Package ‘dhglm’

August 29, 2013

Type Package

Title Double Hierarchical Generalized Linear Models

Version 1.2

Date 2011-09-27

Author Maengseok Noh, Youngjo Lee

Maintainer Maengseok Noh <msnoh@pknu.ac.kr>

Description The dhglm package fits double hierarchical generalized linear models in which the mean, dispersion parameters for variance of random effects, and residual variance (overdispersion) can be further modeled as random-effect models.

Depends R (>= 2.10.0), Matrix, numDeriv, boot

License Unlimited

LazyLoad yes

Respository CRAN

Date/Publication 2011-09-30 05:51:38

Repository CRAN

NeedsCompilation no

R topics documented:

`dhglm-package` *Double Hierarchical Genearlized Linear Models*

Description

The dhglm package is used to fit double hierarchical generalized linear models (DHGLMs) in which random effects can be specified in both the mean and the dispersion components (Lee and Nelder, 2006; Lee, Nelder, and Pawitan, 2006). It can also be used to fit generalized linear models (GLMs) of Nedler and Wedderburn (1972), joint GLMs of Nelder and Lee (1991), and hierarchical GLMs (HGLMs) of Lee and Nelder (1996, 2001). Dispersion parameters of the random effects in the mean model can also be modeled with random effects (Noh, Lee and Pawitan, 2005). The response variable is allowed to follow a Gaussain, binomial, Poisson, or gamma distribution. The distribution of random effects can be specified as Gaussian, gamma, inverse-gamma or beta. It can handle complex structures such as crossed or nested designs in which various combinations of different distributions for random effects can be specified. Fixed effects in the mean can be estimated by maximizing the h-likelihood or a first-order Laplace approximation to the marginal likelihood. Dispersion parameters are estimated by using first-order adjusted profile likelihood, an extension of the restricted maximum likelihood; alternatively, these parameters can be assigned fixed values. The dhglm package also produces model-checking plots for various component of the model.

Details

Package:	dhglm
Type:	Package
Version:	1.0
Date:	2011-09-27
License:	Unlimited
LazyLoad:	yes

This is version 1.2 of the dhglm package.

Author(s)

Manegseok Noh, Youngjo Lee

Maintainer: Maengseok Noh <msnoh@pknu.ac.kr>

References

Lee, Y. and Nelder, J. A. (1996). Hierarchical generalised linear models (with discussion), Journal of the Royal Statistical Society B, 58, 619–678.

Lee, Y. and Nelder, J. A. (2001). Hierarchical generalised linear models : A synthesis of generalised linear models, random-effect model and structured dispersion, Biometrika, 88, 987–1006.

Lee, Y. and Nelder, J. A. (2006). Double hierarchical generalized linear models (with discussion), Applied Statistics 55, 139–185.

Lee, Y. Nelder, J. A. and Pawitan, Y. (2006). Generalised linear models with random effects: unified analysis via h-likelihood. Chapman & Hall: London.

Nelder, J. A. and Lee, Y. (1991). Generalised linear models for the analysis of Taguchi-type experiments, Applied Stochastic Models and Data Analysis, 7, 107–120.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalised linear models, Journal of the Royal Statistical Society A, 135, 370–384.

Noh, M., Lee, Y. and Pawitan, Y. (2005). Robust ascertainment-adjusted parameter estimation, Genetic Epidemiology, 29, 68–75.

