Package 'gremlin'

October 30, 2018

Type Package

Title Mixed-Effects REML Incorporating Generalized Inverses

Version 0.1.0.0

URL http://github.com/matthewwolak/gremlin

BugReports http://github.com/matthewwolak/gremlin/issues

Depends Matrix

Imports methods

Suggests nadiv

LazyData yes

NeedsCompilation yes

Description Fit linear mixed-effects models using restricted (or residual)
maximum likelihood (REML) and with generalized inverse matrices to specify
covariance structures for random effects. In particular, the package is suited
to fit quantitative genetic mixed models, often referred to as 'animal models'
(Kruuk. 2004 <DOI: 10.1098/rstb.2003.1437>). Implements the average
information algorithm as the main tool to maximize the restricted likelihood,
but with other algorithms available (Meyer. 1997. Genet Sel Evol 29:97; Meyer
and Smith. 1998. Genet Sel Evol 28:23.).

License GPL-3

RoxygenNote 6.0.1

Author Matthew Wolak [cre, aut] (<https://orcid.org/0000-0002-7962-0071>)

Maintainer Matthew Wolak <matthewwolak@gmail.com>

Repository CRAN

Date/Publication 2018-10-30 18:30:02 UTC

R topics documented:

1

---

gremlin-package   *Mixed-Effects REML Incorporating Generalized Inverses*

Description

Fit linear mixed-effects models using restricted (or residual) maximum likelihood (REML) and
with generalized inverse matrices to specify covariance structures for random effects. In particular,
the package is suited to fit quantitative genetic mixed models, often referred to as 'animal models'
(Kruuk 2004 <DOI: 10.1098/rstb.2003.1437>). Implements the average information algorithm as
the main tool to maximize the restricted likelihood, but with other algorithms available (Meyer.
1997. Genet Sel Evol 29:97; Meyer & Smith. 1998. Genet Sel Evol 28:23.).

Details

The package also implements the average information algorithm to efficiently maximize the log-
likelihood (Thompson & Johnson 1995; Gilmour et al. 1995; Meyer & Smith 1996). The average
information algorithm combined with sparse matrix techniques can potentially make model fitting
very efficient.

Author(s)

Maintainer: Matthew Wolak <matthewwolak@gmail.com> (0000-0002-7962-0071)

References

Mrode, RA. 2005. Linear Models for the Prediction of Animal Breeding Values, 2nd ed. CABI Pub-
lishing, Cambridge. Meyer, K & Smith, SP. 1996. Restricted maximum likelihood estimation for
animal models using derivatives of the likelihood. Genetics Selection Evolution 28:23-49. Gilmour,
AR, Thompson, R, & Cullis, BR. 1995. Average information REML: An efficient algorithm for
variance parameter estimation in linear mixed models. Biometrics 51:1440-1450. Johnson, DL, &
Thompson, R. 1995. Restricted maximum likelihood estimation of variance components for uni-
variate animal models using sparse matrix techniques and average information. Journal of Dairy
Science 78:449-456.

See Also

MCMCglmm

Examples

```
   require(nadiv)
   Ainv <- makeAinv(Mrode3[-c(1:2), 1:3])$Ainv
   mod11 <- gremlinR(WWG11 ~ sex - 1,
random = ~ calf,
data = Mrode11,
ginverse = list(calf = Ainv),
Gstart = matrix(0.2), Rstart = matrix(0.4),
maxit = 10, v = 2)
```

---

anova.gremlin   *anova( ) for gremlin objects*

Description

REML Likelihood Ratio Tests for gremlin models using anova()

Usage

```
## S3 method for class 'gremlin'
anova(object, ..., model.names = NULL)
```

Arguments

object	An object of class 'gremlin'.
...	Additional objects of class 'gremlin'.
model.names	Optional character vector with model names to be used in the anova table

Value

A data.frame containing the nested comparison of model objects via a REML likelihood ratio test.

Author(s)

<matthewwolak@gmail.com>

Examples

```
mod11 <- gremlinR(WWG11 ~ sex - 1,
random = ~ calf,
data = Mrode11,
Gstart = matrix(0.1), Rstart = matrix(0.4),
maxit = 50, v = 2, algit = "EM")
logLik(mod11)
```

---

gremlinR   *Mixed-effect modeling functions.*

Description

Create and fit linear mixed-effect model (Gaussian data) or checking if an object is a fitted model.

