Package ‘glmmML’

April 30, 2019

Encoding UTF-8
Version 1.1.0
Date 2019-04-30
Title Generalized Linear Models with Clustering
Description Binomial and Poisson regression for clustered data, fixed and random effects with bootstrapping.
License GPL (>= 3)
Author Göran Broström
Depends R (>= 2.13.0)
Maintainer Göran Broström <goran.brostrom@umu.se>
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-04-30 16:30:13 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
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 `glm` 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)

Goran Brostrom and Henrik Holmberg

References


See Also

link{glmmML}, optim, lmer in Matrix, and glmPQL 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))
```
glmmbootFit

**Generalized Linear Models with fixed effects grouping**

**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)),
startcoef = 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.
- `startcoef` 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 frailty 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
## 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
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 glmmML 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)
Göran Broström

References

See Also
glmmboot, glm, optim, lmer in Matrix and glmmPQL in MASS.
Examples

```r
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**

```r
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 = 1e-8, maxit = 200, trace = FALSE),
intercept = TRUE, boot = 0, prior = 0)
```

**Arguments**

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

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

print.glmmboot(x, y, cluster = id)
```

print.glmmboot  "Prints a 'glmmML' object."

Description

A glmmboot object is the output of glmmboot.

Usage

```r
## 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

Description

A glmmML object is the output of glmmML.

Usage

## S3 method for class 'glmmML'
print(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
```r
## 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

*Topic math
   ghq. 2
*Topic nonlinear
   glmmboot. 3
   glmmbootFit. 5
*Topic print
   print.glmmboot. 10
   print.glmmML. 11
   summary.glmmboot. 12
   summary.glmmML. 13
*Topic 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