Diffs_Store": difference in posterior mean of transition probabilities for every pair of covariate levels given levels of the other covariates
- "TP_Exgns_All_Itns": exogenous transition probabilities for every MCMC iteration
- "Clusters": number of clusters for each covariate for each MCMC iteration
- "Type": an identifier for results, which is "Transition Probabilities".

(3) 'duration_results' contains the results of the duration times ('duration_results' is NULL if 'duration_type' is 'None'):

```
"K" <- number of gamma mixture components
"Xexgns" <- covariates related to duration times
"dpreds" <- maximum level for each related covariate
"MCMCparams" <- MCMC parameters: simsize, burnin and thinning factor
"Duration_Times" <- input duration times
"Comp_Assign" <- mixture component assignment for each data point in the last MCMC iteration
"Duration_Exgns_Store" <- posterior mean of mixture probabilities for different combinations of exogenous predictors of each MCMC iteration
"Marginal_Prob" <- estimated marginal mixture probabilities for each iteration
"Shape_Samples" <- estimated shape parameters for gamma mixtures for each iteration
"Rate_Samples" <- estimated rate parameters for gamma mixtures for each iteration
"Clusters" <- number of clusters for each covariate for each MCMC iteration
"Type" <- an identifier for results, which is "Inter-Syllable Intervals".
```

Value

List of results for transition probabilities and durations times, `list(trans_results, duration_results)`. See details.

Author(s)

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

Examples

```
# In the examples, we use a shorted version of the foxp2 dataset, foxp2_sm

# ignores duration times and only models transition probabilities using all three covariates
results <- BMRMM(foxp2_sm,num_cov=2,duration_type='None',simsize=50)

# models duration times as a continuous variable with 5 gamma mixture components,
# using covariate 1 and the previous state
results <- BMRMM(foxp2_sm,num_cov=2,trans_cov_index=c(1),'duration_type'='Continuous',
                duration_cov_index=c(1),duration_num_comp=5,simsize=50)
```

```
# models duration times as a discrete state with discretization 0.25 and
# do not include the previous state as a covariate
results <- BMRMM(foxp2_sm,num_cov=2,duration_type='Discrete',duration_excl_prev_state=TRUE,
                duration_unit=0.025,simsize=50)
```

foxp2	<i>Simulated FoxP2 Data Set.</i>
-------	----------------------------------

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 70818 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}

ISI Modified inter-syllable interval times, $\log(\text{original ISI} + 1)$

foxp2_sm	<i>Shortened Simulated FoxP2 Data Set.</i>
----------	--

Description

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

Usage

```
foxp2_sm
```

Format

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

`get_duration_diagnostic_plots`*Duration Times Plots: MCMC Diagnostic Plots*

Description

Provides the traceplots and autocorrelation plots for each gamma shape and rate parameter

Usage

```
get_duration_diagnostic_plots(results, comps = 1:results$K)
```

Arguments

<code>results</code>	results of duration times, i.e., <code>results\$results_duration</code>
<code>comps</code>	a vector of the components to plot; default is <code>1:duration_num_comp</code>

Value

No return value, called for plotting MCMC diagnostic plots for duration times.

Examples

```
results <- BMRMM(foxp2_sm,num_cov=2,simsize=80)

# get diagnostic plots for all components
get_duration_diagnostic_plots(results$results_duration)

# get diagnostic plots for components 1,2
get_duration_diagnostic_plots(results$results_duration,comps=c(1,2))
```

`get_estimated_post_mean_and_sd`*Transition Probabilities: Posterior Mean and Standard Deviation*

Description

Print and plot the posterior mean and standard deviation for transition probabilities from MCMC samples under given different combinations of covariate levels.

