Package: zoid (via r-universe)

October 29, 2026

Title Bayesian Zero-and-One Inflated Dirichlet Regression Modelling

Version 1.3.1

Description Fits Dirichlet regression and zero-and-one inflated Dirichlet regression with Bayesian methods implemented in Stan. These models are sometimes referred to as trinomial mixture models; covariates and overdispersion can optionally be included.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Biarch true

URL https://noaa-nwfsc.github.io/zoid/

BugReports https://github.com/noaa-nwfsc/zoid/issues

Depends R (>= 3.4.0)

Imports gtools, methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), rstantools (>= 2.1.1)

Suggests testthat, knitr, rmarkdown

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

SystemRequirements GNU make

VignetteBuilder knitr

NeedsCompilation yes

Author Eric J. Ward [aut, cre] (<https://orcid.org/0000-0002-4359-0296>), Alexander J. Jensen [aut] (<https://orcid.org/0000-0002-2911-8884>), Ryan P. Kelly [aut] (<https://orcid.org/0000-0001-5037-2441>), Andrew O. Shelton [aut] (<https://orcid.org/0000-0002-8045-6141>), William H. Satterthwaite [aut]

(<https://orcid.org/0000-0002-0436-7390>), Eric C. Anderson [aut] (<https://orcid.org/0000-0003-1326-0840>)

Maintainer Eric J. Ward <eric.ward@noaa.gov>

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2024-01-25 02:50:19 UTC

RemoteUrl https://github.com/cran/zoid

RemoteRef HEAD

RemoteSha 72f151b47f62bf2a34a2fd48ca923cf777a480c6

Contents

zoid-package . . . 2
broken_stick . . . 3
chinook . . . 4
coddiet . . . 4
fit_dirichlet . . . 5
fit_prior . . . 5
fit_zoid . . . 6
get_fitted . . . 7
get_pars . . . 8
parse_re_formula . . . 9
rmspe_calc . . . 9

Index **10**

`zoid-package` *The 'zoid' package.*

Description

A DESCRIPTION OF THE PACKAGE

References

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. https://mc-stan.org

broken_stick *Random generation of datasets using the dirichlet broken stick method*

Description

Random generation of datasets using the dirichlet broken stick method

Usage

```
broken_stick(
  n_obs = 1000,
  n_groups = 10,
  ess_fraction = 1,
  tot_n = 100,
  p = NULL
)
```

Arguments

n_obs	Number of observations (rows of data matrix to simulate). Defaults to 10
n_groups	Number of categories for each observation (columns of data matrix). Defaults to 10
ess_fraction	The effective sample size fraction, defaults to 1
tot_n	The total sample size to simulate for each observation. This is approximate and the actual simulated sample size will be slightly smaller. Defaults to 100
p	The stock proportions to simulate from, as a vector. Optional, and when not included, random draws from the dirichlet are used

Value

A 2-element list, whose 1st element X_obs is the simulated dataset, and whose 2nd element is the underlying vector of proportions p used to generate the data

Examples

```
y <- broken_stick(n_obs = 3, n_groups = 5, tot_n = 100)

# add custom proportions
y <- broken_stick(
  n_obs = 3, n_groups = 5, tot_n = 100,
  p = c(0.1, 0.2, 0.3, 0.2, 0.2)
)
```

---

chinook — *Data from Satterthwaite, W.H., Ciancio, J., Crandall, E., Palmer-Zwahlen, M.L., Grover, A.M., O'Farrell, M.R., Anson, E.C., Mohr, M.S. & Garza, J.C. (2015). Stock composition and ocean spatial distribution from California recreational chinook salmon fisheries using genetic stock identification. Fisheries Research, 170, 166–178. The data genetic data collected from port-based sampling of recreationally-landed Chinook salmon in California from 1998-2002.*

Description

Data from Satterthwaite, W.H., Ciancio, J., Crandall, E., Palmer-Zwahlen, M.L., Grover, A.M., O'Farrell, M.R., Anson, E.C., Mohr, M.S. & Garza, J.C. (2015). Stock composition and ocean spatial distribution from California recreational chinook salmon fisheries using genetic stock identification. Fisheries Research, 170, 166–178. The data genetic data collected from port-based sampling of recreationally-landed Chinook salmon in California from 1998-2002.

