

Package ‘BEND’

March 14, 2024

Title Bayesian Estimation of Nonlinear Data (BEND)

Version 0.0.0.9000

Description Provides a set of models to estimate nonlinear longitudinal data using Bayesian estimation methods. These models include the: 1) Bayesian Piecewise Random Effects Model (`Bayes_PREM()`) which estimates a piecewise random effects (mixture) model for a given number of latent classes and a latent number of possible change-points in each class, and can incorporate class and outcome predictive covariates, 2) Bayesian Crossed Random Effects Model (`Bayes_CREM()`) which estimates a linear, quadratic, exponential, or piecewise crossed random effects models where individuals are changing groups over time (e.g., students and schools), and 3) Bayesian Bivariate Piecewise Random Effects Model (`Bayes_BPREM()`) which estimates a bivariate piecewise random effects model to jointly model two related outcomes (e.g., reading and math achievement).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

URL <https://github.com/crohlo/BEND>

BugReports <https://github.com/crohlo/BEND/issues>

Depends R (>= 3.6.3)

Imports coda (>= 0.19.4), label.switching (>= 1.8), rjags (>= 4.14)

LazyData true

NeedsCompilation no

Author Corissa T. Rohloff [aut, cre] (<<https://orcid.org/0000-0003-3228-4653>>),
Rik Lamm [aut] (<<https://orcid.org/0000-0002-3317-6243>>),
Yadira Peralta [aut] (<<https://orcid.org/0000-0003-4823-6939>>),
Nidhi Kohli [aut] (<<https://orcid.org/0000-0003-4690-2854>>),
Eric F. Lock [aut] (<<https://orcid.org/0000-0003-4663-2356>>)

Maintainer Corissa T. Rohloff <corissa.wurth@gmail.com>

R topics documented:

Bayes_BPREM	2
Bayes_CREM	4
Bayes_PREM	7

plot_BEND	11
SimData_BPREM	13
SimData_PCREM	13
SimData_PREM	14
summary.BPREM	14
summary.CREM	15
summary.PREM	16

Index	18
--------------	-----------

Bayes_BPREM	<i>Bayesian Bivariate Piecewise Random Effects Model (BPREM)</i>
-------------	--

Description

Estimates a Bayesian bivariate piecewise random effects models (BPREM) for longitudinal data with two interrelated outcomes. See Peralta et al. (2022) for more details.

Usage

```
Bayes_BPREM(
  data,
  id_var,
  time_var,
  y1_var,
  y2_var,
  iters_adapt = 5000,
  iters_burn_in = 1e+05,
  iters_sampling = 50000,
  thin = 15,
  save_full_chains = FALSE,
  save_conv_chains = FALSE
)
```

Arguments

data	Data frame in long format, where each row describes a measurement occasion for a given individual. It is assumed that each individual has the same number of assigned timepoints (a.k.a., rows). There can be missingness in the outcomes (y1_var and y2_var), but there cannot be missingness in time (time_var).
id_var	Name of column that contains ids for individuals with repeated measures in a longitudinal dataset.
time_var	Name of column that contains the time variable. This column cannot contain any missing values.
y1_var	Name of column that contains the first outcome variable. Missing values should be denoted by NA.
y2_var	Name of column that contains the second outcome variable. Missing values should be denoted by NA.
iters_adapt	(optional) Number of iterations for adaptation of jags model (default = 5000).
iters_burn_in	(optional) Number of iterations for burn-in (default = 100000).

`iters_sampling` (optional) Number of iterations for posterior sampling (default = 50000).

`thin` (optional) Thinning interval for posterior sampling (default = 15).

`save_full_chains`
Logical indicating whether the MCMC chains from `rjags` should be saved (default = FALSE). Note, this should not be used regularly as it will result in an object with a large file size.

`save_conv_chains`
Logical indicating whether the MCMC chains from `rjags` should be saved but only for the parameters monitored for convergence (default = FALSE). This would be useful for plotting traceplots for relevant model parameters to evaluate convergence behavior. Note, this should not be used regularly as it will result in an object with a large file size.

Details

For more information on the model equation and priors implemented in this function, see Peralta et al. (2022).