See Also

<dhglmfit>

Examples

```
### DHGLM introducing random effects in the overdispersion for crack growth data
data(data_crack_growth)
model_mu<-DHGLMMODELING(Model="mean", Link="log", LinPred=y~crack0+(1|specimen),
RandDist="inverse-gamma")
model_phi<-DHGLMMODELING(Model="dispersion", Link="log",
LinPred=phi~cycle+(1|specimen), RandDist="gaussian")
res_crack<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)
```

---

data_crack_growth *Crack-growth Data*

Description

Hudak et al. (1978) presented data on crack growth data, which have been listed by Lu and Meeker (1993). There are 21 metallic specimens, each subjected to 120,000 loading cycles, with the crack lengths recorded every 10,000 cycles. We take t=no. cycles/100,000 here, so t(j)=j/1000 for j=1,...,12 coded by cycle. The crack increment sequences look rather irregular. Let l(i,j) be the crack length of the ith specimen at the jth observation and let y(i,j)=l(i,j)-l(i,j-1) be the corresponding increment of crack length, which has always a positive value. l(i,j-1) is coded by crack0 to use as a covariate.

Usage

```
data(data_crack_growth)
```

Format

A data frame with 241 observations of the following 4 variables.

y Increment in crack length, y(i,j)=l(i,j)-l(i,j-1), where l(i,j) is the crack length of the ith specimen at the jth observation.

`crack0` The previous crack length, l(i,j-1).

`specimen` The specimen number for 21 metallic specimens.

`cycle` The number of cycles/100,000; t(j)=j/100, j=1,...,12.

`phi` Inital phi values (=1) to fit the model with residual variances.

`lambda` Inital lambda values (=1) to fit the model with variances of random effects.

References

Hudak, S. J., Saxena, A. Bucci, R. J. and Malcom, R. C. (1978). Development of standard methods of testing and analyzing fatigue crack growth rate data. Technical report. AFML-TR-78-40. Westinghouse R & D Center, Wesinghouse Electric Corp., Pittsburgh, PA.

Lu, C. J. and Meeker, W. Q. (1993). Using degeneration measurements to estimate a time-to-failure distribution, Technometrics 35, 161–174.

`data_epilepsy` *Epilepsy Seizures Data*

Description

Thall and Vail (1990) presented longitudinal data from a clinical trial of 59 epileptics, who were randomized to a new drug or a placebo (T=0 or T=1). Baseline data were available at the start of the trial; the trial included the logarithm of the average number of epileptic seizures recorded in the 8-week period preceding the trial (B), the logarithm of age (A), and visit (V: a linear trend, coded (-3,-1,1,3)/10). A multivariate response variable (y) consists of the seizure counts during 2-week periods before each of four visits to the clinic.

Usage

```
data(data_epilepsy)
```

Format

A data frame with 236 observations of the following 7 variables.

`y` The seizure counts during 2-week periods before each of four visits to the clinic.

`T` Treatment(0=new drug, 1=placebo).

`B` The average number of epileptic seizures in the 8-week period preceding the trial.

`A` The logarithm of age (in years) of each patient.

`V` Linear trend coded (-3, -1, 1, 3)/10 for four visits of each patient.

`patient` Patient number for 59 epileptics.

`id` Observation number for 236 observations.

`phi` Inital phi values (=1) to fit the model with residual variances.

`lambda` Inital lambda values (=1) to fit the model with variances of random effects.

References

Thall, P. F. and Vail, S. C. (1990). Some covariance models for longitudinal count data with overdispersion, Biometrics 46, 657–671.

`dhglmfit` *Fitting Double Hierarchical Generalized Linear Models using h-likelihood Approach*

Description

dhglmfit is used to fit a class of double hierarchical generalized linear models (DHGLMs) in which fixed and random effects can be specified in both the mean and the dispersion components. A variety of distributions and link functions for both the response variables and the random effects are allowed. Fixed and random effects can also be fitted in both the mean and the dispersion components. To call the fitting function dhglmfit, models for the mean and dispersion must be specified by DHGLMMODELING object preferably created by calling the DHGLMMODELING function.

