Package ‘lddmm’

January 17, 2024

**Type** Package

**Title** Longitudinal Drift-Diffusion Mixed Models (LDDMM)

**Version** 0.4.2

**Date** 2024-01-17

**Description** Implementation of the drift-diffusion mixed model for category learning as described in Paulon et al. (2021) <doi:10.1080/01621459.2020.1801448>.

**Depends** R (>= 4.1.0)

**Language** en-US

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** Rcpp (>= 1.0.6), gtools, LaplacesDemon, dplyr, plyr, tidyr, ggplot2, latex2exp, reshape2, RColorBrewer

**LazyData** true

**LinkingTo** Rcpp, RcppArmadillo, RcppProgress, rgen

**RoxygenNote** 7.2.3

**Suggests** rmarkdown, knitr

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Giorgio Paulon [aut, cre], Abhra Sarkar [aut, ctb]

**Maintainer** Giorgio Paulon <giorgio.paulon@utexas.edu>

**Repository** CRAN

**Date/Publication** 2024-01-17 20:10:02 UTC

---

**R topics documented:**

- `B_basis` .................................................. 2
- `compute_WAIC` ........................................... 2
- `data` ..................................................... 3

---
compute_WAIC

Description
Function to compute the Watanabe-Akaike information criterion (Gelman, Hwang, Vehtari, 2014), which estimates the expected out-of-sample-prediction error using a bias-corrected adjustment of within-sample error.

Usage
compute_WAIC(model_fit)

B_basis

Spline Basis Functions

Description
Construct the J basis functions for the splines evaluated on a grid.

Usage
B_basis(xgrid, knots)

Arguments

xgrid grid where we want to evaluate the spline functions (vector of length n)
knots vector of knots for the splines (vector of length K)

Value
n x (K+1) - matrix representing the value of each basis function evaluated on xgrid
data

Arguments

model_fit results of a model fit from the lddmm function

Value

A scalar indicating the WAIC (smaller WAIC denotes better fit)

Description

A toy dataset in the correct format for the LDDMM function call. This dataset has two possible response categories.

Usage
data

Format

A data frame with 24,254 rows and 6 columns

Details

• subject: vector of size n containing the participant labels
• block: vector of size n containing the training blocks (longitudinal units)
• s: vector of size n containing the stimuli
• d: vector of size n containing the decisions
• r_time: vector of size n containing the response times (log transformed)
• cens: vector of size n containing the censoring indicators (1 censored, 0 non censored)

extract_post_draws Parameter posterior draws

Description

Function to extract the posterior draws of the parameters of interest from a lddmm fit object.

Usage

extract_post_draws(data, fit, par = c("drift", "boundary"))
extract_post_mean

Arguments

data dataframe with the following columns:
  • subject: vector of size n containing the participant labels
  • block: vector of size n containing the training blocks (longitudinal units)
  • s: vector of size n containing the stimuli
  • d: vector of size n containing the decisions
  • r_time: vector of size n containing the response times
  • cens: vector of size n containing the censoring indicators (1 censored, 0 non censored)

fit fit from the lddmm function

par parameter to output ('drift', or 'boundary')

Value

Matrix with the following columns:
  • subject: participant labels
  • block: training blocks
  • draw: iteration of the MCMC estimates
  • par_s_d, ...: posterior draws for the requested parameters

Description

Function to extract the posterior means of the parameters of interest from a lddmm fit object.

Usage

extract_post_mean(data, fit, par = c("drift", "boundary"))

Arguments

data dataframe with the following columns:
  • subject: vector of size n containing the participant labels
  • block: vector of size n containing the training blocks (longitudinal units)
  • s: vector of size n containing the stimuli
  • d: vector of size n containing the decisions
  • r_time: vector of size n containing the response times
  • cens: vector of size n containing the censoring indicators (1 censored, 0 non censored)

fit fit from the lddmm function

par parameter to output ('drift', or 'boundary')
**H_ball**

**Value**

Matrix with the following columns:
- subject: participant labels
- block: training blocks
- par_s_d, ...: posterior means for the requested parameters

**Description**

Computes the Hamming Ball centered at x with radius r.

**Usage**

\[ H\_ball(x, S, r) \]

**Arguments**

- \( x \): center of the Hamming Ball
- \( S \): number of states
- \( r \): radius of the Hamming Ball

**Value**

Hamming Ball

---

**LDDMM**

*Drift Diffusion Model Fit*

**Description**

Main function for the Gibbs sampler for the drift-diffusion model. Note that priors are noninformative and calibrated so that, for the most stable performance, the response times (variable \( r\_time \) in the data dataframe) should lie between 0 and 10.

**Usage**

\[
\text{LDDMM(}
    \text{data,}
    \text{hypers,}
    \text{cluster = TRUE,}
    \text{boundaries = "flexible",}
    \text{Niter = 5000,}
    \text{burnin = 2000,}
    \text{thin = 5}
\text{)}
\]
Arguments

data dataframe with the following columns:
- subject: vector of size n containing the participant labels
- block: vector of size n containing the training blocks (longitudinal units)
- s: vector of size n containing the stimuli
- d: vector of size n containing the decisions
- r_time: vector of size n containing the response times. To avoid numerical issues, the unit of measurement should be such that the numerical values of most response times should lie between 0 and 10
- cens: vector of size n containing the censoring indicators (1 censored, 0 non censored)

hypers hyperparameters of the MCMC: list containing "s_sigma_mu" and "s_sigma_b", which are the smoothness parameters for drifts and boundaries, respectively)

cluster should clustering be used? (default = TRUE)

boundaries whether to fit the unrestricted model (flexible), assume constant boundaries over time (constant) or fix the boundaries to the same level across predictors (fixed)