Value

A list (an object of class BPTEM) with elements:

Convergence	Potential scale reduction factor (PSRF) for each parameter (<code>parameter_psr</code>), Gelman multivariate scale reduction factor (<code>multivariate_psr</code>), and mean PSRF (<code>mean_psr</code>) to assess model convergence.
Model_Fit	Deviance (<code>deviance</code>), effective number of parameters (<code>pD</code>), and Deviance information criterion (<code>dic</code>) to assess model fit.
Fitted_Values	Vector giving the fitted value at each timepoint for each individual (same length as long data).
Parameter_Estimates	Data frame with posterior mean and 95% credible intervals for each model parameter.
Run_Time	Total run time for model fitting.
Full_MCMC_Chains	If <code>save_full_chains=TRUE</code> , raw MCMC chains from <code>rjags</code> .
Convergence_MCMC_Chains	If <code>save_conv_chains=TRUE</code> , raw MCMC chains from <code>rjags</code> but only for the parameters monitored for convergence.

Author(s)

Corissa T. Rohloff, Yadira Peralta

References

Peralta, Y., Kohli, N., Lock, E. F., & Davison, M. L. (2022). Bayesian modeling of associations in bivariate piecewise linear mixed-effects models. *Psychological Methods*, 27(1), 44–64. <https://doi.org/10.1037/met0000358>

Examples

```
## Not run:
# load simulated data
data(SimData_BPREM)
# plot observed data
plot_BEND(data = SimData_BPREM,
           id_var = "id",
           time_var = "time",
           y_var = "y1",
           y2_var = "y2")
# fit Bayes_BPREM()
results_bprem <- Bayes_BPREM(data = SimData_BPREM,
                             id_var = "id",
                             time_var = "time",
                             y1_var = "y1",
                             y2_var = "y2")

# result summary
summary(results_bprem)
# plot fitted results
plot_BEND(data = SimData_BPREM,
           id_var = "id",
           time_var = "time",
           y_var = "y1",
           y2_var = "y2",
           results = results_bprem)

## End(Not run)
```

Bayes_CREM

Bayesian Crossed Random Effects Model (CREM)

Description

Estimates a Bayesian crossed random effects models (CREM) for longitudinal data with dynamic group membership. Four different choices for functional forms are provided: linear, quadratic, exponential, and piecewise. See Rohloff et al. (2024) for more details.

Usage

```
Bayes_CREM(
  data,
  ind_id_var,
  cross_id_var,
  time_var,
  y_var,
  form = "linear",
  fixed_effects = NULL,
  iters_adapt = 5000,
  iters_burn_in = 50000,
  iters_sampling = 50000,
  thin = 15,
  save_full_chains = FALSE,
```

```

    save_conv_chains = FALSE
  )

```

Arguments

<code>data</code>	Data frame in long format, where each row describes a measurement occasion for a given individual. It is assumed that each individual has the same number of assigned timepoints (a.k.a., rows). There can be missingness in the outcome (<code>y_var</code>), but there cannot be missingness in time (<code>time_var</code>).
<code>ind_id_var</code>	Name of column that contains ids for individuals with repeated measures in a longitudinal dataset (e.g., students).
<code>cross_id_var</code>	Name of column that contains ids for the crossed factor (e.g., teachers).
<code>time_var</code>	Name of column that contains the time variable. This column cannot contain any missing values.
<code>y_var</code>	Name of column that contains the outcome variable. Missing values should be denoted by NA.
<code>form</code>	Name of the functional form. Options include: 'linear' (default), 'quadratic', 'exponential', 'piecewise'.
<code>fixed_effects</code>	(optional) Starting values for the fixed effects parameters.
<code>iters_adapt</code>	(optional) Number of iterations for adaptation of jags model (default = 5000).
<code>iters_burn_in</code>	(optional) Number of iterations for burn-in (default = 50000).
<code>iters_sampling</code>	(optional) Number of iterations for posterior sampling (default = 50000).
<code>thin</code>	(optional) Thinning interval for posterior sampling (default = 15).
<code>save_full_chains</code>	Logical indicating whether the MCMC chains from rjags should be saved (default = FALSE). Note, this should not be used regularly as it will result in an object with a large file size.
<code>save_conv_chains</code>	Logical indicating whether the MCMC chains from rjags should be saved but only for the parameters monitored for convergence (default = FALSE). This would be useful for plotting traceplots for relevant model parameters to evaluate convergence behavior. Note, this should not be used regularly as it will result in an object with a large file size.

