Package ‘calmr’

March 14, 2024

Title Canonical Associative Learning Models and their Representations

Version 0.6.1

Description Implementations of canonical associative learning models, with tools to run experiment simulations, estimate model parameters, and compare model representations. Experiments and results are represented using S4 classes and methods.

License GPL (>= 3)

URL https://github.com/victor-navarro/calmr,
https://victornavarro.org/calmr/

BugReports https://github.com/victor-navarro/calmr/issues

Depends R (>= 3.5)

Imports data.table, future, future.apply, GA, ggnetwork, ggplot2, grid, methods, network, patchwork, progressr, rlang, stats

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.3

Collate 'HD2022.R' 'HDI2020.R' 'MAC1975.R' 'PKH1982.R' 'RAND.R'
'ANCCR.R' 'rsa_functions.R' 'RW1972.R' 'SM2007.R'
'compare_models.R' 'data.R' 'fit_helpers.R' 'fit_model.R'
'model_parsers.R' 'model_plots.R' 'model_graphs.R'
'model_support_functions.R' 'parse_design.R' 'run_experiment.R'
'phase_parser.R' 'information_functions.R' 'make_experiment.R'
'assertions.R' 'get_parameters.R' 'get_design.R'
'heidi_helpers.R' 'anccr_helpers.R' 'calmr_verbosity.R'
'parallel_helpers.R' 'class_model.R' 'class_design.R'
'class_result.R' 'class_experiment.R' 'class_rsa.R'
'class_fit.R'

NeedsCompilation no

Author Victor Navarro [aut, cre, cph]

Maintainer Victor Navarro <navarrov@cardiff.ac.uk>

Repository CRAN

Date/Publication 2024-03-14 20:10:05 UTC

R topics documented:

`CalmrDesign-class` *S4 class for calmr designs*

Description

S4 class for calmr designs

Slots

`design`**:** A list containing design information.

`mapping`**:** A list containing the object mapping.

`raw_design`**:** The original data.frame.

`augmented`**:** Whether the object has been augmented.

`CalmrDesign-methods` *CalmrDesign methods*

Description

S4 methods for `CalmrDesign` class.

Usage

```
## S4 method for signature 'CalmrDesign'
show(object)

## S4 method for signature 'CalmrDesign'
mapping(object)

## S4 method for signature 'CalmrDesign'
trials(object)
```

Arguments

`object`	A `CalmrDesign` object

Value

`show()` returns NULL (invisibly).

`mapping()` returns a list with trial mappings.

`trials()` returns NULL (invisibly).

CalmrExperiment-class *S4 class for calmr experiments.*

Description

S4 class for calmr experiments.

Slots

`design`: A CalmrDesign object.

`model`: A string specifying the model used.

`groups`: A string specifying the groups in the design.

`parameters`: A list with the parameters used, per group.

`experiences`: A list with the experiences for the model.

`results`: A CalmrExperimentResult object.

`.model`: Internal. The model associated with the iteration.

`.group`: Internal. The group associated with the iteration.

`.iter`: Internal. The iteration number.

See Also

CalmrExperiment-methods

CalmrExperiment-methods *CalmrExperiment methods*

Description

S4 methods for `CalmrExperiment` class.

Usage

```
## S4 method for signature 'CalmrExperiment'
show(object)

## S4 method for signature 'CalmrExperiment'
design(x)

## S4 method for signature 'CalmrExperiment'
trials(object)

## S4 method for signature 'CalmrExperiment'
```

```
parameters(x)

## S4 replacement method for signature 'CalmrExperiment'
parameters(x) <- value

## S4 method for signature 'CalmrExperiment'
experiences(x)

## S4 method for signature 'CalmrExperiment'
results(object)

## S4 method for signature 'CalmrExperiment'
raw_results(object)

## S4 method for signature 'CalmrExperiment'
parsed_results(object)

## S4 method for signature 'CalmrExperiment'
length(x)

## S4 method for signature 'CalmrExperiment'
parse(object, outputs = NULL)

## S4 method for signature 'CalmrExperiment'
aggregate(x, outputs = NULL)

## S4 method for signature 'CalmrExperiment'
plot(x, type = NULL)

## S4 method for signature 'CalmrExperiment'
graph(x, ...)
```

