Package ‘glmmML’

October 27, 2022

Encoding UTF-8
Version 1.1.4
Date 2022-10-26
Title Generalized Linear Models with Clustering
Description Binomial and Poisson regression for clustered data, fixed
and random effects with bootstrapping.
License GPL (>= 3)
Depends R (>= 2.13.0)
Maintainer Göran Broström <goran.brostrom@umu.se>
NeedsCompilation yes
Suggests knitr, rmarkdown, lme4
VignetteBuilder knitr
Author Göran Broström [aut, cre],
       Jianming Jin [ctb],
       Henrik Holmberg [ctb]
Repository CRAN
Date/Publication 2022-10-26 22:25:09 UTC

R topics documented:

ghq ................................................................. 2
glmmboot ....................................................... 3
glmmbootFit .................................................. 5
glmmML .......................................................... 6
  glmmML.fit .................................................. 9
  print.glmmboot .............................................. 10
  print.glmmML ................................................. 11
  summary.glmmboot ........................................... 12
  summary.glmmML ............................................ 13

Index 14
ghq

Gauss-Hermite

Description
Calculates the zeros and weights needed for Gauss-Hermite quadrature.

Usage
ghq(n.points = 1, modified = TRUE)

Arguments

n.points Number of points.
modified Multiply by exp(zeros**2)? Default is TRUE.

Details
Based on a Fortran 66 subroutine written by professor Jianming Jin.

Value
A list with components

zeros The zeros (abscissas).
weights The weights

Note
The code is modified to suit the purpose of glmmML, with the permission of professor Jin.

Author(s)
Jianming Jin, Univ. of Illinois, Urbana-Campaign

References
Gauss-Hermite

See Also
glmmML

Examples
ghq(15, FALSE)
**Description**

Fits grouped GLMs with fixed group effects. The significance of the grouping is tested by simulation, with a bootstrap approach.

**Usage**

```r
glmboot(formula, family = binomial, data, cluster, weights, subset, na.action,
        offset, contrasts = NULL, start.coef = NULL,
        control = list(epsilon = 1e-08, maxit = 200, trace = FALSE), boot = 0)
```

**Arguments**

- `formula`: a symbolic description of the model to be fit. The details of model specification are given below.
- `family`: Currently, the only valid values are binomial and poisson. The binomial family allows for the logit and cloglog links.
- `data`: an optional data frame containing the variables in the model. By default, the variables are taken from `environment(formula)`, typically the environment from which `glmML` is called.
- `cluster`: Factor indicating which items are correlated.
- `weights`: Case weights.
- `subset`: an optional vector specifying a subset of observations to be used in the fitting process.
- `na.action`: See glm.
- `offset`: this can be used to specify an a priori known component to be included in the linear predictor during fitting.
- `contrasts`: an optional list. See the 'contrasts.arg' of 'model.matrix.default'.
- `start.coef`: starting values for the parameters in the linear predictor. Defaults to zero.
- `control`: Controls the convergence criteria. See `glm.control` for details.
- `boot`: number of bootstrap replicates. If equal to zero, no test of significance of the grouping factor is performed.

**Details**

The simulation is performed by simulating new response vectors from the fitted probabilities without clustering, and comparing the maximized log likelihoods. The maximizations are performed by profiling out the grouping factor. It is a very fast procedure, compared to `glm`, when the grouping factor has many levels.
Value

The return value is a list, an object of class 'glmmboot'.

coefficients Estimated regression coefficients
logLik the max log likelihood
cluster.null.deviance Deviance without the clustering
frail The estimated cluster effects
bootLog The logLik values from the bootstrap samples
bootP Bootstrap p value
variance Variance covariance matrix
sd Standard error of regression parameters
boot_rep No. of bootstrap replicates
mixed Logical
deviance Deviance
df.residual Its degrees of freedom
aic AIC
boot Logical
call The function call

Note

There is no overall intercept for this model; each cluster has its own intercept. See frail

Author(s)

Göran Broström and Henrik Holmberg

References


See Also

link{glmmML}, optim, lmer in Matrix, and glmmPQL in MASS.