Usage

```
gremlinR(formula, random = NULL, rcov = ~units, data = NULL,
  ginverse = NULL, Gstart = NULL, Rstart = NULL, Bp = NULL,
  maxit = 20, algit = NULL, vit = 10, v = 1, ...)

mkModMats(formula, random = NULL, rcov = ~units, data = NULL,
  subset = NULL, ginverse = NULL, na.action = na.pass, offset = NULL,
  contrasts = NULL, Xsparse = TRUE, ...)
```

Arguments

formula	A formula for the response variable and fixed effects.
random	A formula for the random effects.
rcov	A formula for the residual covariance structure.
data	A data.frame in which to look for the terms in formula, random, and rcov.
ginverse	A list of (preferably sparse) inverse matrices that are proportional to the covariance structure of the random effects. The name of each element in the list should match a column in data that is associated with a random term. All levels of the random term should appear as rownames for the matrices.
Gstart	A list of starting (co)variance values for the the G-structure or random terms.
Rstart	A list of starting (co)variance values for the R-structure or residual terms.
Bp	A prior specification for fixed effects.
maxit	An integer specifying the maximum number of likelihood iterations.
algit	A character vector of length 1 or more or an expression to be evaluated that specifies the algorithm to use for proposing (co)variances in the next likelihood iteration.
vit	An integer value specifying the verbosity of screen output on each iteration. A value of zero gives no iteration specific output and larger values increase the amount of information printed on the screen.
v	An integer value specifying the verbosity of screen output regarding the model fitting process. A value of zero gives no details and larger values increase the amount of information printed on the screen.
...	Additional arguments to be passed to control the model fitting.
subset	An expression for the subset of data to use.
na.action	What to do with NAs.

offset	Should an offset be specified.
contrasts	Specify the type of contrasts for the fixed effects.
Xsparse	Should sparse matrices be used for the fixed effects design matrix.

Value

A list of class gremlin or gremlinModMats:

call The model call.

modMats A list of the model matrices used to construct the mixed model equations.

y The response vector.

ny The number of responses.

ncy The number of columns of the original response.

X The fixed effects design matrix.

nb The number of columns in X.

Zr The residual design matrix.

Zg A list of the design matrices for each random term.

nG The number of parameters in the G structure.

listGeninv A list of generalized inverse matrices.

logDetG The log-determinants of the generalized inverse matrices - necessary to calculate the log-likelihood.

itMat A matrix of details about each iteration.

sln A two column matrix of solutions and their sampling variances from the mixed model.

residuals A vector of residual deviations, response minus the values expected based on the solutions, corresponding to the order in modMats$y.

theta A matrix of (co)variance components at the last iteration.

AI A matrix of values containing the Average Information matrix, or second partial derivatives of the likelihood with respect to the (co)variance components. The inverse of this matrix gives the sampling variances of the (co)variance components.

dLdtheta A single column matrix of first derivatives of the (co)variance parameters with respect to the log-Likelihood.

Functions

- mkModMats: Generates model matrices.

Author(s)

<matthewwolak@gmail.com>

References

Henderson Mrode. 2005.

Examples

```
library(nadiv)
Ainv <- makeAinv(Mrode3[-c(1:2), 1:3])$Ainv
mod11 <- gremlinR(WWG11 ~ sex - 1,
 random = ~ calf,
 data = Mrode11,
 ginverse = list(calf = Ainv),
 Gstart = matrix(0.2), Rstart = matrix(0.4),
 maxit = 10, v = 2)

is(mod11)
```

---

logLik.gremlin   *Methods to extract log-likelihood and information criterion of a grem-
                              lin model.*

Description

Extracts the log-likelihood or AIC from a gremlin model fit.

Usage

```
## S3 method for class 'gremlin'
logLik(object, ...)

## S3 method for class 'gremlin'
AIC(object, ..., k = 2, fxdDf = FALSE)
```

Arguments

object	An object of class 'gremlin'.
...	Additional arguments.
k	A numeric value for the penalty per parameter. Default is 2, as in classic AIC.
fxdDf	A logical indicating whether to penalize according to the number of fixed effect parameters. Since only models fit by REML can be compared, these must always be the same and so become a constant. Hence, the default is FALSE.

Value

A numeric value for either the log-likelihood and the number of parameters estimated by the model (sum of fixed effects and random effect (co)variance components) or Akaike's Information Criterion.

Author(s)

<matthewwolak@gmail.com>

Examples

```
mod11 <- gremlinR(WWG11 ~ sex - 1,
random = ~ calf,
data = Mrode11,
Gstart = matrix(0.1), Rstart = matrix(0.4),
maxit = 50, v = 2, algit = "EM")
logLik(mod11)
AIC(mod11)
```

Mrode11	*Weight gain data.*

Description

Data from chapter 11 in Mrode 2005. The variables are as follows:

Usage

```
Mrode11
```

Format

An object of class `data.frame` with 5 rows and 5 columns.

