Package 'ggmHMM'

May 19, 2025

Type Package

Title Plotting Bayesian Multilevel Hidden Markov Models from the mHMMbayes Package Using ggplot2

Version 0.1.0

Maintainer Pepijn Vink <p.a.vink@uu.nl>

Description Visualize objects created using the mHMMbayes package wih ggplot2. The plotting functions output 'ggplot' objects that can be edited using the 'ggplot2' package. ggmHMM includes functions to plot obtained emission and transition probability distributions for both the group and the subject level. The package can also be used to obtain trace plots to evaluate convergence, plot infered states obtained using the viterbi algorithm, and plot the posterior distributions of estimates.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports cli,
dplyr,
ggplot2,
magrittr,
mHMMbayes,
rlang,
stats,
tibble,
tidyr,
tidyselect

Suggests knitr,
rmarkdown,
testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Contents

`plot_emiss` *Plot emission distributions of a Bayesian Multilevel Hidden Markov Model*

Description

Plot emission distributions of a Bayesian Multilevel Hidden Markov Model

Usage

```
plot_emiss(
  model,
  type = "bar",
  subject_effects = TRUE,
  cat_labels = NULL,
  position = ggplot2::position_jitter(width = 0.2, height = 0),
  alpha = 0.75,
  line = FALSE,
  subject = NULL
)
```

Arguments

`model`	Object of type ‘mHMMbayes::mHMM‘, created using [mHMMbayes::mHMM()].
`type`	String specifying the type of plot to return. Currently takes "bar" and "boxplot".
`subject_effects`	Logical specifying whether a layer of individual estimates should be plotted.
`cat_labels`	Character vector of labels for the categorical variables.
`position`	Object created with ggplot2::position_jitter indicating the amount of jitter.
`alpha`	Numeric value indicating transparency of subject-specific posterior densities.
`line`	Logical indicating whether to plot lines when plotton individual-level distributions.
`subject`	Vector indicating the subjects to plot when ‘subject_effects = TRUE‘. Default is ‘NULL‘, which means all subjects are plotted.

Value

Object of type ‘ggplot2::gg‘ plotting emission distributions.

Examples

```
## Not run:
library(mHMMbayes)
# simulating multivariate continuous data
n_t     <- 100
n       <- 10
m       <- 3
n_dep   <- 2

gamma   <- matrix(c(0.8, 0.1, 0.1,
                    0.2, 0.7, 0.1,
                    0.2, 0.2, 0.6), ncol = m, byrow = TRUE)

emiss_distr <- list(matrix(c( 50, 10,
                              100, 10,
                              150, 10), nrow = m, byrow = TRUE),
                    matrix(c(5, 2,
                             10, 5,
                             20, 3), nrow = m, byrow = TRUE))

data_cont <- sim_mHMM(n_t = n_t, n = n, data_distr = 'continuous',
                      gen = list(m = m, n_dep = n_dep),
                      gamma = gamma, emiss_distr = emiss_distr,
                      var_gamma = .1, var_emiss = c(5^2, 0.2^2))

# Specify hyper-prior for the continuous emission distribution
manual_prior_emiss <- prior_emiss_cont(
                        gen = list(m = m, n_dep = n_dep),
                        emiss_mu0 = list(matrix(c(30, 70, 170), nrow = 1),
                                         matrix(c(7, 8, 18), nrow = 1)),
                        emiss_K0 = list(1, 1),
                        emiss_V =  list(rep(5^2, m), rep(0.5^2, m)),
                        emiss_nu = list(1, 1),
                        emiss_a0 = list(rep(1.5, m), rep(1, m)),
                        emiss_b0 = list(rep(20, m), rep(4, m)))

# Run the model on the simulated data:
# Note that for reasons of running time, J is set at a ridiculous low value.
# One would typically use a number of iterations J of at least 1000,
# and a burn_in of 200.
out_3st_cont_sim <- mHMM(s_data = data_cont$obs,
                         data_distr = 'continuous',
                         gen = list(m = m, n_dep = n_dep),
                         start_val = c(list(gamma), emiss_distr),
                         emiss_hyp_prior = manual_prior_emiss,
                         mcmc = list(J = 11, burn_in = 5))

plot_emiss(out_3st_cont_sim)

## End(Not run)
```

`plot_gamma` *Heat Plot for Transition Probability Matrix of a Bayesian Multilevel Hidden Markov Model*