Usage

```
dhglmfit(RespDist="gaussian",BinomialDen=NULL, DataMain, MeanModel=,
DispersionModel=,PhiFix=NULL,LamFix=NULL,mord=1,dord=1,Maxiter=200,
convergence=1e-06)
```

Arguments

`RespDist`	The distribution of the response is set by the option RespDist. The user can set it to: "gaussian" (default), "binomial", "poisson", or "gamma".
`BinomialDen`	When RespDist="binomial", one should use the option BinomialDen to specify the denominator for the binomial distribution. This should be "NULL" (default) or a numeric vector of length equal to the length of DataMain. When specified as BinomialDen=NULL and RespDist="binomial", the denominator is 1.
`DataMain`	The option DataMain determines the data frame to be used (non-optional).
`MeanModel`	For the mean model, this option requries DGHLMMODELING object which should specified by the option Model="mean".
`DispersionModel`	For the overdispersion model, this option requries DGHLMMODELING object which should be specified by the option Model="dispersion".
`PhiFix, LamFix`	Two options that determine whether the overdispersion parameters (phi) and random-effect variance (lambda) are to be estimated or maintaned constant. Specifying defaults such as PhiFix =NULL (or LamFix =NULL) implies that phi (or lambda) is to be estimated. If not, phi (or lambda) is fixed at a value specified by PhiFix (or LamFix).
`mord`	The option mord specifies the order of Laplace approximation to the marginal likelihood for fitting the mean parameters. The choice is either 0 or 1 (default).

dord
: The option dord specifies the order of Laplace approximation to the adjusted profile likelihood for fitting the dispersion parameters. The choice is either 1 (default) or 2.

Maxiter
: Maxiter determines the maximum number of iterations for estimating all parameters. The default number is 200 iterations.

convergence
: Setting this option determines the criterion for convergence, which is computed as the absolute difference between the values of all the estimated parameters in the previous and current iterations. The default criterion is 1e-06.

Examples

```
#### Analysis of crack-growth data ####
data(data_crack_growth)
## GLM ##
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~crack0)
model_phi<-DHGLMMODELING(Model="dispersion")
res_glm<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)
## JGLM ##
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~crack0)
model_phi<-DHGLMMODELING(Model="dispersion", Link="log",
LinPred=phi~cycle)
res_jglm<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)
## HGLM I ##
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~crack0+(1|specimen),RandDist="inverse-gamma")
model_phi<-DHGLMMODELING(Model="dispersion")
res_hglm1<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)
## HGLM II ##
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~crack0+(1|specimen),RandDist="inverse-gamma")
model_phi<-DHGLMMODELING(Model="dispersion", Link="log",
LinPred=phi~cycle)
res_hglm2<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)
## DHGLM I ##
model_mu<-DHGLMMODELING(Model="mean", Link="log",
 LinPred=y~crack0+(1|specimen),RandDist="inverse-gamma")
model_phi<-DHGLMMODELING(Model="dispersion", Link="log",
LinPred=phi~cycle+(1|specimen),RandDist="gaussian")
res_dhglm1<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)
## DHGLM II ##
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~crack0+(1|specimen),RandDist="inverse-gamma",
LinkRandVariance="log",
LinPredRandVariance=lambda~1+(1|specimen),
RandDistRandVariance="gaussian")
```

```
model_phi<-DHGLMMODELING(Model="dispersion", Link="log",
LinPred=phi~cycle)
res_dhglm2<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)
## DHGLM III ##
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~crack0+(1|specimen),RandDist="inverse-gamma",
         LinkRandVariance="log",
LinPredRandVariance=lambda~1+(1|specimen),
RandDistRandVariance="gaussian")
model_phi<-DHGLMMODELING(Model="dispersion", Link="log",
LinPred=phi~cycle+(1|specimen),RandDist="gaussian")
res_dhglm3<-dhglmfit(RespDist="gamma",DataMain=data_crack_growth,
MeanModel=model_mu,DispersionModel=model_phi)


#### Analysis of epilepsy data ####
data(data_epilepsy)
## HGLM
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~B+T+A+T*B+V+(1|patient)+(1|id),
RandDist=c("gaussian","gamma"))
model_phi<-DHGLMMODELING(Model="dispersion")
res_hglm<-dhglmfit(RespDist="poisson",DataMain=data_epilepsy,
MeanModel=model_mu,DispersionModel=model_phi)
## DHGLM
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~B+T+A+T*B+V+(1|patient)+(1|id),
RandDist=c("gaussian","gaussian"),LinkRandVariance="log",
LinPredRandVariance=lambda~1+(1|id),
RandDistRandVariance="inverse-gamma")
model_phi<-DHGLMMODELING(Model="dispersion")
res_dhglm<-dhglmfit(RespDist="poisson",DataMain=data_epilepsy,
MeanModel=model_mu,DispersionModel=model_phi)
```

