Package 'glmmsel'

June 26, 2025

Type Package

Title Generalised Linear Mixed Model Selection

Version 1.0.3

Description Provides tools for fitting sparse generalised linear mixed models with l0 regularisation. Selects fixed and random effects under the hierarchy constraint that fixed effects must precede random effects. Uses coordinate descent and local search algorithms to rapidly deliver near-optimal estimates. Gaussian and binomial response families are currently supported. For more details see Thompson, Wand, and Wang (2025) <doi:10.48550/arXiv.2506.20425>.

URL https://github.com/ryan-thompson/glmmsel

BugReports https://github.com/ryan-thompson/glmmsel/issues

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports ggplot2, Rcpp

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, knitr, rmarkdown, lme4, MASS, nlme

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Ryan Thompson [aut, cre] (ORCID: <https://orcid.org/0000-0002-9002-0448>)

Maintainer Ryan Thompson <ryan.thompson-1@uts.edu.au>

Repository CRAN

Date/Publication 2025-06-26 08:00:02 UTC

Contents

`coef.cv.glmmsel` *Coefficient function for* `cv.glmmsel` *object*

Description

Extract cluster coefficients for a cross-validated value of the regularisation parameter.

Usage

```
## S3 method for class 'cv.glmmsel'
coef(object, lambda = "lambda.min", ...)
```

Arguments

`object`	an object of class `cv.glmmsel`
`lambda`	a value of the regularisation parameter
`...`	any other arguments

Value

An array of coefficients.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

coef.glmmsel *Coefficient function for* glmmsel *object*

Description

Extracts coefficients for a specified value of the regularisation parameter.

Usage

```
## S3 method for class 'glmmsel'
coef(object, lambda = NULL, ...)
```

Arguments

object	an object of class glmmsel
lambda	a value of the regularisation parameter
...	any other arguments

Value

An array of coefficients.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

cv.glmmsel *Cross-validated generalised linear mixed model selection*

Description

Fits the regularisation path for a sparse generalised linear mixed model and then cross-validates this path.

Usage

```
cv.glmmsel(
  x,
  y,
  cluster,
  family = c("gaussian", "binomial"),
  lambda = NULL,
  nfold = 10,
  folds = NULL,
  cv.loss = NULL,
  interpolate = TRUE,
  ...
)
```

Arguments

x	a predictor matrix
y	a response vector
cluster	a vector of length nrow(x) with the jth element identifying the cluster that the jth observation belongs to
family	the likelihood family to use; 'gaussian' for a continuous response or 'binomial' for a binary response
lambda	the regularisation parameter for the overlapping penalty on the fixed and random slopes
nfold	the number of cross-validation folds
folds	an optional vector of length nrow(x) with the jth entry identifying the fold that the jth observation belongs to
cv.loss	an optional cross-validation loss-function to use; should accept a vector of predicted values and a vector of actual values
interpolate	a logical indicating whether to interpolate the lambda sequence for the cross-validation fits
...	any other arguments for glmmsel()

Value

An object of class cv.glmmsel; a list with the following components:

cv.mean	a vector of cross-validation means
cv.sd	a vector of cross-validation standard errors
lambda	a vector of cross-validated regularisation parameters
lambda.min	the value of lambda minimising cv.mean
fit	the fit from running glmmsel() on the full data

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

References

Thompson, R., Wand, M. P., and Wang, J. J. J. (2025). 'Scalable subset selection in linear mixed models'. arXiv: 2506.20425.

Examples

```
# Generate data
set.seed(1234)
n <- 100
m <- 4
p <- 10
s <- 5
x <- matrix(rnorm(n * p), n, p)
```

```
beta <- c(rep(1, s), rep(0, p - s))
u <- cbind(matrix(rnorm(m * s), m, s), matrix(0, m, p - s))
cluster <- sample(1:m, n, replace = TRUE)
xb <- rowSums(x * sweep(u, 2, beta, '+')[cluster, ])
y <- rnorm(n, xb)

# Fit sparse linear mixed model
fit <- cv.glmmsel(x, y, cluster)
plot(fit)
fixef(fit)
ranef(fit)
coef(fit)
predict(fit, x[1:3, ], cluster[1:3])
```

`fixef` *Fixed effects function*

Description

Generic function for extracting fixed effects from model objects.