Description

Heat Plot for Transition Probability Matrix of a Bayesian Multilevel Hidden Markov Model

Usage

```
plot_gamma(
  model = NULL,
  level = "group",
  subject = NULL,
  digits = 2,
  facet = TRUE,
  ncol_facet = 2
)
```

Arguments

model	Object of type 'mHMMbayes::mHMM' or 'mHMMbayes::mHMM_gamma', created using [mHMMbayes::mHMM()] or [mHMMbayes::obtain_gamma()].
level	String specifying the level of transition distributions to plot. Options are "group" and "subject".
subject	Optional integer or integer vector specifying the subject(s) to plot.
digits	Integer specifying the number of digits to round to.
facet	Logical specifying whether subjects should be faceted when plotting subject-specific transition probability matrices.
ncol_facet	Integer specifying the number of columns in the facet grid when plotting subject-specific transition probability matrices.

Value

Object of type 'ggplot2::gg' with the visualized transition distributions.

Examples

```
## Not run:
library(mHMMbayes)
# simulating multivariate continuous data
n_t     <- 100
n       <- 10
m       <- 3
n_dep   <- 2

gamma   <- matrix(c(0.8, 0.1, 0.1,
                    0.2, 0.7, 0.1,
                    0.2, 0.2, 0.6), ncol = m, byrow = TRUE)

emiss_distr <- list(matrix(c( 50, 10,
                              100, 10,
                              150, 10), nrow = m, byrow = TRUE),
                    matrix(c(5, 2,
                             10, 5,
                             20, 3), nrow = m, byrow = TRUE))

data_cont <- sim_mHMM(n_t = n_t, n = n, data_distr = 'continuous',
```

```
                      gen = list(m = m, n_dep = n_dep),
                      gamma = gamma, emiss_distr = emiss_distr,
                      var_gamma = .1, var_emiss = c(5^2, 0.2^2))

# Specify hyper-prior for the continuous emission distribution
manual_prior_emiss <- prior_emiss_cont(
                        gen = list(m = m, n_dep = n_dep),
                        emiss_mu0 = list(matrix(c(30, 70, 170), nrow = 1),
                                         matrix(c(7, 8, 18), nrow = 1)),
                        emiss_K0 = list(1, 1),
                        emiss_V =  list(rep(5^2, m), rep(0.5^2, m)),
                        emiss_nu = list(1, 1),
                        emiss_a0 = list(rep(1.5, m), rep(1, m)),
                        emiss_b0 = list(rep(20, m), rep(4, m)))

# Run the model on the simulated data:
# Note that for reasons of running time, J is set at a ridiculous low value.
# One would typically use a number of iterations J of at least 1000,
# and a burn_in of 200.
out_3st_cont_sim <- mHMM(s_data = data_cont$obs,
                         data_distr = 'continuous',
                         gen = list(m = m, n_dep = n_dep),
                         start_val = c(list(gamma), emiss_distr),
                         emiss_hyp_prior = manual_prior_emiss,
                         mcmc = list(J = 11, burn_in = 5))

plot_gamma(out_3st_cont_sim)

## End(Not run)
```

`plot_posterior` *Plot posterior distributions of a Bayesian Multilevel Hidden Markov Model*

Description

Plot posterior distributions of a Bayesian Multilevel Hidden Markov Model

Usage

```
plot_posterior(
  model,
  component = "gamma",
  vrb = NULL,
  state_labels = NULL,
  cat_labels = NULL,
  burnin = NULL,
  alpha = 0.1
)
```

Arguments

`model`	Object of type ‘mHMMbayes::mHMM‘ or ‘mHMMbayes::mHMM_vary‘ created using [mHMMbayes::mHMM()] or [mHMMbayes::mHMM_vary()].

component	Character string specifying the component to plot. Takes "gamma" or "emiss".
vrb	Character string specifying the dependent variable to plot when plotting emission distributions.
state_labels	Optional character string specifying labels to use for the inferred states.
cat_labels	Optional character string used to specify labels for categories when plotting emission distributions of categorical variables.
burnin	Optional integer specifying number of burnin iterations. If unspecified, the number of burnin iterations specified when fitting the model is used.
alpha	Transparency of densities representing subject-specific posterior densities.

Value

Object of type ‘ggplot2::gg‘ plotting posterior distributions.