`DHGLMMODELING` *Defining the Fixed and Random Models for the Mean and Dispersion parameters in DHGLMs*

Description

DHGLMMODELING specifies a GLM, HGLM, DHGLM model for the mean parameters (mu), and a GLM, HGLM model for the overdispersion parameters (phi). GLM for mu, and GLM for phi are specified by adding only fixed terms to the linear predictors for the mu and phi, respectively. These are extended to HGLM for mu and HGLM for phi by adding some random terms. The DHGLM for mu is specified by allowing random effects for the variance of random effects in HGLM for mu. The LMatrix argument allows correlation structures to be defined for random terms. This is done by setting LMatrix to a matrix L that is used as a post-multiplier for the Z matrix of the corresponding random term.

Usage

```
DHGLMMODELING(Model="mean",Link=NULL,LinPred="constant",RandDist=NULL,
Offset=NULL,LMatrix=NULL,LinkRandVariance=NULL,LinPredRandVariance=NULL,
RandDistRandVariance="gaussian",
LinkRandVariance2=NULL,LinPredRandVariance2=NULL)
```

Arguments

`Model` This option specifies a GLM, HGLM or DHGLM model for mu when Model="mean" (default), and a GLM or HGLM for phi when Model="dispersion".

`Link` The link function for the linear predictor is specified by the option Link. For Model="mean", Link can be "identity", "logit", "probit", "cloglog", "log", or "inverse". For Model="dispersion", the choice is either "log" or "inverse". The default, specified as NULL link, is "identity" for Model="mean" and "log" for Model="dispersion".

`LinPred` The option LinPred specifies the fixed and random terms for the linear predictor for mu when specified as Model="mean" or for phi when Model="dispersion". For Model="mean", LinPred=y~x1+x2+(1|id1)+(1|id2) specifies y as the main response, x1 and x2 as fixed covariates and id1 and id2 as random terms. For Model="dispersion", the main response should be phi, e.g. phi~x1+x2+(1|id1)+(1|id2). This option can specify the model without random effects, e.g., LinPred=phi~x1+x2. The default is "constant", which is set to intercept only the model for the corresponding linear predictors.

`RandDist` The option RandDist specifies the distributions of the random terms represented in the option LinPred. It is set as a vector of distribution names from "gaussain" (default), "beta", "gamma", or "inverse-gamma" when Model="mean". For Model="dispersion", the choice is "gaussian" (default), "gamma", or "inverse-gamma". When more than one random terms are specified, e.g., `y~x1+x2+(1|id1)+(1|id2)` in the option LinPred, the different distributions for each random term can be specified, e.g., c("gaussian", "gamma"), which assumes normal distribution for the random term "id1" and gamma distribution for the random term "id2", respectively.

`Offset` The option Offset can be used to specify a known component to be included in the linear predictor specified by LinPred during fitting. This should be the default (NULL) or a numeric vector of length equal to that of the appropriate data.

`LMatrix` The option LMatrix sets a matrix that is used as a post-multiplier for the model matrix of the corresponding random effects. This option allows correlation structures to be defined for random effects. For example, when specified as Model="mean" and Lmatrix=L1+L2, the linear predictor for mu takes X beta + Z1 L1 r1 + Z2 L2 r2, where Z1 and Z2 are the model matrices for the random effects v1=L1 r1 and v2=L2 r2, specified in the option LinPred.