Arguments

`object, x`	A `CalmrExperiment` object.
`value`	A list of parameters (or list of parameter lists).
`outputs`	A character vector specifying the model outputs to parse.
`type`	A character vector specifying the type(s) of plots to create. Defaults to NULL. See supported_plots.
`...`	Extra arguments passed to `calmr_model_graph()`.

Value

`show()` returns NULL (invisibly).

`design()` returns the `CalmrDesign` contained in the object.

`trials()` returns NULL (invisibly).

`parameters()` returns the list of parameters contained in the object.

`parameters()<-` returns the object after updating parameters.

`experiences()` returns a list of `data.frame` objects containing model training routines.

`results()` returns a `data.table` objects with aggregated results.

`raw_results()` returns a list with raw model results.

`parsed_results()` returns a list of `data.table` objects with parsed results.

`length()` returns an integer specifying the total length of the experiment (groups by iterations).

`parse()` returns the object after parsing raw results.

`aggregate()` returns the object after aggregating parsed results.

`plot()` returns a list of 'ggplot' plot objects.

`graph()` returns a list of 'ggplot' plot objects.

`CalmrExperimentResult-class` *S4 class for calmr experiment results*

Description

S4 class for calmr experiment results

Slots

aggregated_results A list of `data.table` objects with aggregated results.

parsed_results A list containing `data.table` objects with parsed results.

raw_results A list with raw model outputs.

`CalmrFit-class` *S4 class for calmr Fit*

Description

S4 class for calmr Fit

Slots

`nloglik`: Numeric. Negative log likelihood of the fit

`best_pars`: Numeric. Best fitting parameters

`model_pars`: Numeric. Parameters used in the model function

`link_pars`: Numeric. Parameters used in the link function

`data`: Numeric. Data used for fit

`model_function`: Function. Model function

`link_function`: Function. Link function

`ll_function`: Function. Objective function (usually nloglikelihood)

`optimizer_options`: List. Options used for the optimizer

`extra_pars`: List. Extra parameters passed to the fit call (...)

See Also

CalmrFit-methods

CalmrFit-methods *CalmrFit methods*

Description

S4 methods for CalmrFit class.

Usage

```
## S4 method for signature 'CalmrFit'
show(object)

## S4 method for signature 'CalmrFit'
predict(object, type = "response", ...)

## S4 method for signature 'CalmrFit'
NLL(object)

## S4 method for signature 'CalmrFit'
AIC(object, k = 2)

## S4 method for signature 'CalmrFit'
BIC(object)
```

Arguments

object	A CalmrFit object.
type	A string specifying the type of prediction to generate.
...	Extra named arguments.
k	Penalty term for AIC method.

Details

With type = "response", the predict() function passed model responses to the link function used to fit the model.

The AIC is defined as 2*k - 2*-NLL, where k is a penalty term and NLL is the negative log likelihood of the model.

The BIC is defined as p*log(n) - 2*-NLL, where p is the number of parameters in the model and n is the number of observations

Value

- show() returns NULL (invisibly).
- predict() returns a numeric vector.
- NLL() returns the negative log likelihood of the model.
- AIC() returns the Akaike Information Criterion (AIC) of the model.
- BIC() returns the Bayesian Information Criterion (BIC) of the model.

CalmrResult-class *S4 class for calmr results*

Description

S4 class for calmr results

Slots

aggregated_results A list of data.table objects with aggregated results.

parsed_results A list containing data.table objects with parsed results.

raw_results A list with raw model outputs.