Usage

```
chinook
```

Format

A data frame.

---

coddiet — *Data from Magnussen, E. 2011. Food and feeding habits of cod (Gadus morhua) on the Faroe Bank. – ICES Journal of Marine Science, 68: 1909–1917. The data here are Table 3 from the paper, with sample proportions (columns w) multiplied by total weight to yield total grams (g) for each sample-diet item combination. Dashes have been replaced with 0s.*

Description

Data from Magnussen, E. 2011. Food and feeding habits of cod (Gadus morhua) on the Faroe Bank. – ICES Journal of Marine Science, 68: 1909–1917. The data here are Table 3 from the paper, with sample proportions (columns w) multiplied by total weight to yield total grams (g) for each sample-diet item combination. Dashes have been replaced with 0s.

Usage

```
coddiet
```

Format

A data frame.

fit_dirichlet *Extract point estimates of compositions from fitted model.*

Description

Extract point estimates of compositions from fitted model.

Usage

```
fit_dirichlet(data)
```

Arguments

data	The data to fit the dirichlet distribution to

fit_prior *Find appropriate standard deviations for prior*

Description

Find appropriate standard deviations for prior

Usage

```
fit_prior(n_bins, n_draws = 10000, target = 1/n_bins, iterations = 5)
```

Arguments

n_bins	Bins for the Dirichlet distribution
n_draws	Numbers of samples to use for doing calculation
target	The goal of the specified prior, e.g. 1 or 1/n_bins
iterations	to try, to ensure robust solution. Defaults to 5

Value

A 3-element list consisting of sd (the approximate standard deviation in transformed space that gives a similar prior to that specified), value (the value of the root mean squared percent error function being minimized), and convergence (0 if convergence occurred, error code from optim() otherwise)

Examples

```
# fit model with 3 components / alpha = 1
set.seed(123)
f <- fit_prior(n_bins = 3, n_draws = 1000, target = 1)
# fit model with 20 components / alpha = 1/20
f <- fit_prior(n_bins = 20, n_draws = 1000, target = 1 / 20)
```

`fit_zoid` *Fit a trinomial mixture model with Stan*

Description

Fit a trinomial mixture model that optionally includes covariates to estimate effects of factor or continuous variables on proportions.

Usage

```
fit_zoid(
  formula = NULL,
  design_matrix,
  data_matrix,
  chains = 3,
  iter = 2000,
  warmup = floor(iter/2),
  overdispersion = FALSE,
  overdispersion_sd = 5,
  posterior_predict = FALSE,
  moment_match = FALSE,
  prior_sd = NA,
  ...
)
```

Arguments

`formula`	The model formula for the design matrix. Does not need to have a response specified. If =NULL, then the design matrix is ignored and all rows are treated as replicates
`design_matrix`	A data frame, dimensioned as number of observations, and covariates in columns
`data_matrix`	A matrix, with observations on rows and number of groups across columns
`chains`	Number of mcmc chains, defaults to 3
`iter`	Number of mcmc iterations, defaults to 2000
`warmup`	Number iterations for mcmc warmup, defaults to 1/2 of the iterations
`overdispersion`	Whether or not to include overdispersion parameter, defaults to FALSE

overdispersion_sd
: Prior standard deviation on 1/overdispersion parameter, Defaults to inv-Cauchy(0,5)

posterior_predict
: Whether or not to return draws from posterior predictive distribution (requires more memory)

moment_match
: Whether to do moment matching via `loo::loo_moment_match()`. This increases memory by adding all temporary parmaeters to be saved and returned

prior_sd
: Parameter to be passed in to use as standard deviation of the normal distribution in transformed space. If covariates are included this defaults to 1, but for models with single replicate, defaults to 1/n_bins.

...
: Any other arguments to pass to `rstan::sampling()`.

