Package: miniLNM (via r-universe)

October 29, 2026

Title Miniature Logistic-Normal Multinomial Models

Version 0.1.2

Description Logistic-normal Multinomial (LNM) models are common in problems with multivariate count data. This package gives a simple implementation with a 30 line 'Stan' script. This lightweight implementation makes it an easy starting point for other projects, in particular for downstream tasks that require analysis of ``compositional'' data. It can be applied whenever a multinomial probability parameter is thought to depend linearly on inputs in a transformed, log ratio space. Additional utilities make it easy to inspect, create predictions, and draw samples using the fitted models. More about the LNM can be found in Xia et al. (2013) ``A Logistic Normal Multinomial Regression Model for Microbiome Compositional Data Analysis'' <doi:10.1111/biom.12079> and Sankaran and Holmes (2023) ``Generative Models: An Interdisciplinary Perspective'' <doi:10.1146/annurev-statistics-033121-110134>.

License CC0

Encoding UTF-8

Depends R (>= 4.1.0)

Imports Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), cli, dplyr, fansi, formula.tools, glue, methods, posterior, rstan (>= 2.18.1), rstantools (>= 2.4.0), tidyselect

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

BugReports https://github.com/krisrs1128/miniLNM/issues/

URL https://github.com/krisrs1128/miniLNM/,
https://krisrs1128.github.io/miniLNM/

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Kris Sankaran [aut, cre] (ORCID:
<https://orcid.org/0000-0002-9415-1971>)

Maintainer Kris Sankaran <ksankaran@wisc.edu>

Config/pak/sysreqs make

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

Date/Publication 2026-07-21 02:40:02 UTC

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

RemoteRef HEAD

RemoteSha 5cbbb888be77bcddca42f033c5c897c878aec083

Contents

ansi_aware_handler 2
beta_mean 3
beta_samples 4
lnm 5
lnm-class 6
lnm_data 6
phi_inverse 7
predict,lnm-method 8
prepare_newdata 8
sample,lnm-method 9

Index **11**

`ansi_aware_handler` *Pretty Printing*

Description

Helper function for printing ANSI in Rmarkdown output. Use this at the start of your Rmarkdown files to include colors in the printed object names in the final compiled output.

Usage

```
ansi_aware_handler(x, options)
```

Arguments

x	A character vector potentially including ANSI.
options	Unused placeholder argument.

Details

Taken from the post at

https://blog.djnavarro.net/posts/2021-04-18_pretty-little-clis/

Value

A string with HTML reformatted to ensure colors appear in printed code blocks in rmarkdown output.

Examples

```
knitr::knit_hooks$set(output = ansi_aware_handler)
options(crayon.enabled = TRUE)
```

---

beta_mean *LNM Posterior Mean*

Description

Average the samples for the beta parameter from the VB posterior mean. This is used to get predicted compositions when using `predict` on an lnm model.

Usage

```
beta_mean(fit)
```

Arguments

fit	An object of class `lnm` whose estimate slot contains the 'rstan' fitted logistic normal multinomial model.

Value

A matrix whose rows are predictors and columns are outcomes in the beta parameter for the LNM model.

Examples

```
example_data <- lnm_data(N = 200, K = 10)
xy <- dplyr::bind_cols(example_data[c("X", "y")])
fit <- lnm(
    starts_with("y") ~ starts_with("x"), xy,
    iter = 1000, output_samples = 200
)
beta_mean(fit)
```

`beta_samples` *LNM Posterior Samples*

Description

Return multiple samples for the beta parameter from the VB posterior mean. This is used to simulate new compositions when using `sample` on an lnm model.

Usage

```
beta_samples(fit, size = 1)
```

Arguments

`fit`	An object of class `lnm` whose estimate slot contains the 'rstan' fitted logistic normal multinomial model.
`size`	The number of draws from the posterior to return.

Value

A matrix whose rows are predictors and columns are outcomes in the beta parameter for the LNM model.

Examples

```
example_data <- lnm_data(N = 200, K = 10)
xy <- dplyr::bind_cols(example_data[c("X", "y")])
fit <- lnm(
    starts_with("y") ~ starts_with("x"), xy,
    iter = 1000, output_samples = 200
)
beta_samples(fit, size = 2)
```

`lnm` *Fit a logistic normal multinomial model using R's formula interface.*

Description

This function fits a logistic normal multinomial (LNM) model to the data using R's formula interface. The LNM model is a generalization of the multinomial logistic regression model, allowing for correlated responses within each category of the response variable. It can be used to learn the relationship between experimental/environmental factors and community composition. It is a statistical model that estimates the probabilities of different outcomes in a multinomial distribution, given a set of covariates. The LNM model assumes that a log-ratio of the outcome probabilities follow a multivariate normal distribution. By fitting the LNM model to observed data, we can infer the effects of the covariates on the outcome compositions.

Usage

```
lnm(formula, data, sigma_b = 2, l1 = 10, l2 = 10, ...)
```

Arguments

`formula`	A formula specifying the model structure.
`data`	A data frame containing the variables specified in the formula.
`sigma_b`	The prior standard deviation of the beta coefficients in the LNM model. See the 'Stan' code definition in inst/stan/lnm.stan for the full model specification.
`l1`	The first inverse gamma hyperprior parameter for sigmas_mu.
`l2`	The first inverse gamma hyperprior parameter for sigmas_mu.
`...`	Additional arguments to be passed to the underlying vb() call from 'rstan'.

Value

An object of class "lnm" representing the fitted LNM model.