See Also

CalmrResults-methods

CalmrResult-methods *CalmrResult methods*

Description

S4 methods for CalmrResults class.

Usage

```
## S4 method for signature 'CalmrResult'
show(object)
```

Arguments

object A CalmrResults object.

Value

- show() returns NULL (invisibly).

CalmrRSA-class *S4 class for calmr representational similarity analysis*

Description

S4 class for calmr representational similarity analysis

Slots

`corr_mat`**:** An array containing the correlation matrix

`distances`**:** A list of pairwise distance matrices

`args`**:** A list of the arguments used to create the object.

`test_data`**:** A list with permutation data, only populated after testing the object.

CalmrRSA-methods *CalmrRSA methods*

Description

S4 methods for `CalmrRSA` class.

Usage

```
## S4 method for signature 'CalmrRSA'
show(object)

## S4 method for signature 'CalmrRSA'
test(object, n_samples = 1000, p = 0.95)

## S4 method for signature 'CalmrRSA'
plot(x)
```

Arguments

`object, x`	A `CalmrRSA` object.
`n_samples`	The number of samples for the permutation test (default = 1e3)
`p`	The critical threshold level for the permutation test (default = 0.95)

Value

- `show()` returns NULL (invisibly).
- `test()` returns the `CalmrRSA` object with permutation test data.
- `plot()` returns a list of 'ggplot' plot objects.

calmr_model_graph *Create a graph with calmr data*

Description

Create a graph with calmr data

Usage

```
calmr_model_graph(
  x,
  loops = TRUE,
  limits = max(abs(x$value)) * c(-1, 1),
  colour_key = FALSE,
  t = max(x$trial),
  options = get_graph_opts()
)
```

Arguments

x	A data.frame-like with data to use in the plot. Contains a column named value.
loops	Logical. Whether to draw arrows back and forth
limits	Numerical. Limits for color scale. Defaults to max(abs(x$value))*c(-1,1).
colour_key	Logical. Whether to show the color key
t	The trial from which weights are obtained (defaults to the maximum trial in the data).
options	A list with graph options, as returned by get_graph_opts().
trial	Numerical. The trial to graph.

Value

A 'ggplot' object

Note

You should probably be getting graphs via the graph() method for CalmrExperiment.

`calmr_model_plot` *Create a plot with calmr data*

Description

Create a plot with calmr data

Usage

```
calmr_model_plot(dat, type)
```

Arguments

`dat`	An `data.table` containing aggregated data from a CalmrExperiment
`type`	A character specifying the type of plot.

Value

A 'ggplot' object.

Note

You should probably be getting plots via the `plot()` method for CalmrExperiment.

`calmr_verbosity` *Set verbosity options for calmr*

Description

Whether to show verbosity messages and progress bars

Usage

```
calmr_verbosity(verbose)
```

Arguments

`verbose`	A logical

Value

The list of progressr handlers (invisibly).

Note

Progress bars are handled by the progressr package. This is just a convenience function.

See package 'progressr' for further details.

compare_models *Run models given a set of parameters*

Description

Run models given a set of parameters

Usage

```
compare_models(x, models = NULL, ...)
```

Arguments

`x` A list of CalmrExperiment objects or a design data.frame.

`models` A character vector of length m, specifying the models to run. Ignored if x is a list of CalmrExperiment objects.

`...` Arguments passed to make_experiment.

Value

A list of CalmrExperiment objects

Examples

```
# By making experiment beforehand (recommended)
df <- get_design("blocking")
models <- c("HD2022", "RW1972", "PKH1982")
exps <- lapply(models, function(m) {
  make_experiment(df,
    parameters = get_parameters(df, model = m),
    model = m
  )
})
comp <- compare_models(exps)

# By passing minimal arguments (not recommended; default parameters)
comp <- compare_models(df, models = models)
```

fit_model *Fit model to data*

Description

Obtain MLE estimates for model, given data.

