Package 'RegDDM'

December 5, 2024

Title Generalized Linear Regression with DDM

Version 1.0

Description Drift-Diffusion Model (DDM) has been widely used to model binary decision-making tasks, and many research studies the relationship between DDM parameters and other characteristics of the subject. This package uses 'RStan' to perform generalized liner regression analysis over DDM parameters via a single Bayesian Hierarchical model. Compared to estimating DDM parameters followed by a separate regression model, 'RegDDM' reduces bias and improves statistical power.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

URL `https://github.com/biorabbit/RegDDM`

BugReports `https://github.com/biorabbit/RegDDM/issues`

Imports rstan, stringr, dplyr, tidyr, purrr, rtdists, rlang, stats,

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Zekai Jin [aut, cre]

Maintainer Zekai Jin <`Jin.Zekai@nyspi.columbia.edu`>

Repository CRAN

Date/Publication 2024-12-05 19:00:12 UTC

Contents

`generate_sim_data` *Generate simulated binary decision data using DDM*

Description

This function generates a simulated dataset under different configurations It can be used to test the performance and functionality of **RegDDM**. The outcome variable is `y`, which is influenced by different variables.

Usage

```
generate_sim_data(
  N = 30,
  n_each = 100,
  n_xvar = 2,
  beta_0 = 0,
  beta_c1 = 0,
  beta_c2 = 0,
  beta_v_0 = 0,
  beta_v_x1 = 0,
  beta_v_x2 = 0,
  sigma_y = 1,
  sigma_v = 0,
  y_family = "gaussian"
)
```

Arguments

`N`	Number of subjects.
`n_each`	Number of trials per subject
`n_xvar`	Number of trial-level variables influencing drift rate
`beta_0`	Intercept
`beta_c1`	Slope of c1
`beta_c2`	Slope of c2
`beta_v_0`	Slope of v_0
`beta_v_x1`	Slope of v_x1
`beta_v_x2`	Slope of v_x2

`sigma_y` Standard deviation of error term of y, Only used when `y_family` is "gaussian"

`sigma_v` Contaminant level for drift rate v.

`y_family` Family of distribution of y. Can be either "gaussian", "bernoulli" or "poisson"

Value

A named list with four elements. `data1_true` and `data2_true` are true values of DDM parameters of each subject and trial. `data1` and `data2` removed those hidden variables.

Examples

```
sim_data = generate_sim_data()
sim_data$data1
sim_data$data2
```

`get_stan_fit` *Get the rstan fit of regddmfit objects*

Description

Get the `stanfit` object of the `regddmfit` object to perform further analysis and diagnosis.

Usage

```
get_stan_fit(fit)
```

Arguments

`fit` A `regddmfit` object to summary

Value

A `stanfit` object.

print.regddmfit *Printed summary of* **RegDDM** *fit object*

Description

Summarize the posterior distributions of estimated regression coefficients and print necessary information.

Usage

```
## S3 method for class 'regddmfit'
print(x, digits = 3, ...)
```

Arguments

`x`	An `regddmfit` object to print
`digits`	digits of the output results. Default value is 3.
`...`	Unused...

Value

No values are returned.

See Also

`summary.regddmfit` Table summaries of the `regddmfit` object.

regddm *Bayesian hierachical generalized linear regression using Drift-Diffusion Model*

Description

`regddm` makes it easy to fit a single Bayesian hierarchical drift-diffusion model (DDM) that estimates the DDM parameters of each subject and uses the estimated parameters as variables in a generalized linear regression.

Usage

```
regddm(
  data1,
  data2,
  model = list(),
  family = "gaussian",
  init = "default",
  prior = TRUE,
  stan_filename = "",
  gen_model = TRUE,
  fit_model = TRUE,
  warmup = 500,
  iter = 1000,
  chains = 4,
  cores = 4,
  ...
)
```

Arguments

`data1` Subject-level dataframe with column such as age and gender. It must contain an `id` column unique for each subject.

