Package ‘mbest’

October 13, 2022

Version 0.6
Date 2018-05-24
Title Moment-Based Estimation for Hierarchical Models
Maintainer Patrick O. Perry <patperry@gmail.com>
Description Fast moment-based hierarchical model fitting. Implements methods from the papers
``Fast Moment-Based Estimation for Hierarchical Models,'' by Perry (2017)
and
``Fitting a Deeply Nested Hierarchical Model to a Large Book Review Dataset Using a Moment-Based Estimator,'' by Zhang, Schmaus, and Perry (2018).
Depends nlme (>= 3.1-124)
Imports abind, bigmemory, foreach, lme4, logging
Suggests testthat
LazyData Yes
License Apache License (= 2.0) | file LICENSE
URL https://github.com/patperry/r-mbest
Encoding UTF-8
NeedsCompilation yes
Author Patrick O. Perry [aut, cre],
Timothy Sweetser [ctb],
Kyle Schmaus [ctb],
Ningshan Zhang [aut, ctb]
Repository CRAN
Date/Publication 2018-05-25 07:26:56 UTC

R topics documented:

  mbest-package .......................................................... 2
effects ................................................................. 3
Description


Details

Package: mbest
Version: 0.6
Date: 2018-05-24
Title: Moment-Based Estimation for Hierarchical Models
Authors@R: c(person(c("Patrick", "O."), "Perry", role = c("aut", "cre"), email = "patperry@gmail.com"), person("Timothy", "Sweetser", role = "ctb"), person("Kyle", "Schmaus", role = "ctb"), person("Ningshan", "Zhang", role = c("aut", "ctb"))
Maintainer: Patrick O. Perry <patperry@gmail.com>
Depends: nlme (>= 3.1-124)
Imports: abind, bigmemory, foreach, lme4, logging
Suggests: testthat
LazyData: Yes
License: Apache License (== 2.0) | file LICENSE
URL: https://github.com/patperry/r-mbest
Encoding: UTF-8
Author: Patrick O. Perry [aut, cre], Timothy Sweetser [ctb], Kyle Schmaus [ctb], Ningshan Zhang [aut, ctb]

Index of help topics:

firthglm.fit  Fitting Generalized Linear Models with Firth's Bias Reduction
fixef       Mixed Effects
mbest-package Moment-Based Estimation for Hierarchical Models
mhglm       Fitting Moment Hierarchical Generalized Linear Models
mhglm.control Auxiliary for Controlling Moment Hierarchical
GLM Fitting

mhglm_sim Simulate response patterns
model.matrix.mhglm Terms and Model Matrix
predict.mhglm Prediction

Basic usage is to call `mhglm`.

References

P. O. Perry (2017) "Fast moment-based estimation for hierarchical models."

See Also


effects Mixed Effects

Description

Get the fixed effects, random effect variances, and empirical Bayes random effect estimates.

Usage

```r
## S3 method for class 'mhglm'
fixef(object, ...)

## S3 method for class 'mhglm'
ranef(object, condVar = FALSE, ...)

## S3 method for class 'mhglm'
vcov(object, ...)

## S3 method for class 'mhglm_ml'
fixef(object, ...)

## S3 method for class 'mhglm_ml'
ranef(object, condVar = FALSE, ...)

## S3 method for class 'mhglm_ml'
vcov(object, ...)
```
## S3 method for class 'mhglm_ml'
VarCorr(x, sigma = 1, ...)

Arguments

- **object, x**: an mhglm object.
- **sigma**: a factor by which to scale the random effect variance-covariance matrix.
- **condVar**: a logical indicating whether conditional covariance matrices for the random effects should be returned.
- **...**: further arguments passed to or from other methods.

Details

- `fixef` returns the fixed effects, while `vcov` returns the variance-covariance matrix of the fixed effect estimates.
- `VarCorr` returns the random effect covariance matrix. `ranef` returns the empirical Bayes random effect estimates.

These functions behave like their counterparts in the `nlme` package.

See Also

- `fixef`, `ranef`, `VarCorr`, from package `nlme`.