Usage

```
get_estimated_post_mean_and_sd(
  results,
  cov_labels = NULL,
  state_labels = 1:results$Num_States,
  cov_levels = NULL,
  decimal_pts = 2,
  include_plot = TRUE
)
```

Arguments

results	results of transition probabilities, i.e., results\$results_trans
cov_labels	a matrix such that row i represents the labels for covariate i; default labels for covariate i is 1:i
state_labels	a vector of strings that represent the state labels; default is 1:Num_States
cov_levels	a matrix such that each row is a combination of covariate levels; default is all possible combinations of covariates
decimal_pts	specify the number of decimal points of the results; default is 2
include_plot	display plot if TRUE; default is TRUE

Details

For each row of 'cov_levels', the function returns two matrices of size $d_0 \times d_0$ where d_0 is the number of states: (1) the posterior mean and (2) the posterior standard deviation of transition probabilities, computed from MCMC samples after burn-ins and thinning. The default for 'cov_levels' is all possible combinations of covariate levels.

Value

No return value, called for printing and plotting posterior distribution of transition probabilities.

Examples

```
# Examples using the shortened version of the simulated Foxp2 data set, foxp2_sm

# get results for all combinations of covariate levels
results <- BMRMM(foxp2_sm,num_cov=2,duration_type='None',simsize=50)
get_estimated_post_mean_and_sd(results$results_trans)

# get results for covariate levels ("HET","U") and ("WT","U")
cov_labels <- matrix(c("HET","WT","","U","L","A"),nrow=2,byrow=TRUE)
cov_levels <- matrix(c(1,1,2,1),nrow=2,byrow=TRUE)
get_estimated_post_mean_and_sd(results$results_trans,cov_labels,cov_levels=cov_levels)
```

`get_global_test_results`*Global Test Results for Transition Probabilities and Duration Times*

Description

For each covariate, print (and plot) the percentage of the number of clusters for each covariate in stored MCMC iterations.

Usage

```
get_global_test_results(results, decimal_pts = 2, include_plot = TRUE)
```

Arguments

<code>results</code>	either <code>results\$results_trans</code> or <code>results\$results_duration</code>
<code>decimal_pts</code>	specify the number of decimal points of the results; default is 2
<code>include_plot</code>	display plot if TRUE; default is TRUE

Details

The height of the bar represents the percentage of the number of the clusters within the stored MCMC samples.

$\Pr(\# \text{ clusters for covariate } j == 1)$ is the probability for the null hypothesis that the covariate j is not significant for transition probabilities or duration times, depending on the result type.

Value

No return value, called for printing and plotting global test results.

Examples

```
results <- BMRMM(foxp2_sm, num_cov=2, simsize=50)
get_global_test_results(results$results_trans)
get_global_test_results(results$results_duration)
```

get_heatmap_by_cov	<i>Duration Times Plot: Heat-map of Mixture Probabilities for Different Covariates</i>
--------------------	--

Description

Print and plot the heat-map of mixture probabilities for covariate levels

Usage

```
get_heatmap_by_cov(
  results,
  cov_labels = NULL,
  cov_index = 1:ncol(results$Xexgns),
  decimal_pts = 2,
  include_plot = TRUE
)
```

Arguments

results	results of duration times, i.e., results\$results_duration
cov_labels	a matrix such that row i represents the labels for covariate i; default labels for covariate i is 1:i
cov_index	a vector of indices of covariate to plot; default is 1:Num_Covariates
decimal_pts	specify the number of decimal points of the results; default is 2
include_plot	display plot if TRUE; default is TRUE

Details

Print a matrix for each covariate. Each matrix has K rows, where K is the number of components. The number of columns of a matrix equals to the total number of levels of the covariate.

Value

No return value, called for printing and plotting heatmaps for mixture probabilities for each covariate.

Examples

```
results <- BMRMM(foxp2_sm, num_cov=2, simsize=50)
get_heatmap_by_cov(results$results_duration)
get_heatmap_by_cov(results$results_duration, cov_index=c(1,2))
```

get_heatmap_mix_param *Duration Times Plot: Heat-map of Mixture Gamma Shape & Rate Parameters*

Description

Print and plot the heat-map for the gamma mixture parameters for each component

Usage

```
get_heatmap_mix_param(results, decimal_pts = 2, include_plot = TRUE)
```

Arguments

results	results of duration times, i.e., results\$results_duration
decimal_pts	specify the number of decimal points of the results; default is 2
include_plot	display plot if TRUE; default is TRUE

Details

Print a matrix of size $K \times 2$, where K is the number of components. Each row i represents the shape and rate parameter of component i .