Examples

```
example_data <- lnm_data(N = 200, K = 10)
xy <- dplyr::bind_cols(example_data[c("X", "y")])
fit <- lnm(
    starts_with("y") ~ starts_with("x"), xy,
    iter = 1000, output_samples = 200
)
```

lnm-class *S4 Class for a Logistic Normal Multinomial Model*

Description

The Logistic Normal Multinomial (LNM) model is used to learn the relationship between experimental/environmental factors and community composition. It is a statistical model that estimates the probabilities of different outcomes in a multinomial distribution, given a set of covariates. The LNM model assumes that a log-ratio of the outcome probabilities follow a multivariate normal distribution. By fitting the LNM model to observed data, we can infer the effects of the covariates on the outcome compositions.

Details

This class combines all information into three slots:

Slots

`estimate` The fitted logistic normal multinomial model, with parameter B relating covariates to outcome compositions.

`template` The data used to estimate the parameters in the estimate slot.

`formula` The R formula representation of the relationship between output compositions and input variables.

lnm_data *Simulates data from a Logistic Normal Multinomial Model.*

Description

Simulates data from a Logistic Normal Multinomial Model.

Usage

```
lnm_data(N = 100, D = 5, K = 10)
```

Arguments

`N`	The number of samples in the output data.
`D`	The number of covariates, each of which can influence the response composition vector (e.g., the timepoint or disease status).
`K`	The number of output dimensions (e.g., number of taxa).

Value

A list with the following components:

X An N x D matrix of covariates.

y The N x K simulated samples.

B The D x K relationship between covariates and outputs.

Examples

```
lnm_data(5, 3, 3)
```

`phi_inverse` *Inverse log ratio transformation*

Description

This function applies the inverse logistic function to a vector, which maps the values of the vector to the range (0, 1).

Usage

```
phi_inverse(mu)
```

Arguments

`mu`	A numeric vector to transform using an inverse log ratio transformation.

Value

A numeric vector with values mapped to the range (0, 1) and a reference coordinate added.

Examples

```
phi_inverse(c(-5, 0, 5))
```

predict,lnm-method *LNM Fitted Probabilities*

Description

Given an input dataset, predict the output composition. Specifically, this outputs $\phi^{-1}(Bx)$, for the inverse log ratio transformation ϕ^{-1} and fitted covariate matrix B.

Usage

```
## S4 method for signature 'lnm'
predict(object, newdata = NULL, ...)
```

Arguments

object	An object of class lnm with fitted parameters $\hat{B}$ and which we want to use to form predictions on new samples.
newdata	New samples on which to form predictions. Defaults to NULL, in which case predictions are made at the same design points as those used during the original training.
...	Additional keyword arguments, for consistency with R's predict generic (never used).

Value

A matrix with predictions along rows and outcomes along columns. Rows sum up to one.

Examples

```
example_data <- lnm_data(N = 200, K = 10)
xy <- dplyr::bind_cols(example_data[c("X", "y")])
fit <- lnm(
    starts_with("y") ~ starts_with("x"), xy,
    iter = 1000, output_samples = 200
)
head(predict(fit))
```

prepare_newdata *Design Matrix for a Model*

Description

This is a helper function to form the design matrix for an LNM regression starting from a fitted model's formula object. It is an analog of model.matrix for the multiresponse setting.

Usage

```
prepare_newdata(fit, newdata = NULL)
```

Arguments

fit	An object of class lnm whose estimate slot contains the 'rstan' fitted logistic normal multinomial model.
newdata	A data.frame containing variables in the formula definition of the fit, but which hasn't been converted into the matrix format needed for internal prediction.

Value

A matrix containing the design matrix that can be multiplied with the fitted Beta parameter to get fitted compositions.

Examples

```
example_data <- lnm_data(N = 200, K = 5)
xy <- dplyr::bind_cols(example_data[c("X", "y")])
fit <- lnm(
    starts_with("y") ~ starts_with("x"), xy,
    iter = 1000, output_samples = 200
)
prepare_newdata(fit, example_data[["X"]])
```

---

sample,lnm-method *LNM Fitted Probabilities*

Description

Given an input dataset, sample compositions that are consistent with the input. Specifically, this samples from a multinomial with mean $\phi^{-1}(Bx)$. The default depth is 5e4. Modify the "depth" parameter to change this.

Usage

```
## S4 method for signature 'lnm'
sample(x, size = 1, depth = 50000, newdata = NULL, ...)
```

Arguments

x	An object of class lnm with fitted parameters $\hat{B}$ and which we want to use to form predictions on new samples.
size	The number of samples to generate.
depth	The depth to use when sampling the multinomial for each simulated element.

newdata	New samples on which to form predictions. Defaults to NULL, in which case predictions are made at the same design points as those used during the original training.
...	Additional keyword arguments, for consistency with R's predict generic (never used).

Value

A matrix of dimension size x n_outcomes, where each row represents one sample from the posterior predictive of the fitted logistic-normal multinomial model. Each row sums up to the depth argument, which defaults to 5e4.

Examples

```
example_data <- lnm_data(N = 200, K = 10)
xy <- dplyr::bind_cols(example_data[c("X", "y")])
fit <- lnm(
    starts_with("y") ~ starts_with("x"), xy,
    iter = 1000, output_samples = 200
)
head(sample(fit))
```

Index

ansi_aware_handler, 2

beta_mean, 3
beta_samples, 4

lnm, 5
lnm-class, 6
lnm_data, 6

phi_inverse, 7
predict,lnm-method, 8
prepare_newdata, 8

sample,lnm-method, 9