Examples

```r
## Not run:
id <- factor(rep(1:20, rep(5, 20)))
y <- rbinom(100, prob = rep(runif(20), rep(5, 20)), size = 1)
x <- rnorm(100)
dat <- data.frame(y = y, x = x, id = id)
res <- glmmboot(y ~ x, cluster = id, data = dat, boot = 5000)
## End(Not run)
##system.time(res.glm <- glm(y ~ x + id, family = binomial))
```
Description

'glmmbootFit' is the workhorse in the function glmmboot. It is suitable to call instead of 'glmmboot', e.g. in simulations.

Usage

```r
glmmbootFit(X, Y, weights = rep(1, NROW(Y)),
             start.coef = NULL, cluster = rep(1, length(Y)),
             offset = rep(0, length(Y)), family = binomial(),
             control = list(epsilon = 1.e-8, maxit = 200, trace = FALSE),
             boot = 0)
```

Arguments

- **X**: The design matrix (n * p).
- **Y**: The response vector of length n.
- **weights**: Case weights.
- **start.coef**: start values for the parameters in the linear predictor (except the intercept).
- **cluster**: Factor indicating which items are correlated.
- **offset**: this can be used to specify an a priori known component to be included in the linear predictor during fitting.
- **family**: Currently, the only valid values are `binomial` and `poisson`. The binomial family allows for the `logit` and `cloglog` links.
- **control**: A list. Controls the convergence criteria. See `glm.control` for details.
- **boot**: number of bootstrap replicates. If equal to zero, no test of significance of the grouping factor is performed. If non-zero, it should be large, at least, say, 2000.

Value

A list with components

- **coefficients**: Estimated regression coefficients (note: No intercept).
- **logLik**: The maximised log likelihood.
- **cluster.null.deviance**: deviance from a model without cluster.
- **frail**: The estimated cluster effects.
- **bootLog**: The maximised bootstrap log likelihood values. A vector of length `boot`.
- **bootP**: The bootstrap p value.
- **variance**: The variance-covariance matrix of the fixed effects (no intercept).
- **sd**: The standard errors of the coefficients.
- **boot_rep**: The number of bootstrap replicates.
Note

A profiling approach is used to estimate the cluster effects.

Author(s)

Göran Broström

See Also

glmmboot

Examples

```r
## Not run
x <- matrix(rnorm(1000), ncol = 1)
id <- rep(1:100, rep(10, 100))
y <- rbinom(1000, size = 1, prob = 0.4)
fit <- glmmbootFit(x, y, cluster = id, boot = 200)
summary(fit)
## End(Not run)
## Should show no effects. And boot too small.
```

---

**glmmML**

*Generalized Linear Models with random intercept*

**Description**

Fits GLMs with random intercept by Maximum Likelihood and numerical integration via Gauss-Hermite quadrature.

**Usage**

```r
glmmML(formula, family = binomial, data, cluster, weights, 
cluster.weights, subset, na.action, 
offset, contrasts = NULL, prior = c("gaussian", "logistic", "cauchy"), 
start.coef = NULL, start.sigma = NULL, fix.sigma = FALSE, x = FALSE, 
control = list(epsilon = 1e-08, maxit = 200, trace = FALSE), 
method = c("Laplace", "ghq"), n.points = 8, boot = 0)
```