---

### Description

A drop-in replacement for `glm.fit` which uses Firth’s bias-reduced estimates instead of maximum likelihood.

### Usage

```r
firthglm.fit(x, y, weights = rep(1, nobs),
             start = NULL, etastart = NULL, mustart = NULL,
             offset = rep(0, nobs), family = gaussian(),
             control = list(...), intercept = TRUE, singular.ok = TRUE, ...)
```

```r
firthglm.control(epsilon = 1e-8, maxit = 25, qr.tol = 1e-7,
                 improve.tol = 1e-4, curvature.tol = 0.9,
                 linesearch.method = "linesearch",
                 linesearch.maxit = 20, trace = FALSE)
```
Arguments

x, y, weights, start, etastart, mustart, offset, family, control, intercept, singular.ok, ...
arguments that have the same functions as for glm.fit.

qr.tol  
tolerance parameter for determining the rank of x.

epsilon, maxit  convergence parameters for the quasi-Newton method.

linesearch.method  
line search methods (one of "linesearch", "backtrack", or "blindsearch")

improve.tol, curvature.tol, linsearch.maxit  
tolerance parameters for the linesearch procedure.

trace  
logical indicating if output should be produced for each iteration.

Details

Firth's modified score function gives rise to estimates with smaller biases than their maximum likelihood counterparts. Unlike the maximum likelihood estimates, if the design matrix is of full rank, then the Firth bias-reduced estimate is finite.

By default, the fitting procedure uses a quasi-Newton optimization method, with a More-Thuente linesearch.

Value

firthglm.fit returns an object having the same components that a call to glm.fit would produce.

Note

Currently, only families with canonical link functions are supported.

Author(s)

Patrick O. Perry

References


See Also

logistf (package logistf) and brglm (package brglm) for alternative implementations of Firth’s bias-reduced estimators.
## Examples

```r
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)

# Use firthglm.fit instead of glm.fit:
glm.D93 <- glm(counts ~ outcome + treatment, family = poisson(), method="firthglm.fit")
summary(glm.D93)
```

---

### mhglm

#### Description

mhglm is used to fit a moment hierarchical generalized linear model of one level. mhglm_ml is used to fit a moment hierarchical generalized linear model of arbitrary number of levels (including one level).

#### Usage

```r
mhglm(formula, family = gaussian, data, weights, subset, na.action, start = NULL, etastart, mustart, offset, control = list(), model = TRUE, method = "mhglm.fit", x = FALSE, z = FALSE, y = TRUE, group = TRUE, contrasts = NULL)
```

```r
mhglm.fit(x, z, y, group, weights = rep(1, nobs), start = NULL, etastart = NULL, mustart = NULL, offset = rep(0, nobs), family = gaussian(), control = list(), intercept = TRUE, dispersion = NULL)
```

```r
mhglm_ml(formula, family = gaussian, data, weights, subset, na.action, start = NULL, etastart, mustart, offset, control = list(), model = TRUE, method = "mhglm_ml.fit", x = FALSE, z = FALSE, y = TRUE, group = TRUE, contrasts = NULL)
```

```r
mhglm_ml.fit(x, z, y, group, weights = rep(1, nobs), start = NULL, etastart = NULL, mustart = NULL, offset = rep(0, nobs), family = gaussian(), control = list(), intercept = TRUE)
```
**Arguments**

- **formula**: A symbolic description of the model to be fitted. This includes the numeric response variable, a series of terms which describe the fixed effects, and random effects terms that are specified as (random_effects | grouping_factor).

- **family**: A description of the error distribution and link function to be used in the model. This should be a description of a family object (see family in ?glm).

- **data**: A data frame containing the variables needed to fit the model. This data frame should contain all the variables specified in the formula.

- **weights**: An optional vector of prior weights for the response. These weights are used with a generalized linear model and need to be proportional to the expected values of the response.

- **subset**: A logical expression that can be used to select a subset of the data upon which the model is to be fitted.

- **na.action**: A function that indicates what should happen when the data contain missing values.