Usage

```
fit_model(data, model_function, optimizer_options, file = NULL, ...)
```

Arguments

`data` A numeric vector containing data to fit model against.

`model_function` A function that runs the model and returns data.frame of value, organized as in `data`.

`optimizer_options` A list with options for the optimizer, as returned by get_optimizer_opts.

`file` A path to save the model fit. If the arguments to the fit call are found to be identical to those in the file, the model just gets loaded.

`...` Extra parameters passed to the optimizer call.

Value

A CalmrFit object

Note

See the calmr_fits vignette for examples

See Also

`get_optimizer_opts()`

Examples

```
# Make some fake data
df <- data.frame(g = "g", p1 = "3A>(US)", r1 = TRUE)
pars <- get_parameters(df, model = "RW1972")
pars$alphas["US"] <- 0.9
exper <- make_experiment(df, parameters = pars, model = "RW1972")
res <- run_experiment(exper, outputs = "rs")
rs <- results(res)$rs$value

# define model function
model_fun <- function(p, ex) {
  np <- parameters(ex)
  np[[1]]$alphas[] <- p
  parameters(ex) <- np
  results(run_experiment(ex))$rs$value
}

# Get optimizer options
optim_opts <- get_optimizer_opts(
  model_pars = names(pars$alphas),
  ll = rep(.05, 2), ul = rep(.95, 2),
  optimizer = "optim", family = "identity"
)
```

```
optim_opts$initial_pars[] <- rep(.6, 2)

fit_model(rs, model_fun, optim_opts,
  ex = exper, method = "L-BFGS-B",
  control = list(maxit = 1)
)
```

---

get_design *Get basic designs*

Description

Get basic designs

Usage

```
get_design(design_name = NULL)
```

Arguments

design_name	A string specifying a design name (default = NULL)

Value

If design_name is not NULL, a data.frame containing the design. Otherwise, a list containing all available designs.

See Also

parse_design()

Examples

```
names(get_design())
get_design("blocking")
```

`get_graph_opts` *Get options for calmr graph*

Description

Get options for calmr graph

Usage

```
get_graph_opts(graph_size = "small")
```

Arguments

`graph_size`	A string (either "small" or "large"). to return default values for small or large graphs

Value

A list with graph options, to be passed to `ggnetwork::geom_nodes()`.

`get_optimizer_opts` *Get optimizer options*

Description

Get optimizer options

Usage

```
get_optimizer_opts(
  model_pars,
  initial_pars = rep(NA, length(model_pars)),
  ll = rep(NA, length(model_pars)),
  ul = rep(NA, length(model_pars)),
  optimizer = NULL,
  family = NULL
)
```

Arguments

`model_pars`	A character vector specifying the name of the parameters to fit.
`initial_pars`	A numeric vector specifying the initial parameter values to #' evaluate the model at (required by `optim`). Defaults to 0 for each parameter.
`ll, ul`	A numeric vector specifying the lower and upper limits of the parameters to fit, respectively

`optimizer`	A string specifying the optimizer to use. One from `c("optim", "ga")`
`family`	A string specifying the family function to generate responses (and calculate the likelihood function with). One from `c("identity", "normal", "poisson")`.

Value

A list with optimizer options.

Note

Whenever a family function other than the identity is used, the family-specific parameters will always be appended to the end of the relevant lists.

See Also

`fit_model()`

`get_parameters` *Get model parameters*

Description

Get model parameters

Usage

```
get_parameters(design, model = NULL)
```

Arguments

`design`	An `data.frame` containing the experimental design.
`model`	A string specifying a model. One in `supported_models()`.

Value

A list with model parameters depending on model

Examples

```
block <- get_design("blocking")
get_parameters(block, model = "SM2007")
```

make_experiment *Make CalmrExperiment*

Description

Makes a CalmrExperiment object containing the arguments necessary to run an experiment.