Usage

```
fixef(object, ...)
```

Arguments

`object`	a model object
`...`	any other arguments

Value

Depends on the specific method implementation.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

fixef.cv.glmmsel *Fixed effects function for* cv.glmmsel *object*

Description

Extract fixed effects for a cross-validated value of the regularisation parameter.

Usage

```
## S3 method for class 'cv.glmmsel'
fixef(object, lambda = "lambda.min", ...)
```

Arguments

object	an object of class cv.glmmsel
lambda	a value of the regularisation parameter
...	any other arguments

Value

A matrix of fixed effects.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

fixef.glmmsel *Fixed effects function for* glmmsel *object*

Description

Extracts fixed effects for a specified value of the regularisation parameter.

Usage

```
## S3 method for class 'glmmsel'
fixef(object, lambda = NULL, ...)
```

Arguments

object	an object of class glmmsel
lambda	a value of the regularisation parameter
...	any other arguments

Value

A matrix of fixed effects.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

`glmmsel` *Generalised linear mixed model selection*

Description

Fits the regularisation path for a sparse generalised linear mixed model (GLMM).

Usage

```
glmmsel(
  x,
  y,
  cluster,
  family = c("gaussian", "binomial"),
  local.search = FALSE,
  max.nnz = 100,
  nlambda = 100,
  lambda.step = 0.99,
  lambda = NULL,
  alpha = 0.8,
  intercept = TRUE,
  random.intercept = TRUE,
  standardise = TRUE,
  eps = 1e-04,
  max.cd.iter = 10000,
  max.ls.iter = 100,
  max.bls.iter = 30,
  t.init = 1,
  t.scale = 0.5,
  max.pql.iter = 100,
  active.set = TRUE,
  active.set.count = 3,
  sort = TRUE,
  screen = 100,
  warn = TRUE
)
```

Arguments

x	a predictor matrix
y	a response vector
cluster	a vector of length nrow(x) with the jth element identifying the cluster that the jth observation belongs to
family	the likelihood family to use; 'gaussian' for a continuous response or 'binomial' for a binary response
local.search	a logical indicating whether to perform local search after coordinate descent; typically leads to higher quality solutions
max.nnz	the maximum number of predictors ever allowed to be active
nlambda	the number of regularisation parameters to evaluate when lambda is computed automatically
lambda.step	the step size taken when computing lambda from the data; should be a value strictly between 0 and 1; larger values typically lead to a finer grid of subset sizes
lambda	an optional vector of regularisation parameters
alpha	the hierarchical parameter
intercept	a logical indicating whether to include a fixed intercept
random.intercept	a logical indicating whether to include a random intercept; applies only when intercept = TRUE
standardise	a logical indicating whether to scale the data to have unit root mean square; all parameters are returned on the original scale of the data
eps	the convergence tolerance; convergence is declared when the relative maximum difference in consecutive parameter values is less than eps
max.cd.iter	the maximum number of coordinate descent iterations allowed
max.ls.iter	the maximum number of local search iterations allowed
max.bls.iter	the maximum number of backtracking line search iterations allowed
t.init	the initial value of the gradient step size during backtracking line search
t.scale	the scaling parameter of the gradient step size during backtracking line search
max.pql.iter	the maximum number of penalised quasi-likelihood iterations allowed
active.set	a logical indicating whether to use active set updates; typically lowers the run time
active.set.count	the number of consecutive coordinate descent iterations in which a subset should appear before running active set updates
sort	a logical indicating whether to sort the coordinates before running coordinate descent; typically leads to higher quality solutions
screen	the number of predictors to keep after gradient screening; smaller values typically lower the run time
warn	a logical indicating whether to print a warning if the algorithms fail to converge

Value

An object of class `glmmsel`; a list with the following components:

`beta0`	a vector of fixed intercepts
`gamma0`	a vector of random intercept variances
`beta`	a matrix of fixed slopes
`gamma`	a matrix of random slope variances
`u`	an array of random coefficient predictions
`sigma2`	a vector of residual variances
`loss`	a vector of loss function values
`cd.iter`	a vector indicating the number of coordinate descent iterations for convergence
`ls.iter`	a vector indicating the number of local search iterations for convergence
`pql.iter`	a vector indicating the number of penalised quasi-likelihood iterations for convergence
`nnz`	a vector of the number of nonzeros
`lambda`	a vector of regularisation parameters used for the fit
`family`	the likelihood family used
`clusters`	a vector of cluster identifiers
`alpha`	the value of the hierarchical parameter used for the fit
`intercept`	whether a fixed intercept is included in the model
`random.intercept`	whether a random intercept is included in the model

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

References

Thompson, R., Wand, M. P., and Wang, J. J. J. (2025). 'Scalable subset selection in linear mixed models'. arXiv: 2506.20425.