Details

For more information on the model equation and priors implemented in this function, see Rohloff et al. (2024).

Note, this function differs from the above reference by estimating the covariances between the random effects parameters. The variance-covariance matrices of the individual and group random effects have a scaled inverse-Wishart prior (see Peralta et al., 2022).

Value

A list (an object of class CREM) with elements:

Convergence	Potential scale reduction factor (PSRF) for each parameter (<code>parameter_psrfr</code>), Gelman multivariate scale reduction factor (<code>multivariate_psrfr</code>), and mean PSRF (<code>mean_psrfr</code>) to assess model convergence.
-------------	---

Model_Fit	Deviance (deviance), effective number of parameters (pD), and Deviance information criterion (dic) to assess model fit.
Fitted_Values	Vector giving the fitted value at each timepoint for each individual (same length as long data).
Functional_Form	Functional form fitted.
Parameter_Estimates	Data frame with posterior mean and 95% credible intervals for each model parameter.
Run_Time	Total run time for model fitting.
Full_MCMC_Chains	If save_full_chains=TRUE, raw MCMC chains from rjags.
Convergence_MCMC_Chains	If save_conv_chains=TRUE, raw MCMC chains from rjags but only for the parameters monitored for convergence.

Author(s)

Corissa T. Rohloff

References

- Peralta, Y., Kohli, N., Lock, E. F., & Davison, M. L. (2022). Bayesian modeling of associations in bivariate piecewise linear mixed-effects models. *Psychological Methods*, 27(1), 44–64. <https://doi.org/10.1037/met0000358>
- Rohloff, C. T., Kohli, N., & Lock, E. F. (2024). Identifiability and estimability of Bayesian linear and nonlinear crossed random effects models. *British Journal of Mathematical and Statistical Psychology*. <https://doi.org/10.1111/bmsp.12334>

Examples

```
## Not run:
# load simulated data
data(SimData_PCREM)
# plot observed data
plot_BEND(data = SimData_PCREM,
           id_var = "id",
           time_var = "time",
           y_var = "y")
# fit Bayes_CREM()
results_pcrem <- Bayes_CREM(data = SimData_PCREM,
                           ind_id_var = "id",
                           cross_id_var = "teacherid",
                           time_var = "time",
                           y_var = "y",
                           form="piecewise")

# result summary
summary(results_pcrem)
# plot fitted results
plot_BEND(data = SimData_PCREM,
           id_var = "id",
           time_var = "time",
           y_var = "y",
           results = results_pcrem)
```

```
## End(Not run)
```

Bayes_PREM

Bayesian Piecewise Random Effects Model (PREM) + Extensions

Description

Estimates a Bayesian piecewise random effects model (PREM), with some useful extensions. There are three model options included in this function:

- PREM estimates a Bayesian piecewise random effects model with a latent number of change-points (default). Allows the inclusion of outcome-predictive covariates (CI-PREM).
- PREMM estimates a piecewise random effects mixture model for a given number of latent classes and a latent number of possible changepoints in each class.
- CI-PREMM estimates a covariate influenced piecewise random effects mixture model for a given number of latent classes and a latent number of possible changepoints in each class. Allows the inclusion of outcome- and/or class-predictive covariates. See Lock et al. (2018) and Lamm (2022) for more details.

Usage

```
Bayes_PREM(
  data,
  id_var,
  time_var,
  y_var,
  n_class = 1,
  max_cp = 2,
  class_predictive_vars = NULL,
  outcome_predictive_vars = NULL,
  scale_prior = "uniform",
  alpha = 1,
  cp_prior = "binomial",
  binom_prob = 0.5,
  iters_adapt = 1000,
  iters_burn_in = 20000,
  iters_sampling = 30000,
  thin = 15,
  save_full_chains = FALSE,
  save_conv_chains = FALSE
)
```

Arguments

data	Data frame in long format, where each row describes a measurement occasion for a given individual. It is assumed that each individual has the same number of assigned timepoints (a.k.a., rows). There can be missingness in the outcome (y_var), but there cannot be missingness in time (time_var).
------	---