`data2` Trial-level dataframe. It must contain three columns: `id`, `response` and `rt`. It can also contain additional trial-level variables such as experiment condition.

`model` A list containing 0-5 formulas, specifying the dependence structure between variables.

`family` Family of distribution of `y`. Can be `gaussian`, `bernoulli` or `poisson`.

`init` Either `default` or other values supported by `stan` function of **RStan**

`prior` A losigtic value, specifying whether or not to use default prior for DDM parameters. By default, `prior` = TRUE.

`stan_filename` A string specifying the automatically generated `stan` file name. By default, an empty string `''` is provided. A temporary file will be created and deleted after the model is fit.

`gen_model` A logistic value indicating weather or not to generate the model. If not, RegDDM will not generate the code but use the existing `stan` code from `stan_filename` instead.

`fit_model` A logistic value indicating weather or not to fit the model. If not, RegDDM will only generate the code and return an unfitted `regddmfit` object.

`warmup` Number of warm-up iterations. Default is 500.

`iter` Number of iterations, which must be greater than `warmup`. Default value is 1000.

`chains` Number of chains to run for diagnosis. Default value is 4.

`cores` Number of cores to run the chains. It is best to make `cores` = `chains`. Default value is 4.

`...` Other parameters sent to `stan` function of **RStan**.

Value

A regddmfit object.

References

To be added

Examples

```
# Note: each example takes about 20 minutes to run. During this period, you
# may not be able to open/save files or see the progress. To prevent this,
# it is recommended to copy, paste and run the example code in the console.

## Not run:
# Example analysis over the simulated tutorial dataset.
data(regddm_tutorial)
model = list(v ~ x1, y ~ v_0 + v_x1 + c1)
fit1 = regddm(
  regddm_tutorial$data1,
  regddm_tutorial$data2,
  model,
  stan_filename = ""
)
print(fit1)

# Alternatively, subjects' DDM parameters can be used as the outcome.
model = list(v ~ x1, v_x1 ~ y + c1)
fit2 = regddm(
  regddm_tutorial$data1,
  regddm_tutorial$data2,
  model,
  stan_filename = ""
)
print(fit2)

## End(Not run)
```

regddmfit — *Class of models fitted by* **RegDDM**

Description

regddmfit is an S3 object sotring the fitted models of **RegDDM**. It contains information used to fit the model and the resulting stanfit and can be summarized and printed using summary and print.

Usage

```
regddmfit(data1, data2, model, family, stan_fit)
```

Arguments

`data1`	Subject-level data frame.
`data2`	Trial-level data frame.
`model`	A list containing 0-5 formulas, specifying the dependence structure between variables.
`family`	Family of distribution of the outcome.
`stan_fit`	Fitted `stan` model.

Details

Use `methods(class = "regddmfit")` for a list of available methods.

`regddm_tutorial` *Simulated example dataset*

Description

This is the same simulated dataset used in the tutorial part of the RegDDM paper.

$v = x_1 \times v_{x_1}$

$y \sim N(1 + 1 \times v_{x_1} + 1 \times c_1, 1)$

Usage

```
regddm_tutorial
```

Format

An object of class `list` of length 2.

`summary.regddmfit` *Summary of* **RegDDM** *fit object*

Description

Summarize the posterior distributions of estimated parameters and group them into four categories.

Usage

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

Arguments

`object`	A `regddmfit` object to summary
`...`	parameters passed to `summary,stanfit-method`

Value

The summary method returns a named list of `glm_coefficiets`, `subject_ddm_param`, `group_param`, and `missing_value`. Each element is a `tibble` data frame of posterior summary statistics of the regression coefficient, DDM parameter of each subject, group mean and standard deviation of DDM parameters and covariates, plus the estimated missing values.

See Also

`print.regddmfit` a printed summary of the `regddmfit` object.

Index