- **start**: Starting values for the solver. These are used by the algorithm to get started.

- **etastart**: Starting values for the effective degrees of freedom. These are used for calculations involving effective degrees of freedom.

- **mustart**: Starting values for the fitted values. These are used for calculations involving the fitted values.

- **offset**: A function determining what part of the model formula represents offset terms.

- **model**: Logical. If TRUE the model frame is returned, otherwise the model frame is not returned.

- **contrasts**: A list of contrasts to be used when the model frame is evaluated. These are used for the model matrix.

- **intercept**: Logical. If TRUE, the intercept is included in the model. If FALSE, the intercept is omitted.

- **control**: A list of parameters for controlling the fitting process. For mhglm.fit, this is passed to mhglm.control.

- **method**: The method to be used in fitting the model. The default method "mhglm.fit" uses moment-based estimates; the alternative "model.frame" returns the model frame and does no fitting.

- **x, z, y, group**: For mhglm and mhglm_ml: logical values indicating whether the response vector, model matrices, and grouping factor used in the fitting process should be returned as components of the returned value.

  For mhglm.fit: x is a fixed effect design matrix of dimension n * p, z is a random effect design matrix of dimension n * q, y is a vector of observations of length n, and group is a grouping factor vector of length n.

  For mhglm_ml.fit: x is a fixed effect design matrix of dimension n * p, z is a list of L elements, with L the depth of nested hierarchies, each element of z is a random effect design matrix of dimension n * q_i, with q_i the feature dimension on tree depth i, y is a vector of observations of length n, and group is a list of L elements (same L as z), each element of group is a grouping factor vector of length n.

- **dispersion**: If NULL, will estimate from data; otherwise use this argument as dispersion parameter.

**Details**

These functions are analogues of glm and glm.fit, meant to be used for fitting hierarchical generalized linear models. A typical predictor has the form response ~ terms + (reterms | group) where response is the (numeric) response vector, terms is a series of terms which specifies a linear predictor for response, reterms is a series of terms with random coefficients (effects), and group is a grouping factor; observations with the same grouping factor share the same random effects.

mhglm and mhglm.fit only allow one random effect term, along with a single level of hierarchy. mhglm_ml and mhglm_ml.fit allow multiple random effect terms so long as levels of random effects are hierarchically nested. If the random effect design matrices are the same for each level, a predictor has the form response ~ terms + (reterms | g1/.../gQ). If the random effects design matrices differ from level to level, colons are used to delineate the nesting structure; for example, response ~ fe + (re1 | g1) + (re2 | g2:g1) + (re3 | g3:g2:g1).

mhglm allows || in the formula response ~ terms + (reterms || group) to indicate that random effects are independent, that is the random effects covariance matrix has non-zero value only on its diagonal. mhglm_ml currently does not support ||, to indicate independent random effects, set control=list(diagcov = TRUE).

**Value**

mhglm returns an object of class inheriting from "mhglm". mhglm_ml returns an object of class inheriting from "mhglm_ml".
The function `summary` can be used to obtain or print a summary of the results.

The generic accessor functions `fixef`, `ranef`, `VarCorr`, `sigma`, `fitted.values` and `residuals` can be used to extract various useful features of the value returned by `mhglm` or `mhglm_ml`.

Note

If the moment-based random effect covariance is not positive-semidefinite, then a warning will be issued, and a projection of the estimate to the positive-semidefinite cone will be used instead.

References

P. O. Perry (2017) "Fast moment-based estimation for hierarchical models."


See Also

`terms.mhglm`, `model.matrix.mhglm`, and `predict.mhglm` for `mhglm` methods, and the generic functions `fitted.values`, `residuals`, `summary`, `vcov`, and `weights`.

Generic functions `fixef`, `ranef`, `VarCorr`, and `sigma` for features related to mixed effect models.

`glmer` (package `lme4`) for fitting generalized linear mixed models with likelihood-based estimates.