Value

No return value, called for printing and plotting heatmaps for mixture parameters.

Examples

```
results <- BMRMM(foxp2_sm, num_cov=2, simsize=50)
get_heatmap_mix_param(results$results_duration)
```

get_histogram_by_component

Duration Times Plots: Histogram of duration times by Component

Description

Plots the histogram of each mixture component superimposed the gamma distribution with shape and rate parameters taken from the last MCMC iteration

Usage

```
get_histogram_by_component(results, comp = 1:results$K)
```

Arguments

results results of duration times, i.e., results\$results_duration
 comp a vector of the components to plot; default is 1:isi_num_comp

Value

No return value, called for plotting histogram for each mixture component.

Examples

```
results <- BMRMM(foxp2_sm,num_cov=2,simsize=50)
get_histogram_by_component(results$results_duration)
```

```
get_histogram_with_post_mean
```

*Duration Times Plots: Histogram of duration times with Estimated
 Posterior Mean*

Description

Plots the histogram of Duration Times superimposed the posterior mean mixture gamma distribution

Usage

```
get_histogram_with_post_mean(
  results,
  x_range = c(min(results$Duration_Times), max(results$Duration_Times)),
  breaks = (max(results$Duration_Times) - min(results$Duration_Times)) *
    length(results$Duration_Times)^(1/3)/2/IQR(results$Duration_Times)
)
```

Arguments

results results of duration times, i.e., results_duration
 x_range custom range for x-axis; default is (min(durations),max(durations))
 breaks custom breaks for the histogram; default breaks follows the Freedman-Diaconis
 rule, which would be (max(durations)-min(durations))*n^(1/3)/2/IQR(durations)

Value

No return value, called for plotting histogram superimposed with the posterior distribution of mixture gamma.

Examples

```
results <- BMRMM(foxp2_sm,num_cov=2,simsize=50)
get_histogram_with_post_mean(results$results_duration,x_range=c(0,2),breaks=50)
```

get_lpml_waic_scores *Duration Times Result: Model Selection Scores for the Number of Components*

Description

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

Usage

```
get_lpml_waic_scores(results)
```

Arguments

results results of duration times, i.e., results\$results_duration

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

No return value, called for printing 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(foxp2_sm,num_cov=2,simsize=50)
get_lpml_waic_scores(results$results_duration)
```

 get_tp_diagnostic_plots

Transition Probabilities: MCMC Diagnostic Plots

Description

Provides the traceplots and autocorrelation plots for each transition type under combinations of covariate levels

Usage

```
get_tp_diagnostic_plots(
  results,
  from = 1:results$Num_States,
  to = 1:results$Num_States,
  cov_levels = NULL,
  cov_labels = NULL,
  state_labels = 1:results$Num_States
)
```

Arguments

results	results of transition probabilities, i.e., results\$results_trans
from	list of "from" states in target state transitions; default is all states
to	list of "to" states in target state transitions; default is all states
cov_levels	a matrix such that each row is a combination of covariate levels; default is all possible combinations of covariates
cov_labels	a matrix such that row i represents the labels for covariate i; default labels for covariate i is 1:i
state_labels	a vector of strings that represent the state labels; default is 1:Num_States

Value

No return value, called for plotting MCMC diagnostic plots for transition probabilities.

Examples

```
# Examples using the shortened built-in simulated Foxp2 data set, foxp2_sm
# recall the covariate labels are: ("HET", "WT") and ("U", "L", "A")
# recall the state labels are 'd', 'm', 's', 'u'

results <- BMRMM(foxp2_sm, num_cov=2, duration_type='None', simsize=50)

# results for all transition types for every combinations of covariates
get_tp_diagnostic_plots(results$results_trans)
```

```
# results for all transition types for covariate levels (HET,U) and (WT,A)
get_tp_diagnostic_plots(results$results_trans,cov_levels=matrix(c(1,1,2,3),nrow=2,byrow=TRUE))