Examples

```
y <- matrix(c(3.77, 6.63, 2.60, 0.9, 1.44, 0.66, 2.10, 3.57, 1.33),
  nrow = 3, byrow = TRUE
)
# fit a model with no covariates
fit <- fit_zoid(data_matrix = y, chains = 1, iter = 100)

# fit a model with 1 factor
design <- data.frame("fac" = c("spring", "spring", "fall"))
fit <- fit_zoid(formula = ~fac, design_matrix = design, data_matrix = y, chains = 1, iter = 100)

# try a model with random effects
set.seed(123)
y <- matrix(runif(99,1,4), ncol=3)
design <- data.frame("fac" = sample(letters[1:5], size=nrow(y), replace=TRUE))
design$fac <- as.factor(design$fac)
fit <- fit_zoid(formula = ~(1|fac), design_matrix = design, data_matrix = y, chains = 1, iter = 100)
```

---

get_fitted *Extract estimates of predicted latent proportions.*

Description

Extract point estimates of compositions from fitted model.

Usage

```
get_fitted(fitted_model, conf_int = 0.05)
```

Arguments

fitted_model
: The fitted model returned as an rstan object from the call to zoid

conf_int
: Parameter controlling confidence intervals calculated, defaults to 0.05 for 95% intervals

Value

A list containing the posterior summaries of estimated parameters, with element `mu` (the predicted values in normal space). For predictions in transformed space, or overdispersion, see `get_pars()`

Examples

```
y <- matrix(c(3.77, 6.63, 2.60, 0.9, 1.44, 0.66, 2.10, 3.57, 1.33),
  nrow = 3, byrow = TRUE
)
# fit a model with no covariates
fit <- fit_zoid(data_matrix = y)
p_hat <- get_fitted(fit)
```

`get_pars` *Extract parameters from fitted model.*

Description

Extract estimated parameters from fitted model.

Usage

```
get_pars(fitted_model, conf_int = 0.05)
```

Arguments

`fitted_model`	The fitted model returned as an rstan object from the call to zoid
`conf_int`	Parameter controlling confidence intervals calculated, defaults to 0.05 for 95% intervals

Value

A list containing the posterior summaries of estimated parameters. At minimum, this will include `p` (the estimated proportions) and `betas` (the predicted values in transformed space). For models with overdispersion, an extra element `phi` will also be returned, summarizing overdispersion. For models with random intercepts, estimates of the group level effects will also be returned as `zetas` (again, in transformed space). For predictions in normal space, see `get_fitted()`

Examples

```
y <- matrix(c(3.77, 6.63, 2.60, 0.9, 1.44, 0.66, 2.10, 3.57, 1.33),
  nrow = 3, byrow = TRUE
)
# fit a model with no covariates
fit <- fit_zoid(data_matrix = y)
p_hat <- get_pars(fit)
```

parse_re_formula *Fit a trinomial mixture model that optionally includes covariates to estimate effects of factor or continuous variables on proportions.*

Description

Fit a trinomial mixture model that optionally includes covariates to estimate effects of factor or continuous variables on proportions.

Usage

```
parse_re_formula(formula, data)
```

Arguments

formula	The model formula for the design matrix.
data	The data matrix used to construct RE design matrix

rmspe_calc *Find appropriate prior for a given target distribution.*

Description

Extract point estimates of compositions from fitted model.

Usage

```
rmspe_calc(par, n_bins, n_draws, target)
```

Arguments

par	The parameter (standard deviation) to be searched over to find a Dirichlet equivalent
n_bins	Bins for the Dirichlet distribution
n_draws	Numbers of samples to use for doing calculation
target	The goal of the specified prior, e.g. 1 or 1/n_bins

Index

∗ **datasets**
chinook, 4
coddiet, 4

broken_stick, 3

chinook, 4
coddiet, 4

fit_dirichlet, 5
fit_prior, 5
fit_zoid, 6

get_fitted, 7
get_fitted(), *8*
get_pars, 8
get_pars(), *8*

loo::loo_moment_match(), *7*

optim(), *5*

parse_re_formula, 9

rmspe_calc, 9
rstan::sampling(), *7*

trinomix *(*zoid-package*)*, 2

zoid-package, 2