Examples

```r
library(lme4)
## The following examples are adapted from lme4:
(fm1 <- mhglm(Reaction ~ Days + (Days | Subject), gaussian, sleepstudy))
(fm2 <- mhglm(Reaction ~ Days + (Days || Subject), gaussian, sleepstudy))
(gm <- mhglm(cbind(incidence, size - incidence) ~ period + (1 | herd),
          data = cbpp, family = binomial))

## The following examples are for multilevel models
g_data <- mhglm_sim(n = 30, m_per_level = c(10, 5, 2), sd_intercept = c(1, 1, 1),
                    sd_slope = c(1, 1, 1), family = "gaussian", seed = 12345)

(m1 <- mhglm_ml(y ~ 1 + x + (1 + x | g1/g2/g3), gaussian, g_data))
# or equivalent
(m2 <- mhglm_ml(y ~ 1 + x + (1 + x | g1) + (1 + x | g2:g1) + (1 + x | g3:g2:g1),
                gaussian, g_data))
```
mhglm.control

Auxiliary for Controlling Moment Hierarchical GLM Fitting

Description

Auxiliary function for mhglm fitting. Typically only used internally by mhglm.fit, but may be used to construct a control argument to either function.

Usage

mhglm.control(standardize = TRUE, steps = 1, parallel = FALSE, diagcov = FALSE, fit.method = "firthglm.fit", fixef.rank.warn = FALSE, cov.rank.warn = FALSE, cov.psd.warn = TRUE, fit.control = list(...), ...)

mhglm_ml.control(standardize = FALSE, steps = 1, parallel = FALSE, diagcov = FALSE, fit.method = "firthglm.fit", fixef.rank.warn = FALSE, cov.rank.warn = FALSE, cov.psd.warn = FALSE, fit.control = list(...), ...)

Arguments

standardize logical indicating if predictors should be standardized before moment-based fitted
steps number of refinement steps
parallel fit the group-specific estimates in parallel rather than sequentially
diagcov estimate random effect covariance matrix with diagonal approximation
fit.method method for obtaining group-specific effect estimates
fixef.rank.warn if TRUE, print warnings when fixef is unidentifiable
cov.rank.warn if TRUE, print warnings when covariance matrix is unidentifiable
cov.psd.warn if TRUE, print warnings when moment based estimates of covariance matrix is not positive semi-definite
fit.control control parameters for fit.method
... arguments to be used to form the fit.control argument if it is not supplied directly.

Details

Setting standardize = TRUE ensures that the procedure is equivariant, and generally leads to better estimation performance. Right now standardize = TRUE is not allowed for mhglm_ml.

The steps argument gives the number of refinement steps for the moment based parameters. In each step, the previous fixed effect and random effect covariance matrix estimates are used to weight the subpopulation-specific effect estimates. In principle, higher values of steps could lead to more accurate estimates, but in simulations, the differences are negligible.
Value
A list with components named as the arguments.

See Also
mhglm.fit, the fitting procedure used by mhglm.
firthglm.fit, the default subpopulation-specific fitting method.

Examples
library(lme4) # for cbpp data

# The default fitting method uses Firth’s bias-corrected estimates
(gm.firth <- mhglm(cbind(incidence, size - incidence) ~ period + (1 | herd),
                   data = cbpp, family = binomial,
                   control=mhglm.control(fit.method="firthglm.fit")))

# Using maximum likelihood estimates is less reliable
(gm.ml <- mhglm(cbind(incidence, size - incidence) ~ period + (1 | herd),
                data = cbpp, family = binomial,
                control=mhglm.control(fit.method="glm.fit")))

Description
Simulate response patterns for generalized linear models of gaussian or binomial families, with both an intercept and slope covariate. Used primarily for testing purposes.

Usage
mhglm_sim(n, m_per_level, sd_intercept, sd_slope,
          family = c("gaussian", "binomial"), seed)

Arguments
n an integer scalar, the number of observations at the lowest grouping level.
m_per_level an integer vector, the number of grouping levels nested under the level above.
sd_intercept a numeric vector, the standard deviations of the intercept random effects.
sd_slope a numeric vector, the standard deviations of the slope random effects.
family a character scalar, either "gaussian" or "binomial".
seed a single value, interpreted as an integer, or NULL as in set.seed.