<code>id_var</code>	Name of column that contains ids for individuals with repeated measures in a longitudinal dataset.
<code>time_var</code>	Name of column that contains the time variable. This column cannot contain any missing values.
<code>y_var</code>	Name of column that contains the outcome variable. Missing values should be denoted by NA.
<code>n_class</code>	Number of latent classes (default = 1). Note, CI-PREMM only allows for two classes.
<code>max_cp</code>	Maximum number of changepoints in each latent class (default = 2).
<code>class_predictive_vars</code>	Name(s) of column(s) that contain class-predictive covariates (time-invariant only). Give a vector of names if multiple covariates. Note, there cannot be any missingness in the covariates.
<code>outcome_predictive_vars</code>	Name(s) of column(s) that contain outcome-predictive covariates (time-varying or -invariant). Give a vector of names if multiple covariates. Note, there cannot be any missingness in the covariates.
<code>scale_prior</code>	Prior for the scale parameter for the hierarchical random effects. Options include: 'uniform' (scaled uniform prior; default) or 'hc' (scaled half-cauchy prior).
<code>alpha</code>	Concentration parameter for Dirichlet prior for latent classes (default = 1). This can be a vector of values corresponding to the number of classes (specified by <code>n_class</code>). Note, this is not used for CI-PGMM.
<code>cp_prior</code>	Prior for the number of changepoints in each class. Options include: 'binomial' (default) or 'uniform'.
<code>binom_prob</code>	Probability for binomial prior, if specified (default = 0.5).
<code>iters_adapt</code>	(optional) Number of iterations for adaptation of jags model (default = 1000).
<code>iters_burn_in</code>	(optional) Number of iterations for burn-in (default = 20000).
<code>iters_sampling</code>	(optional) Number of iterations for posterior sampling (default = 30000).
<code>thin</code>	(optional) Thinning interval for posterior sampling (default = 15).
<code>save_full_chains</code>	Logical indicating whether the MCMC chains from rjags should be saved (default = FALSE). Note, this should not be used regularly as it will result in an object with a large file size.
<code>save_conv_chains</code>	Logical indicating whether the MCMC chains from rjags should be saved but only for the parameters monitored for convergence (default = FALSE). This would be useful for plotting traceplots for relevant model parameters to evaluate convergence behavior. Note, this should not be used regularly as it will result in an object with a large file size.

Details

For more information on the model equation and priors implemented in this function, see Lamm et al. (2022; CI-PREMM) and Lock et al. (2018; PREMM).

Value

A list (an object of class PREM) with elements:

Convergence	Potential scale reduction factor (PSRF) for each parameter (parameter_psr), Gelman multivariate scale reduction factor (multivariate_psr), and mean PSRF (mean_psr) to assess model convergence.
Model_Fit	Deviance (deviance), effective number of parameters (pD), and Deviance information criterion (dic) to assess model fit.
Fitted_Values	Vector giving the fitted value at each timepoint for each individual (same length as long data).
Parameter_Estimates	Data frame with posterior mean and 95% credible intervals for each model parameter.
Run_Time	Total run time for model fitting.
Full_MCMC_Chains	If save_full_chains=TRUE, raw MCMC chains from rjags.
Convergence_MCMC_Chains	If save_conv_chains=TRUE, raw MCMC chains from rjags but only for the parameters monitored for convergence.

Class_Information contains a list with elements:

class_membership	Vector of length n with class membership assignments for each individual.
individ_class_probability	nxC matrix with each individual's probabilities of belonging to each class conditional on their class-predictive covariates (when applicable) and growth curve.
unconditional_class_probability	This output will differ based on which model was fit. For a PREM or CI-PREM, this will equal 1 as there is only one class. For a PREMM or CI-PREMM with only outcome-predictive covariates, this will be a vector of length C denoting the population probability of belonging to each class. For a CI-PREMM with class-predictive covariates, this will be a vector of length n denoting the probability of each individual belonging to the non-reference class (Class 2) based on their class-predictive covariates only.