`LinkRandVariance`
The option LinkRandVariance specifies the link function for the linear predictor of the random-effect variances. The choice is either "log" (default) or "inverse". When more than two random terms are specified in the option LinPred, the user

can set different link functions, e.g., LinkRandVariance=c("log","inverse") for each random term.

`LinPredRandVariance`
: The option LinPredRandVariance specifies the fixed and random terms for the linear predictor of the random-effect variances for Model="mean". When y~x1+x2+(1|id1)+(1|id2) is specified in the option LinPred,

  LinPredRandVariance=c(lambda~xx1+(1|id11),lambda~xx2+(1|id12)) specifies xx1 and xx2 as fixed covariates and id11 and id12 as random terms in the lienar predictors for the variances of the random terms id1 and id2, respectively. For Model="dispersion", the random term is not allowed in the linear predictor of the random-effect variance. The default (NULL) is set to intercept only model for the corresponding linear predictors.

`RandDistRandVariance`
: The option RandDistRandVaraince specifies the distributions for the random terms in the LinPredRandVariance. The choice is "gaussian" (default), "gamma", or "inverse-gamma".

`LinkRandVariance2`
: This option specifies the link function for the linear predictor of the variance of random effects, which are specified in the option LinPredRandVariance. The choice is either "log" (default) or "inverse".

`LinPredRandVariance2`
: This option specifies the fixed terms for the linear predictor of the variance of random effects, which is specified in the option LinPredRandVariance. For example, when LinPredRandVariance=c(lambda~xx1+(1|id11),lambda~xx1+(1|id12)) is specified, LinPredRandVariance2=c(~xxx1,~xxx2) specifies xxx1 and xxx2 as fixed covariates for the linear predictor of random-effect variances for id11 and id12, respectively. The default (NULL) is set to constant variance for the random effects in LinPredRandVariance.

See Also

<`dhglmfit`>

`plotdhglm` *Produce Model-Checking Plots for a Double Hierarchical Generalized Linear Model Analysis*

Description

Plots residuals for the mean and dispersion models

Usage

```
plotdhglm(OUTPUT, type="mean", random=NULL)
```

Arguments

OUTPUT	The <dhglmfit> object to be plotted
type	Type of model required (mean, dispersion)
random	Random term whose residuals are to be plotted. Default (NULL) is the residuals from the full model

Details

Four types of plot are available (normal probability plot for residuals, histogram of residuals, residuals versus fitted values and absolute values of residuals versus fitted values).

See Also

<dhglmfit>

Examples

```
#### Model checking plot for crack-growth data
data(data_crack_growth)
model_mu<-DHGLMMODELING(Model="mean", Link="log",
LinPred=y~crack0+(1|specimen),RandDist="inverse-gamma")
model_phi<-DHGLMMODELING(Model="dispersion", Link="log",
LinPred=phi~cycle+(1|specimen),RandDist="gaussian")
res_crack_dhglm1<-dhglmfit(RespDist="gamma", DataMain=data_crack_growth,
MeanModel=model_mu, DispersionModel=model_phi)
plotdhglm(res_crack_dhglm1)
```

---

summary.dhglm *Summary Method for dhglmfit Object*

Description

It provides standard summary statistics for the fitted dhglmfit objects.

Usage

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

Arguments

object	A dhglmfit object.
...	other arguments

Details

When applied to a dhglmfit object, summary statistics are produced if the underlying model estimation converges. Otherwise, an error message is generated.

Value

It returns a list of summary.dhglm object containing the following statistics and tables.

- `FixCoef` — A two-dimensional matrix of fixed effect parameter estimates, respective standard errors, and t-values.
- `RandCoef` — A two-dimensional matrix of random effect estimates and respective standard errors.
- `iter` — Number of iterations used in dhglmfit estimation.
- `convergence` — A string, "converged" if the dhglmfit estimation converged, and "did not converge" otherwise.
- `mtwolikelihood` — Vectors of -2 times maximum and restricted likelihood.

Index