Niter total number of iterations

burnin burnin of the chain

thin thinning factor

Value

List with the following MCMC posterior samples:
- post_mean_delta: posterior samples for the population offset parameters
- post_mean_mu: posterior samples for the population drift parameters
- post_mean_b: posterior samples for the population boundary parameters
- post_ind_delta: posterior samples for the individual offset parameters
- post_ind_mu: posterior samples for the individual drift parameters
- post_ind_b: posterior samples for the individual boundary parameters
- sigma2_mu_us: posterior samples for the random effects drift smoothness parameters
- sigma2_mu_ua: posterior samples for the random effects drift variance parameters
- sigma2_b_us: posterior samples for the random effects boundary smoothness parameters
- sigma2_b_ua: posterior samples for the random effects boundary variance parameters
- sigma2_1_mu: posterior samples for the drift smoothness parameters
- sigma2_1_b: posterior samples for the boundary smoothness parameters
- pred_ans: predicted population-level categories
- pred_time: predicted population-level response times
- pred_ans_ind: predicted individual-level categories
- pred_time_ind: predicted individual-level response times
**log_likelihood**

*Log-likelihood computation*

**Description**

Compute the log-likelihood for the drift-diffusion model, including the censored data contribution.

**Usage**

```r
log_likelihood(tau, mu, b, delta, cens, D, log)
```

**Arguments**

- `tau`: vector of size n containing the response times
- `mu`: matrix of size (n x d1) containing the drift parameters corresponding to the n response times for each possible d1 decision
- `b`: matrix of size (n x d1) containing the boundary parameters corresponding to the n response times for each possible d1 decision
- `delta`: vector of size n containing the offset parameters corresponding to the n response times
- `cens`: vector of size n containing censoring indicators (1 censored, 0 not censored) corresponding to the n response times
- `D`: (n x 2) matrix whose first column has the n input stimuli, and whose second column has the n decision categories
- `log`: should the results be returned on the log scale?

**log_likelihood_ind**

*Log-likelihood computation for a single observation*

**Description**

Compute the log-likelihood for the drift-diffusion model, including the censored data contribution, for a single observation.

**Usage**

```r
log_likelihood_ind(tau, mu, b, delta, cens, D)
```
plot_accuracy

Arguments

- tau: vector of size n containing the response times
- mu: matrix of size (n x d1) containing the drift parameters corresponding to the n response times for each possible d1 decision
- b: matrix of size (n x d1) containing the boundary parameters corresponding to the n response times for each possible d1 decision
- delta: vector of size n containing the offset parameters corresponding to the n response times
- cens: vector of size n containing censoring indicators (1 censored, 0 not censored) corresponding to the n response times
- D: (n x 2) matrix whose first column has the n input stimuli, and whose second column has the n decision categories

plot_accuracy(data)

Description

Plot the accuracy of the raw data.

Usage

plot_accuracy(data)

Arguments

data: dataframe with the following columns:

- subject: vector of size n containing the participant labels
- block: vector of size n containing the training blocks (longitudinal units)
- s: vector of size n containing the stimuli
- d: vector of size n containing the decisions
- r_time: vector of size n containing the response times
- cens: vector of size n containing the censoring indicators (1 censored, 0 non censored)

Value

Individual and population level raw accuracies
**plot_post_pars** *Plot posterior estimates*

**Description**

Function to plot the posterior mean and credible intervals of the parameters of interest from a lddmm fit object.

**Usage**

```r
plot_post_pars(data, fit, par = c("drift", "boundary"))
```

**Arguments**

- `data` dataframe with the following columns:
  - `subject`: vector of size n containing the participant labels
  - `block`: vector of size n containing the training blocks (longitudinal units)
  - `s`: vector of size n containing the stimuli
  - `d`: vector of size n containing the decisions
  - `r_time`: vector of size n containing the response times
  - `cens`: vector of size n containing the censoring indicators (1 censored, 0 non censored)
- `fit` fit from the lddmm function
- `par` parameter to output ('drift', or 'boundary')

**Value**

Posterior mean and 95% CI

---

**plot_RT** *Descriptive plots*

**Description**

Plot the mean response times of the raw data.

**Usage**

```r
plot_RT(data)
```
Arguments
data dataframe with the following columns:
• subject: vector of size n containing the participant labels
• block: vector of size n containing the training blocks (longitudinal units)
• s: vector of size n containing the stimuli
• d: vector of size n containing the decisions
• r_time: vector of size n containing the response times
• cens: vector of size n containing the censoring indicators (1 censored, 0 non censored)

Value
Population level raw response times

P_smooth1
Spline Penalty Matrix

Description
Construct the covariance matrix P of the smoothness inducing prior for the spline coefficients

Usage
P_smooth1(K)

Arguments
K Number of spline knots

Value
Covariance of the smoothness inducing prior (penalizing first differences in the spline coefficients)
Index

* datasets
  data, 3
  B_basis, 2
  compute_WAIC, 2
  data, 3
  extract_post_draws, 3
  extract_post_mean, 4
  H_ball, 5
  LDDMM, 5
  log_likelihood, 7
  log_likelihood_ind, 7
  P_smooth1, 10
  plot_accuracy, 8
  plot_post_pars, 9
  plot_RT, 9