Author(s)

Corissa T. Rohloff, Rik Lamm, Eric F. Lock

References

- Lamm, R. (2022). Incorporation of covariates in Bayesian piecewise growth mixture models. <http://conservancy.umn.edu/handle/11299/252533>
- Lock, E. F., Kohli, N., & Bose, M. (2018). Detecting multiple random changepoints in Bayesian piecewise growth mixture models. *Psychometrika*, 83(3), 733–750. <https://doi.org/10.1007/s11336-017-9594-5>

Examples

```

## Not run:
# load simulated data
data(SimData_PREM)
# plot observed data
plot_BEND(data = SimData_PREM,
          id_var = "id",
          time_var = "time",
          y_var = "y")

# PREM -----
# fit Bayes_PREM()
results_prem <- Bayes_PREM(data = SimData_PREM,
                          id_var = "id",
                          time_var = "time",
                          y_var = "y")

# result summary
summary(results_prem)
# plot fitted results
plot_BEND(data = SimData_PREM,
          id_var = "id",
          time_var = "time",
          y_var = "y",
          results = results_prem)

# CI-PREM -----
# fit Bayes_PREM()
results_ciprem <- Bayes_PREM(data = SimData_PREM,
                            id_var = "id",
                            time_var = "time",
                            y_var = "y",
                            outcome_predictive_vars = "outcome_pred_1")

# result summary
summary(results_ciprem)
# plot fitted results
plot_BEND(data = SimData_PREM,
          id_var = "id",
          time_var = "time",
          y_var = "y",
          results = results_ciprem)

# PREMM -----
# fit Bayes_PREM()
results_premm <- Bayes_PREM(data = SimData_PREM,
                           id_var = "id",
                           time_var = "time",
                           y_var = "y",
                           n_class = 2)

# result summary
summary(results_premm)
# plot fitted results
plot_BEND(data = SimData_PREM,
          id_var = "id",
          time_var = "time",
          y_var = "y",
          results = results_premm)

```

```

# CI-PREMM -----
# fit Bayes_PREM()
results_cipremm <- Bayes_PREM(data = SimData_PREM,
                             id_var = "id",
                             time_var = "time",
                             y_var = "y",
                             n_class = 2,
                             class_predictive_vars = c("class_pred_1", "class_pred_2"),
                             outcome_predictive_vars = "outcome_pred_1")

# result summary
summary(results_cipremm)
# plot fitted results
plot_BEND(data = SimData_PREM,
          id_var = "id",
          time_var = "time",
          y_var = "y",
          results = results_cipremm)

## End(Not run)

```

plot_BEND

Plot a BEND Model (PREM, CREM, BPREM)

Description

Generates a "spaghetti plot" of observed longitudinal trajectories for each individual. If the results from a BEND function are supplied, the trajectory defined by the mean parameters is shown in bold. If fitting a mixture (PREMM or CI-PREMM) or bivariate model (BPREM), the mean trajectories for classes or outcomes will be distinguished by color.

Usage

```

plot_BEND(
  data,
  id_var,
  time_var,
  y_var,
  y2_var = NULL,
  results = NULL,
  xlab = "X",
  ylab = "Y",
  colors = NULL,
  mean_colors = NULL,
  legend_pos = "topright",
  ...
)

```

Arguments

<code>data</code>	Data frame in long format, where each row describes a measurement occasion for a given individual. It is assumed that each individual has the same number of assigned timepoints (a.k.a., rows).
<code>id_var</code>	Name of column that contains ids for individuals with repeated measures in a longitudinal dataset.
<code>time_var</code>	Name of column that contains the time variable.
<code>y_var</code>	Name of column that contains the outcome variable.
<code>y2_var</code>	(for BPREM only) Name of column that contains the second outcome variable.
<code>results</code>	The output of BEND model to the data. If results=NULL, only a spaghetti plot of the data will be generated.
<code>xlab</code>	X-axis label for the generated plot.
<code>ylab</code>	Y-axis label for the generated plot.
<code>colors</code>	Colors for each class (PREMM or CI-PREMM) or outcome (BPREM). By default, up to 5 colors are provided in the following order: “blue” (class 1 and outcome 1), “red” (class 2 and outcome 2), “green” (class 3), “gold” (class 4), “gray” (class 5).
<code>mean_colors</code>	Colors for the trajectory defined by the mean parameters for each class (PREMM or CI-PREMM) or outcome (BPREM). By default, up to 5 colors are provided in the following order: “darkblue” (class 1 and outcome 1), “darkred” (class 2 and outcome 2), “darkgreen” (class 3), “gold4” (class 4), “darkgray” (class 5).
<code>legend_pos</code>	(optional) Option to change legend position (default = “topright”).
<code>...</code>	(optional) Other parameters to pass to the <code>plot()</code> function.