Usage

```
make_experiment(
  design,
  parameters = NULL,
  model = NULL,
  iterations = 1,
  miniblocks = TRUE,
  .callback_fn = NULL,
  ...
)
```

Arguments

design	A design data.frame.
parameters	Parameters for a model as returned by get_parameters().
model	A string specifying the model name. One of supported_models().
iterations	An integer specifying the number of iterations per group.
miniblocks	Whether to organize trials in miniblocks.
.callback_fn	A function for keeping track of progress. Internal use.
...	Extra parameters passed to other functions.

Value

A CalmrExperiment object.

Note

The miniblocks option will direct the sampling function to create equally-sized miniblocks with random trials within a phase. For example, the phase string "2A/2B" will create two miniblocks with one of each trial. The phase string "2A/4B" will create two miniblocks with one A trial, and 2 B trials. However, the phase string "2A/1B" will not result in miniblocks, even if miniblocks here is set to TRUE.

See Also

parse_design(),

Examples

```
des <- data.frame(Group = "G1", P1 = "10A>(US)", R1 = TRUE)
ps <- get_parameters(des, model = "HD2022")
make_experiment(
  design = des, parameters = ps,
  model = "HD2022", iterations = 2
)
```

model_information *Model information functions*

Description

An assortment of functions to return model information.

Usage

```
supported_models()

supported_optimizers()

supported_families()

supported_plots(model = NULL)

get_model(model)

parameter_info(model = NULL)

model_outputs(model = NULL)
```

Arguments

`model`	A string specifying a model. One from `supported_models()`.

Value

`supported_models()` returns a character vector.

`supported_optimizers()` returns a character vector.

`supported_families()` returns a character vector.

`supported_plots()` returns a character vector or list (if model is NULL).

`get_model()` returns a model function.

`parameter_info()` returns a list or a list of lists (if model is NULL).

`model_outputs()` returns a character vector or list (if model is NULL).

Examples

```
# Outputs and plots supported by the RW1972 model
model_outputs("RW1972")

# Getting the model function implementing the PKH1982 model
pkh_func <- get_model("PKH1982")
head(pkh_func, 10)

# Getting the parameters required by SM2007
parameter_info("SM2007")
```

---

parse_design *Parse design data.frame*

Description

Parse design data.frame

Usage

```
parse_design(df, model = NULL, ...)
```

Arguments

`df`	A `data.frame` of dimensions (groups) by (2*phases+1).
`model`	(Optional) model to augment the design.
`...`	Other arguments passed to augment function.

Value

A CalmrDesign object.

Note

Each entry in even-numbered columns of df is a string formatted as per `phase_parser()`.

See Also

`phase_parser()`

Examples

```
df <- data.frame(
  Group = c("Group 1", "Group 2"),
  P1 = c("10AB(US)", "10A(US)"), R1 = c(TRUE, TRUE)
)
parse_design(df)
```

patch_graphs *Patch Calmr graphs*

Description

Convenience function to patch graphs with 'patchwork'

Usage

```
patch_graphs(graphs, selection = names(graphs))
```

Arguments

`graphs`	A list of named graphs, as returned by `graph()` or `calmr_model_graph()`
`selection`	A character or numeric vector determining the plots to patch.

Value

A 'patchwork' object

patch_plots *Patch Calmr plots*

Description

Convenience function to patch plots with `patchwork`

Usage

```
patch_plots(plots, selection = names(plots), plot_options = get_plot_opts())
```

Arguments

`plots`	A list of named plots, as returned by `calmr::plot`
`selection`	A character or numeric vector determining the plots to patch
`plot_options`	A list of plot options as returned by `get_plot_opts`

Value

A `patchwork` object

`pati` *Rat responses from Patittucci et al. 2016*

Description

A dataset containing rat nose pokes and lever presses when levers were associated with different appetitive stimuli.