**Arguments**

- **formula**: a symbolic description of the model to be fit. The details of model specification are given below.
- **family**: Currently, the only valid values are binomial and poisson. The binomial family allows for the logit and cloglog links.
- **data**: an optional data frame containing the variables in the model. By default the variables are taken from `environment(formula)`, typically the environment from which `glmmML` is called.
cluster Factor indicating which items are correlated.
weights Case weights. Defaults to one.
cluster.weights Cluster weights. Defaults to one.
subset an optional vector specifying a subset of observations to be used in the fitting process.
na.action See glm.
start.coef starting values for the parameters in the linear predictor. Defaults to zero.
start.sigma starting value for the mixing standard deviation. Defaults to 0.5.
fix.sigma Should sigma be fixed at start.sigma?
x If TRUE, the design matrix is returned (as x).
offset this can be used to specify an a priori known component to be included in the linear predictor during fitting.
contrasts an optional list. See the 'contrasts.arg' of 'model.matrix.default'.
prior Which "prior" distribution (for the random effects)? Possible choices are "gaussian" (default), "logistic", and "cauchy".
control Controls the convergence criteria. See glm.control for details.
method There are two choices "Laplace" (default) and "ghq" (Gauss-Hermite).
n.points Number of points in the Gauss-Hermite quadrature. If n.points == 1, the Gauss-Hermite is the same as Laplace approximation. If method is set to "Laplace", this parameter is ignored.
boot Do you want a bootstrap estimate of cluster effect? The default is No (boot = 0). If you want to say yes, enter a positive integer here. It should be equal to the number of bootstrap samples you want to draw. A recommended absolute minimum value is boot = 2000.

Details
The integrals in the log likelihood function are evaluated by the Laplace approximation (default) or Gauss-Hermite quadrature. The latter is now fully adaptive; however, only approximate estimates of variances are available for the Gauss-Hermite (n.points > 1) method.
For the binomial families, the response can be a two-column matrix, see the help page for glm for details.

Value
The return value is a list, an object of class 'glmmML'. The components are:

boot No. of boot replicates
converged Logical
coefficients Estimated regression coefficients
coef.sd Their standard errors
sigma The estimated random effects’ standard deviation
sigma.sd  Its standard error
variance The estimated variance-covariance matrix. The last column/row corresponds to
           the standard deviation of the random effects (sigma)
aic      AIC
bootP Bootstrap p value from testing the null hypothesis of no random effect (sigma =
           0)
deviance Deviance
mixed    Logical
df.residual Degrees of freedom
cluster.null.deviance Deviance from a glm with no clustering. Subtracting deviance gives a test
                      statistic for the null hypothesis of no clustering. Its asymptotic distribution is a
                      symmetric mixture a constant at zero and a chi-squared distribution with one df.
                      The printed p-value is based on this.
cluster.null.df Its degrees of freedom
posterior.modes Estimated posterior modes of the random effects
terms    The terms object
info     From hessian inversion. Should be 0. If not, no variances could be estimated.
           You could try fixing sigma at the estimated value and rerun.
prior    Which prior was used?
call     The function call
x        The design matrix if asked for, otherwise not present

Note
The optimization may not converge with the default value of start.sigma. In that case, try different
start values for sigma. If still no convergence, consider the possibility to fix the value of sigma at
several values and study the profile likelihood.

Author(s)
G"oran Brostr"om

References
Brostr"om, G. and Holmberg, H. (2011). Generalized linear models with clustered data: Fixed and
random effects models. Computational Statistics and Data Analysis 55:3123-3134.

See Also
glmmboot, glm, optim, lmer in Matrixand glmmPQL in MASS.
Examples
id <- factor(rep(1:20, rep(5, 20)))
y <- rbinom(100, prob = rep(runif(20), rep(5, 20)), size = 1)
x <- rnorm(100)
dat <- data.frame(y = y, x = x, id = id)
glmmML(y ~ x, data = dat, cluster = id)

Description
This function is called by glmmML, but it can also be called directly by the user.