Examples

```
## Not run:
library(mHMMbayes)
# simulating multivariate continuous data
n_t     <- 100
n       <- 10
m       <- 3
n_dep   <- 2

gamma   <- matrix(c(0.8, 0.1, 0.1,
                    0.2, 0.7, 0.1,
                    0.2, 0.2, 0.6), ncol = m, byrow = TRUE)

emiss_distr <- list(matrix(c( 50, 10,
                              100, 10,
                              150, 10), nrow = m, byrow = TRUE),
                    matrix(c(5, 2,
                             10, 5,
                             20, 3), nrow = m, byrow = TRUE))

data_cont <- sim_mHMM(n_t = n_t, n = n, data_distr = 'continuous',
                      gen = list(m = m, n_dep = n_dep),
                      gamma = gamma, emiss_distr = emiss_distr,
                      var_gamma = .1, var_emiss = c(5^2, 0.2^2))

# Specify hyper-prior for the continuous emission distribution
manual_prior_emiss <- prior_emiss_cont(
                        gen = list(m = m, n_dep = n_dep),
                        emiss_mu0 = list(matrix(c(30, 70, 170), nrow = 1),
                                         matrix(c(7, 8, 18), nrow = 1)),
                        emiss_K0 = list(1, 1),
                        emiss_V =  list(rep(5^2, m), rep(0.5^2, m)),
                        emiss_nu = list(1, 1),
                        emiss_a0 = list(rep(1.5, m), rep(1, m)),
                        emiss_b0 = list(rep(20, m), rep(4, m)))

# Run the model on the simulated data:
# Note that for reasons of running time, J is set at a ridiculous low value.
# One would typically use a number of iterations J of at least 1000,
# and a burn_in of 200.
```

```
out_3st_cont_sim <- mHMM(s_data = data_cont$obs,
                         data_distr = 'continuous',
                         gen = list(m = m, n_dep = n_dep),
                         start_val = c(list(gamma), emiss_distr),
                         emiss_hyp_prior = manual_prior_emiss,
                         mcmc = list(J = 11, burn_in = 5))

plot_posterior(model = out_3st_cont_sim)

## End(Not run)
```

`plot_trace` *Plot trace plots to assess convergence of a Bayesian Multilevel Hidden Markov Model*

Description

Plot trace plots to assess convergence of a Bayesian Multilevel Hidden Markov Model

Usage

```
plot_trace(
  model,
  component = "gamma",
  param = NULL,
  level = "group",
  vrb = NULL,
  prob = TRUE,
  subject = NULL,
  state_labels = NULL,
  cat_labels = NULL,
  alpha = 1
)
```

Arguments

`model`	Object or a list of objects of type ‘mHMMbayes::mHMM‘ created using [mHMMbayes::mHMM()].
`component`	Character string specifying the component to plot. Takes "gamma" or "emiss".
`param`	Optional character string specifying the parameter to plot for the plotted component. If ‘NULL‘ (default), plots the means (or probabilities). Takes "var" for between-person variances, "sd" for standard deviations of normal emission distributions, and "beta" for regression coefficients.
`level`	Character string specifying the level of parameter to plot. Takes "group" or "subject".
`vrb`	Optional character string specifying the variable to plot when plotting categorical emission distributions.
`prob`	Logical specifying whether converence of transitions or categorical emissions should be plotted on the probability scale, rather than the log scale.
`subject`	Integer specifying the subject to plot subject specific parameters for.

state_labels	Optional character string specifying labels to use for the states.
cat_labels	Optional character string used to specify labels for categories when plotting emission distributions of categorical variables.
alpha	Numeric value specifying the transparency of the lines in the plot. Default is 1.

Value

Object of type 'ggplot2::gg', plotting parameter distributions.