Usage

```
pati
```

Format

A data.frame with the following variables:

subject subject identifier

block the 2-session block of training (1 to 8)

lever lever presented on the trial: L = left; R = right

us the stimulus that followed the lever: P = pellet; S = sucrose

response the response: lp = lever press; np = nose poke

rpert responses per trial ...

Source

Patittucci et al. (2016). JEP:ALC

`phase_parser` *Parses a phase string*

Description

Parses a phase string

Usage

```
phase_parser(phase_string)
```

Arguments

`phase_string` A string specifying trials within a phase.

Value

A named list with:

trial_info: A trial-named list of lists.

general_info: General phase information.

Note

This function is meant for internal use only, but we expose it so you can test your strings.

See Also

`parse_design()`

Examples

```
# A silly (but valid) string
phase_parser("10#Rescorla>Wagner")

# An invalid string that needs trial repetitions for one of trials.
try(phase_parser("10#Rescorla/Wagner"))
```

`rsa` *Perform representational similarity analysis*

Description

Perform representational similarity analysis

Usage

```
rsa(x, comparisons, test = FALSE, ...)
```

Arguments

`x`	A list of CalmrExperiment objects
`comparisons`	A model-named list containing the model outputs to compare.
`test`	Whether to test the RSA via permutation test. Default = FALSE.
`...`	Additional parameters passed to `stats::dist()` and `stats::cor()`

Value

A CalmrRSA object

Note

The object returned by this function can be later tested via its own `test()` method.

Examples

```
# Comparing the associations in three models
exp <- data.frame(
  Group = c("A", "B"),
  P1 = c("2(A)>(US)/1B>(US)", "1(A)>(US)/2B>(US)"),
  R1 = TRUE
)
models <- c("HD2022", "RW1972", "PKH1982")
parameters <- sapply(models, get_parameters, design = exp)
exp_res <- compare_models(exp,
  models = models
)
comparisons <- list(
  "HD2022" = c("vs"),
  "RW1972" = c("vs"),
  "PKH1982" = c("eivs")
)
res <- rsa(exp_res, comparisons = comparisons)
test(res, n_samples = 20)
```

`run_experiment` *Run experiment*

Description

Runs an experiment with minimal parameters.

Usage

```
run_experiment(x, outputs = NULL, parse = TRUE, aggregate = TRUE, ...)
```

Arguments

`x`	A CalmrExperiment or design `data.frame`
`outputs`	A character vector specifying which outputs to parse and aggregate. Defaults to NULL, in which case all model outputs are parsed/aggregated.
`parse`	A logical specifying whether the raw results should be parsed. Default = TRUE.
`aggregate`	A logical specifying whether the parsed results should be aggregated. Default = TRUE.
`...`	Arguments passed to other functions

Value

A CalmrExperiment with results.

Examples

```
# Using a data.frame only (throws warning)
df <- get_design("relative_validity")
run_experiment(df, model = "RW1972")

# Using custom parameters
df <- get_design("relative_validity")
pars <- get_parameters(df, model = "HD2022")
pars$alphas["US"] <- 0.6
run_experiment(df, parameters = pars, model = "HD2022")

# Using make_experiment, for more iterations
df <- get_design("blocking")
pars <- get_parameters(df, model = "SM2007")
exper <- make_experiment(df,
  parameters = pars, model = "SM2007",
  iterations = 4
)
run_experiment(exper)

# Only parsing the associations in the model, without aggregation
run_experiment(exper, outputs = "vs", aggregate = FALSE)
```

`set_reward_parameters` *Set reward parameters for ANCCR model*

Description

Set reward parameters for ANCCR model

Usage

```
set_reward_parameters(parameters, rewards = c("US"))
```

Arguments

`parameters` A list of parameters, as returned by `get_parameters()`

`rewards` A character vector specifying the reward stimuli. Default = `c("US")`

Value

A list of parameters

Note

The default behaviour of `get_parameters` for the ANCCR model is to set every reward-related parameter to its non-zero default value. This function will set those parameters to zero for non-reward stimuli

Index