Usage
glmmML.fit(X, Y, weights = rep(1, NROW(Y)), cluster.weights = rep(1, NROW(Y)),
start.coef = NULL, start.sigma = NULL,
fix.sigma = FALSE,
cluster = NULL, offset = rep(0, nobs), family = binomial(),
method = 1, n.points = 1,
control = list(epsilon = 1.e-8, maxit = 200, trace = FALSE),
intercept = TRUE, boot = 0, prior = 0)

Arguments
X
Y
weights
cluster.weights
start.coef
start.sigma
fix.sigma
cluster
offset
family
method
n.points
control
intercept
boot
prior

Design matrix of covariates.
Response vector. Or two-column matrix.
Case weights. Defaults to one.
Cluster weights. Defaults to one.
Starting values for the coefficients.
Starting value for the mixing standard deviation.
Should sigma be fixed at start.sigma?
The clustering variable.
The offset in the model.
Family of distributions. Defaults to binomial with logit link. Other possibilities
are binomial with cloglog link and poisson with log link.
Laplace (1) or Gauss-hermite (0)?
Number of points in the Gauss-Hermite quadrature. Default is n.points = 1,
which is equivalent to Laplace approximation.
Control of the iterations. See glm.control.
Logical. If TRUE, an intercept is fitted.
Integer. If > 0, bootstrapping with boot replicates.
Which prior distribution? 0 for "gaussian", 1 for "logistic", 2 for "cauchy".
print.glmmboot

Details

In the optimisation, "vmmin" (in C code) is used.

Value

A list. For details, see the code, and glmmML.

Author(s)

Göran Broström

References

Broström (2003)

See Also

glmmML, glmmPQL, and lmer.

Examples

x <- cbind(rep(1, 14), rnorm(14))
y <- rbinom(14, prob = 0.5, size = 1)
id <- rep(1:7, 2)

glmmML.fit(x, y, cluster = id)

print.glmmboot

Prints a 'glmmML' object.

Description

A glmmboot object is the output of glmmboot.

Usage

## S3 method for class 'glmmboot'
print(x, digits = max(3, getOption("digits") - 3), na.print = "", ...)

Arguments

  x          The glmmboot object
  digits     Number of printed digits.
  na.print   How to print NAs
  ...        Additional parameters, which are ignored.
print.glmmML

Details

Nothing in particular.

Value

A short summary of the object is printed.

Note

This is the only summary method available for the moment.

Author(s)

Göran Broström

See Also

glmmboot

print.glmmML

Prints a 'glmmML' object.

Description

A glmmML object is the output of glmmML.

Usage

```r
## S3 method for class 'glmmML'
pred(x, digits = max(3, getOption("digits") - 3), na.print = "", ...)
```

Arguments

- **x**: The glmmML object
- **digits**: Number of printed digits.
- **na.print**: How to print NAs
- **...**: Additional parameters, which are ignored.

Details

Nothing in particular.

Value

A short summary of the object is printed.
Note

This is the only summary method available for the moment.

Author(s)

Göran Broström

See Also

glmmML

summary.glmmboot

Summary of a glmmboot object

Description

It simply calls print.glmmboot

Usage

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

Arguments

object A glmmboot object
... Additional arguments

Details

A summary method will be written soon.

Value

Nothing is returned.

Note

Preliminary

Author(s)

Göran Broström

See Also

print.glmmboot
Summary of a `glmmML` object

Description

It simply calls `print.glmmML`

Usage

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

Arguments

- `object`: A `glmmML` object
- `...`: Additional arguments

Value

Nothing is returned.

Note

Preliminary

Author(s)

Göran Broström

See Also

`print.glmmML`
Index

* math
  ghq. 2
* nonlinear
  glmmboot, 3
  glmmbootFit, 5
* print
  print.glmmboot, 10
  print.glmmML, 11
  summary.glmmboot, 12
  summary.glmmML, 13
* regression
  glmmboot, 3
  glmmbootFit, 5
  glmmML, 6
  glmmML.fit, 9
ghq, 2
glm, 3, 8
glm.control, 3, 5, 7, 9
glmmboot, 3, 6, 8, 11
  glmmbootFit, 5
  glmmML, 2, 6, 10, 12
  glmmML.fit, 9
  glmmPQL, 4, 8, 10
lmer, 4, 8, 10
optim, 4, 8
print.glmmboot, 10, 12
print.glmmML, 11, 13
summary.glmmboot, 12
summary.glmmML, 13