Details

- calf. a factor with levels 4 5 6 7 8
- dam. a factor with levels 2 5 6
- sire. a factor with levels 1 3 4
- sex. a factor with levels `male` `female`
- WWG11. a numeric vector

Source

Mrode. 2005.

Examples

```
data(Mrode11)
```

---

nobs.gremlin   *Number of observations in data from gremlin model fit objects*

Description

Extract the number of 'observations' in a gremlin model fit.

Usage

```
## S3 method for class 'gremlin'
nobs(object, use.fallback = FALSE, ...)
```

Arguments

object	An object of class 'gremlin'.
use.fallback	logical: should fallback methods be used to try to guess the value? Included for compatibility.
...	Further arguments to be passed to the methods.

Value

A single number, usually an integer, but can be NA.

Author(s)

<matthewwolak@gmail.com>

Examples

```
mod11 <- gremlinR(WWG11 ~ sex - 1,
random = ~ calf,
data = Mrode11,
Gstart = matrix(0.1), Rstart = matrix(0.4),
maxit = 50, v = 2, algit = "EM")
nobs(mod11)
```

---

residuals.gremlin   *Residuals of* class *'gremlin'*

Description

Residuals of class 'gremlin'.

Usage

```
## S3 method for class 'gremlin'
residuals(object, type = "response", scaled = FALSE, ...)
```

Arguments

object	An object of `class` 'gremlin'.
type	The type of residuals which should be returned. Only implement "response" currently. Can be abbreviated.
scaled	Logical value indicating whether to scale residuals by the residual standard deviation.
...	Additional arguments.

Value

A `numeric` vector of residuals.

Author(s)

`<matthewwolak@gmail.com>`

Examples

```
mod11 <- gremlinR(WWG11 ~ sex - 1,
random = ~ calf,
data = Mrode11,
Gstart = matrix(0.1), Rstart = matrix(0.4),
maxit = 50, v = 2, algit = "EM")
residuals(mod11)
```

---

stTrans	*Transformation of starting parameters.*

Description

Transform start parameters into format `gremlin` expects.

Usage

```
stTrans(x)
```

Arguments

x	A `list` of starting parameters.

Value

A sparse 'dsCMatrix'

Author(s)

`<matthewwolak@gmail.com>`

summary.gremlin   *Gremlin model summary.*

Description

Summarize and print results of linear mixed model fitted with gremlin.

Usage

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

## S3 method for class 'summary.gremlin'
print(x, digits = max(3, getOption("digits") - 3),
  ...)
```

Arguments

object, x	An object of `class` 'gremlin' or 'summary.gremlin'.
...	Additional arguments to be passed to control the output.
digits	An `integer` used for number formatting with 'signif()'.

Value

A `list` of class `summary.gremlin` or a printed value to the screen with no return values.

logLik Model log-likelihood.

formulae Model fixed, random, and residual formulae.

varcompSummary Table of variance components and approximate standard errors (calculated from the inverse of the average information matrix).

fxdSummary Table of fixed effects and standard errors (calculated from the corresponding diagonal elements of the inverse of the coefficient matrix).

Author(s)

<matthewwolak@gmail.com>

See Also

gremlin

---

tr *Matrix trace methods.*

Description

Methods to efficiently calculate a matrix trace depending on the class of matrix.

Usage

```
tr(X, ...)

## Default S3 method:
tr(X, ...)

## S3 method for class 'dgCMatrix'
tr(X, ...)

## S3 method for class 'dsCMatrix'
tr(X, ...)
```

Arguments

X	A matrix.
...	Additional arguments.

Value

A `numeric` value for the sum of the diagonal elements.

Methods (by class)

- `default`: Default method
- `dgCMatrix`: Method for matrix X of class Matrix:::dgCMatrix
- `dsCMatrix`: Method for matrix X of class Matrix:::dsCMatrix

Author(s)

`<matthewwolak@gmail.com>`

Examples

```
M <- matrix(seq(9), nrow = 3)
tr(M)
# Create sparse 'Matrix'
dgcM <- as(M, "dgCMatrix")
# compare `tr()` methods
tr(M) == tr(dgcM)  #<-- should be TRUE
```

---

vech2matlist   *Vector to list of matrices.*

Description

Converts a vector of (co)variance parameters to a list of covariance matrices.

Usage

```
vech2matlist(vech, skeleton)
```

Arguments

vech   A vector of (co)variance parameters.

skeleton   An example structure to map vech onto.

Value

A list of matrices of the same structure as skeleton.

Author(s)

<matthewwolak@gmail.com>

Index