

# Package ‘mbest’

August 29, 2016

**Version** 0.5

**Date** 2016-03-07

**Title** Moment-Based Estimation for Hierarchical Models

**Maintainer** Patrick O. Perry <pperry@stern.nyu.edu>

**Description** Implements methods from the paper  
“Fast Moment-Based Estimation for Hierarchical Models,” by Perry (2016).

**Depends** nlme (>= 3.1-124)

**Imports** lme4, bigmemory, foreach, 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]

**Repository** CRAN

**Date/Publication** 2016-03-08 06:22:53

## R topics documented:

mbest-package . . . . .	2
effects . . . . .	2
firthglm.fit . . . . .	3
mhglm . . . . .	5
mhglm.control . . . . .	7
model.matrix.mhglm . . . . .	8
predict . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

mbest-package

*Fast Moment-Based Estimation for Hierarchical Models*

---

### Description

This package implements moment-based fitting methods for hierarchical models described in the paper "Fast Moment-Based Estimation for Hierarchical Models" (Perry, 2015).

### Details

Package: mbest  
Type: Package  
Version: 0.3  
Date: 2015-04-20  
License: Apache License, Version 2.0

Basic usage is to call [mhglm](#).

### Author(s)

Patrick O. Perry

### See Also

[mhglm](#), [fixef.mhglm](#), [ranef.mhglm](#), [VarCorr.mhglm](#), [predict.mhglm](#)

---

effects

*Mixed Effects*

---

### Description

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

### Usage

```
## 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'
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**.

---

firthglm.fit

*Fitting Generalized Linear Models with Firth's Bias Reduction*

---

### Description

A drop-in replacement for [glm.fit](#) which uses Firth's bias-reduced estimates instead of maximum likelihood.

### Usage

```
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, ...)
```

```
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`, `linesearch.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**

Firth, D. (1993) Bias reduction of maximum likelihood estimates. *Biometrika* **80**, 27--38.

More, J. J. and Thuente, D. J. (1994) Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software* **20** 286-307.

**See Also**

`logistf` (package `logistf`) and `brglm` (package `brglm`) for alternative implementations of Firth's bias-reduced estimators.

**Examples**

```
## Dobson (1990) Page 93: Randomized Controlled Trial :
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

*Fitting Moment Hierarchical Generalized Linear Models***Description**

mhglm is used to fit a moment hierarchical generalized linear model.

**Usage**

```
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)

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)
```

**Arguments**

formula, family, data, weights, subset, na.action, start, etastart, mustart, offset, model, contrasts

These arguments are analogous to the similarly-named arguments for the [glm](#) and [glm.fit](#) functions.

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: 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.

## 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.

Currently, only one random effect term is allowed, along with a single level of hierarchy; random effect terms of the form `reterms | g1/. . ./gQ` are not supported.

## Value

`mhglm` returns an object of class inheriting from `"mhglm"`.

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`.

## 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.

## Author(s)

Patrick O. Perry

## References

Perry, P. O. (2015) "Fast Moment-Based Estimation for Hierarchical Models", Preprint.

## 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

```
library(lme4)
## The following examples are adapted from lme4:
(fm1 <- mhglm(Reaction ~ Days + (Days | Subject), gaussian, sleepstudy))

(gm <- mhglm(cbind(incidence, size - incidence) ~ period + (1 | herd),
  data = cbpp, family = binomial))
```

**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,  
              fit.method = "firthglm.fit", fit.control = list(...), ...)
```

**Arguments**

<code>standardize</code>	logical indicating if predictors should be standardized before moment-based fitted
<code>steps</code>	number of refinement steps
<code>parallel</code>	fit the group-specific estimates in parallel rather than sequentially
<code>fit.method</code>	method for obtaining group-specific effect estimates
<code>fit.control</code>	control parameters for <code>fit.method</code>
<code>...</code>	arguments to be used to form the <code>fit.control</code> argument if it is not supplied directly.

**Details**

Setting `standardize = TRUE` ensures that the procedure is equivariant, and generally leads to better estimation performance.

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"))
```

---

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'
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	<i>Prediction</i>
---------	-------------------

---

### Description

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

### Usage

```
## 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 <a href="#">predict.glm</a> . 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

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