Examples

```
# Generate data
set.seed(1234)
n <- 100
m <- 4
p <- 10
s <- 5
x <- matrix(rnorm(n * p), n, p)
beta <- c(rep(1, s), rep(0, p - s))
u <- cbind(matrix(rnorm(m * s), m, s), matrix(0, m, p - s))
cluster <- sample(1:m, n, replace = TRUE)
xb <- rowSums(x * sweep(u, 2, beta, '+')[cluster, ])
```

```
y <- rnorm(n, xb)

# Fit sparse linear mixed model
fit <- glmmsel(x, y, cluster)
plot(fit)
fixef(fit, lambda = 10)
ranef(fit, lambda = 10)
coef(fit, lambda = 10)
predict(fit, x[1:3, ], cluster[1:3], lambda = 10)
```

`plot.cv.glmmsel` *Plot function for* `cv.glmmsel` *object*

Description

Plot the cross-validation loss as a function of the regularisation parameter.

Usage

```
## S3 method for class 'cv.glmmsel'
plot(x, ...)
```

Arguments

`x`	an object of class `cv.glmmsel`
`...`	any other arguments

Value

A plot of the cross-validation results.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

`plot.glmmsel` *Plot function for* `glmmsel` *object*

Description

Plots the coefficients as a function of the regularisation parameter.

Usage

```
## S3 method for class 'glmmsel'
plot(x, cluster = NULL, ...)
```

Arguments

x	an object of class glmmsel
cluster	the cluster whose coefficients to plot
...	any other arguments

Value

A plot of the coefficient profiles.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

---

predict.cv.glmmsel *Predict function for* cv.glmmsel *object*

Description

Generates predictions for new data using a cross-validated value of the regularisation parameter.

Usage

```
## S3 method for class 'cv.glmmsel'
predict(object, x.new, cluster.new, lambda = "lambda.min", ...)
```

Arguments

object	an object of class cv.glmmsel
x.new	a matrix of new values for the predictors
cluster.new	a vector identifying the clusters that the rows of x.new belong to
lambda	a value of the regularisation parameter
...	any other arguments

Value

A matrix of predictions.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

predict.glmmsel *Predict function for* glmmsel *object*

Description

Generates predictions for new data using a specified value of the regularisation parameter.

Usage

```
## S3 method for class 'glmmsel'
predict(object, x.new, cluster.new, lambda = NULL, ...)
```

Arguments

object	an object of class glmmsel
x.new	a matrix of new values for the predictors
cluster.new	a vector identifying the clusters that the rows of x.new belong to
lambda	a value of the regularisation parameter
...	any other arguments

Value

A matrix of predictions.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

ranef *Random effects function*

Description

Generic function for extracting random effects from model objects.

Usage

```
ranef(object, ...)
```

Arguments

object	a model object
...	any other arguments

Value

Depends on the specific method implementation.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

---

ranef.cv.glmmsel *Random effects function for* cv.glmmsel *object*

Description

Extract random effects for a cross-validated value of the regularisation parameter.

Usage

```
## S3 method for class 'cv.glmmsel'
ranef(object, lambda = "lambda.min", ...)
```

Arguments

object	an object of class cv.glmmsel
lambda	a value of the regularisation parameter
...	any other arguments

Value

A matrix of random effects.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

---

ranef.glmmsel *Random effects function for* glmmsel *object*

Description

Extracts random effects for a specified value of the regularisation parameter.

Usage

```
## S3 method for class 'glmmsel'
ranef(object, lambda = NULL, ...)
```

Arguments

`object`	an object of class `glmmsel`
`lambda`	a value of the regularisation parameter
`...`	any other arguments

Value

A matrix of random effects.

Author(s)

Ryan Thompson <ryan.thompson-1@uts.edu.au>

Index