Examples

```
## Not run:
library(mHMMbayes)
# simulating multivariate continuous data
n_t     <- 100
n       <- 10
m       <- 3
n_dep   <- 2

gamma   <- matrix(c(0.8, 0.1, 0.1,
                    0.2, 0.7, 0.1,
                    0.2, 0.2, 0.6), ncol = m, byrow = TRUE)

emiss_distr <- list(matrix(c( 50, 10,
                              100, 10,
                              150, 10), nrow = m, byrow = TRUE),
                    matrix(c(5, 2,
                             10, 5,
                             20, 3), nrow = m, byrow = TRUE))

data_cont <- sim_mHMM(n_t = n_t, n = n, data_distr = 'continuous',
                      gen = list(m = m, n_dep = n_dep),
                      gamma = gamma, emiss_distr = emiss_distr,
                      var_gamma = .1, var_emiss = c(5^2, 0.2^2))

# Specify hyper-prior for the continuous emission distribution
manual_prior_emiss <- prior_emiss_cont(
                        gen = list(m = m, n_dep = n_dep),
                        emiss_mu0 = list(matrix(c(30, 70, 170), nrow = 1),
                                         matrix(c(7, 8, 18), nrow = 1)),
                        emiss_K0 = list(1, 1),
                        emiss_V =  list(rep(5^2, m), rep(0.5^2, m)),
                        emiss_nu = list(1, 1),
                        emiss_a0 = list(rep(1.5, m), rep(1, m)),
                        emiss_b0 = list(rep(20, m), rep(4, m)))

# Run the model on the simulated data:
# Note that for reasons of running time, J is set at a ridiculous low value.
# One would typically use a number of iterations J of at least 1000,
# and a burn_in of 200.
out_3st_cont_sim <- mHMM(s_data = data_cont$obs,
                         data_distr = 'continuous',
                         gen = list(m = m, n_dep = n_dep),
                         start_val = c(list(gamma), emiss_distr),
                         emiss_hyp_prior = manual_prior_emiss,
                         mcmc = list(J = 11, burn_in = 5))
```

```
plot_trace(model = out_3st_cont_sim,
              param = "gamma",
              level = "group",
              prob = TRUE)

## End(Not run)
```

---

plot_viterbi *Plot inferred states of a Bayesian Multilevel Hidden Markov Model*

Description

Plot inferred states of a Bayesian Multilevel Hidden Markov Model

Usage

```
plot_viterbi(states, s_data, subject = NULL)
```

Arguments

states	Data Frame with inferred states obtained using [mHMMbayes::vit_HMM()] or object of class 'mHMMbayes:mHMM'.
s_data	Data Frame with data used to infer states using the viterbi algorithm. Only required when the object given to 'states' is of class 'mHMMbayes::mHMM'.
subject	Optional numeric vector with indices of subjects to plot.

Value

Object of type 'ggplot2::gg' with the plotted inferred states over time.

Examples

```
## Not run:
library(mHMMbayes)
# simulating multivariate continuous data
n_t     <- 100
n       <- 10
m       <- 3
n_dep   <- 2

gamma   <- matrix(c(0.8, 0.1, 0.1,
                    0.2, 0.7, 0.1,
                    0.2, 0.2, 0.6), ncol = m, byrow = TRUE)

emiss_distr <- list(matrix(c( 50, 10,
                              100, 10,
                              150, 10), nrow = m, byrow = TRUE),
                    matrix(c(5, 2,
                             10, 5,
                             20, 3), nrow = m, byrow = TRUE))

data_cont <- sim_mHMM(n_t = n_t, n = n, data_distr = 'continuous',
```

```
                      gen = list(m = m, n_dep = n_dep),
                      gamma = gamma, emiss_distr = emiss_distr,
                      var_gamma = .1, var_emiss = c(5^2, 0.2^2))

# Specify hyper-prior for the continuous emission distribution
manual_prior_emiss <- prior_emiss_cont(
                        gen = list(m = m, n_dep = n_dep),
                        emiss_mu0 = list(matrix(c(30, 70, 170), nrow = 1),
                                         matrix(c(7, 8, 18), nrow = 1)),
                        emiss_K0 = list(1, 1),
                        emiss_V =  list(rep(5^2, m), rep(0.5^2, m)),
                        emiss_nu = list(1, 1),
                        emiss_a0 = list(rep(1.5, m), rep(1, m)),
                        emiss_b0 = list(rep(20, m), rep(4, m)))

# Run the model on the simulated data:
# Note that for reasons of running time, J is set at a ridiculous low value.
# One would typically use a number of iterations J of at least 1000,
# and a burn_in of 200.
out_3st_cont_sim <- mHMM(s_data = data_cont$obs,
                         data_distr = 'continuous',
                         gen = list(m = m, n_dep = n_dep),
                         start_val = c(list(gamma), emiss_distr),
                         emiss_hyp_prior = manual_prior_emiss,
                         mcmc = list(J = 11, burn_in = 5))

states <- vit_HMM(s_data = data_cont$obs,
                  object = out_3st_cont_sim)
plot_viterbi(states)

## End(Not run)
```

Index