Details
returns a data.frame with design matrix, response, and group levels.
Examples

mhglm_sim(n = 2, m_per_level = c(3, 3), sd_intercept = c(1, 2),
  sd_slope = c(2, 1), family = "gaussian", seed = 123)

mhglm_sim(n = 2, m_per_level = c(3, 3), sd_intercept = c(1, 2),
  sd_slope = c(2, 1), family = "binomial", seed = 123)

model.matrix.mhglm

Terms and Model Matrix

Description

Get the terms or model matrix from an mhglm object.

Usage

## S3 method for class 'mhglm'
model.matrix(object, type = c("fixed", "random"), ...)
## S3 method for class 'mhglm.ml'
model.matrix(object, type = c("fixed", "random"), ...)

## S3 method for class 'mhglm'
terms(x, type = c("fixed", "random"), ...)
## S3 method for class 'mhglm.ml'
terms(x, type = c("fixed", "random"), ...)

Arguments

object, x an mhglm object.
type which terms to get (for the fixed or for the random effects).
... further arguments passed to or from other methods.

See Also

model.matrix, terms
predict gives empirical Bayes predictions of the response, while sigma gives the dispersion parameter.

Usage

```r
## S3 method for class 'mhglm'
predict(object, newdata = NULL, type = c("link", "response"),
    se.fit = FALSE, na.action = na.pass, ...)

## S3 method for class 'mhglm'
sigma(object, ...)
```

Arguments

- `object`: an `mhglm` object
- `newdata`, `type`, `se.fit`, `na.action`: these arguments behave as in `predict.glm`. See Details, below.
- `...`: further arguments passed to or from other methods.

Details

The `predict` function gives empirical Bayes posterior mean estimates of response values. If `se.fit = TRUE`, then the conditional variances of the random effects are used along with the fixed effect variance-covariance matrix to estimate the standard errors.

The `sigma` function gives the square root of the dispersion parameter or the model; for linear models, this is the error standard deviation.

See Also

- `predict`
- `sigma`, from package `lme4`
Index

* datagen
  * mhglm_sim, 10

* models
  * effects, 3
    * firthglm.fit, 4
    * mhglm, 6
    * mhglm.control, 9
    * model.matrix.mhglm, 11
    * predict, 12
  * optimize
    * mhglm.control, 9
  * package
    * mbest-package, 2
* regression
  * effects, 3
    * firthglm.fit, 4
    * mhglm, 6
  * firthglm.control (firthglm.fit), 4
  * firthglm.fit, 4, 10
  * fitted.values, 8
  * fixef, 4, 8
  * fixef (effects), 3
  * fixef.mhglm, 3
  * glm, 7
  * glm.fit, 4, 5, 7
  * glmer, 8
  * logistf, 5
  * mbest (mbest-package), 2
    * mbest-package, 2
    * mhglm, 3, 6, 9, 10
    * mhglm.control, 7, 9
    * mhglm.fit, 9, 10
    * mhglm_ml (mhglm), 6
    * mhglm_ml.control (mhglm.control), 9
  * mhglm_sim, 10
  * model.matrix, 11
  * model.matrix.mhglm, 8, 11
  * model.matrix.mhglm_ml
    * (model.matrix.mhglm), 11
  * predict, 12, 12
  * predict.glm, 12
  * predict.mhglm, 3, 8
  * ranef, 4, 8
  * ranef (effects), 3
  * ranef.mhglm, 3
  * residuals, 8
  * sigma, 8, 12
  * sigma (predict), 12
  * summary, 8
  * terms, 11
  * terms.mhglm, 8
  * terms.mhglm (model.matrix.mhglm), 11
  * terms.mhglm_ml (model.matrix.mhglm), 11
  * VarCorr, 4, 8
  * VarCorr (effects), 3
  * VarCorr.mhglm, 3
  * vcov, 8
  * vcov.mhglm (effects), 3
  * vcov.mhglm_ml (effects), 3
  * weights, 8