Author(s)

Corissa T. Rohloff

Examples

```
## Not run:
# load simulated data
data(SimData_PREM)
# plot observed data
plot_BEND(data = SimData_PREM,
           id_var = "id",
           time_var = "time",
           y_var = "y")

# fit Bayes_PREM()
results_prem <- Bayes_PREM(data = SimData_PREM,
                           id_var = "id",
                           time_var = "time",
                           y_var = "y")

# plot fitted results
plot_BEND(data = SimData_PREM,
           id_var = "id",
           time_var = "time",
           y_var = "y",
           results = results_prem)
```

```
## End(Not run)
```

SimData_BPTEM	<i>Simulated data for a BPTEM</i>
---------------	-----------------------------------

Description

Simulated data for a bivariate piecewise random effects model (BPTEM) with 7 timepoints collected on 30 individuals.

Usage

```
data(SimData_BPTEM)
```

Format

A data frame with 210 rows and 4 variables.

Details

- id ID for each individual.
- time Timepoints for each individual.
- y1 Outcome 1.
- y2 Outcome 2.

SimData_PCTEM	<i>Simulated data for a PCTEM</i>
---------------	-----------------------------------

Description

Simulated data for a piecewise crossed random effects model (PCTEM) with 7 timepoints collected on 30 individuals.

Usage

```
data(SimData_PCTEM)
```

Format

A data frame with 210 rows and 4 variables.

Details

- id ID for each individual.
- teacherid ID for each teacher.
- time Timepoints for each individual.
- y Outcome.

SimData_PREM	<i>Simulated data for a PREM + Extensions</i>
--------------	---

Description

Simulated data for a piecewise random effects model (PREM) and useful extensions (CI-PREM, PREMM, CI-PREMM) with 18 timepoints collected on 30 individuals.

Usage

```
data(SimData_PREM)
```

Format

A data frame with 540 rows and 6 variables.

Details

- id ID for each individual.
- time Timepoints for each individual.
- y Outcome.
- class_pred_1 First class predictive covariate (time-invariant).
- class_pred_2 Second class predictive covariate (time-invariant).
- outcome_pred_1 Outcome predictive covariate (time-varying).

summary.BPREM	<i>Summarize the results of a bivariate piecewise random effects model (BPREM)</i>
---------------	--

Description

Provides a summary of a BPREM model, as returned by Bayes_BPREM().

Usage

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

Arguments

object	An object of class "BPREM" (returned by Bayes_BPREM(...)).
...	Additional arguments.

Value

Prints estimates for key parameters in the BPREM. Also returns a list of these values.

Author(s)

Corissa T. Rohloff

Examples

```
## Not run:
# load simulated data
data(SimData_BPREM)
# fit Bayes_BPREM()
results_bprem <- Bayes_BPREM(data = SimData_BPREM,
                             id_var = "id",
                             time_var = "time",
                             y1_var = "y1",
                             y2_var = "y2")

# result summary
summary(results_bprem)

## End(Not run)
```

summary.CREM

*Summarize the results of a crossed random effects model (CREM)***Description**

Provides a summary of a CREM model, as returned by Bayes_CREM().

Usage

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

Arguments

object	An object of class "CREM" (returned by Bayes_CREM(...)).
...	Additional arguments.

Value

Prints estimates for key parameters in the CREM. Also returns a list of these values.

Author(s)

Corissa T. Rohloff


```
summary(results_prem)
```

```
## End(Not run)
```

Index

* **datasets**

SimData_BPTEM, [13](#)

SimData_PCTEM, [13](#)

SimData_PREM, [14](#)

Bayes_BPTEM, [2](#)

Bayes_CTEM, [4](#)

Bayes_PREM, [7](#)

plot_BEND, [11](#)

SimData_BPTEM, [13](#)

SimData_PCTEM, [13](#)

SimData_PREM, [14](#)

summary_BPTEM, [14](#)

summary_CTEM, [15](#)

summary_PREM, [16](#)