# results for all transition from (d,m) to (d,s) for covariate levels (HET,U) and (WT,A)
get_tp_diagnostic_plots(results$results_trans,from=c(1),to=c(2,3),
                        cov_levels=matrix(c(1,1,2,3),nrow=2,byrow=TRUE))
```

```
get_tp_local_test_results
```

Transition Probabilities: Local Tests for a Specific Covariate

Description

Given some covariate 'cov' and the levels of other covariates, prints and plots the local test results for pairs of levels of 'cov', including the absolute differences in transition probabilities, and the probabilities for the null hypothesis.

Usage

```
get_tp_local_test_results(
  results,
  cov,
  delta,
  all_cov_labels = NULL,
  state_labels = 1:results$Num_States,
  other_cov_levels = NULL,
  comp_pairs = NULL,
  decimal_pts = 2,
  include_plot = TRUE
)
```

Arguments

results	results of transition probabilities, i.e., results\$results_trans
cov	the index of the covariate to perform local tests on
delta	threshold for the null hypothesis H_0 ; see Details
all_cov_labels	a matrix such that row i represents the labels for covariate i ; default labels for covariate i is $1:i$
state_labels	a vector of strings that represent the state labels; default is $1:\text{Num_States}$
other_cov_levels	a matrix such that each row is a combination of covariate levels excluding 'cov'; default is all possible combinations of covariates excluding 'cov'
comp_pairs	a matrix such that each row is a pair of levels of 'cov' to compare
decimal_pts	specify the number of decimal points of the results; default is 2
include_plot	display plot if TRUE; default is TRUE

Details

Fix a covariate i as the target of the local test.

The function provides two matrices of size $d_0 \times d_0$ where d_0 is the number of states:

- (1) the mean of the absolute difference in all transition types for each pair of levels of covariate i ;
- (2) the probability of the null hypothesis of the local test for all transition types.

Given a pair of levels of covariate i , say i_1 and i_2 , and given the levels of other covariates, the null hypothesis is that the difference between i_1 and i_2 is not significant for transition probabilities. It is calculated as the percentage of the samples with absolute difference less than 'delta'.

'comp_pairs' are user-specified pairs of 'cov' for the local test. Each row of 'comp_pairs' is a pair of indices of 'cov'.

Value

No return value, called for printing and plotting local test results for transition probabilities.

Examples

```
# Examples using the shortened built-in simulated Foxp2 data set, foxp2_sm
# recall the covariate labels are: ("HET", "WT") and ("U", "L", "A")

results <- BMRMM(foxp2_sm, num_cov=2, duration_type='None', simsize=50)

# results for genotypes (HET, WT) under all three social contexts
get_tp_local_test_results(results$results_trans, cov=1, delta=0.02)

# results for genotypes (HET, WT) under social contexts U and A
get_tp_local_test_results(results$results_trans, cov=1, delta=0.02,
  other_cov_levels=matrix(c(1,3), nrow=2))

# results for contexts (U,L), (U,A) and (L,A) under two genotypes
get_tp_local_test_results(results$results_trans, cov=2, delta=0.02)

# results for contexts (U,L), (U,A) under genotype HET (specify pairs)
get_tp_local_test_results(results$results_trans, cov=2, delta=0.02,
  other_cov_levels=matrix(1, nrow=1),
  comp_pairs=matrix(c(1,2,1,3), nrow=2, byrow=2))
```

Index

* datasets

foxp2, [5](#)

foxp2_sm, [5](#)

BMRMM, [2](#)

foxp2, [5](#)

foxp2_sm, [5](#)

get_duration_diagnostic_plots, [6](#)

get_estimated_post_mean_and_sd, [6](#)

get_global_test_results, [8](#)

get_heatmap_by_cov, [9](#)

get_heatmap_mix_param, [10](#)

get_histogram_by_component, [10](#)

get_histogram_with_post_mean, [11](#)

get_lpml_waic_scores, [12](#)

get_tp_diagnostic_plots, [13](#)

get_tp_local_test_results, [14](#)