Package 'robmixglm'

September 27, 2024

Type Package

Title Robust Generalized Linear Models (GLM) using Mixtures

Version 1.2-4

Date 2024-09-26

Maintainer Ken Beath <ken@kjbeath.id.au>

Contact Ken Beath <ken@kjbeath.id.au>

Author Ken Beath [aut, cre]

Description Robust generalized linear models (GLM) using a mixture method, as described in Beath (2018) <doi:10.1080/02664763.2017.1414164>. This assumes that the data are a mixture of standard observations, being a generalised linear model, and outlier observations from an overdispersed generalized linear model. The overdispersed linear model is obtained by including a normally distributed random effect in the linear predictor of the generalized linear model.

Depends R(>= 3.2.0)

Suggests R.rsp, robustbase, lattice, forward

VignetteBuilder R.rsp

Imports fastGHQuad, stats, bbmle, VGAM, actuar, Rcpp (>= 0.12.15), methods, boot, numDeriv, parallel, doParallel, foreach, doRNG, MASS

LazyLoad yes

LazyData yes

NeedsCompilation yes

License GPL (>= 2)

LinkingTo Rcpp

Repository CRAN

Date/Publication 2024-09-27 11:30:02 UTC

Contents

`robmixglm-package` *Fits random effects meta-analysis models including robust models*

Description

Robust generalized linear models (GLM) using a mixture method, as described in Beath (2018) <doi:10.1080/02664763.2017.1414164>.

The robmixglm function

This is the main function that allows fitting the models. The robmixglm objects may be tested for outliers using outlierTest. The results of test.outliers may also be plotted.

Author(s)

Ken Beath <ken.beath@mq.edu.au>

References

Beath, K. J. A mixture-based approach to robust analysis of generalised linear models, Journal of Applied Statistics, 45(12), 2256-2268 (2018) DOI: 10.1080/02664763.2017.1414164

Examples

```
# for the following cores is set to 1 to satisfy the CRAN testing requirements
# removing will reduce the time taken depending on number of cores available
# animal brain vs body weight
library(MASS)
data(Animals)
Animals$logbrain <- log(Animals$brain)
Animals$logbody <- log(Animals$body)
lm1 <- lm(logbrain~logbody, data = Animals)
lm2 <- robmixglm(logbrain~logbody, data = Animals, cores = 1)
plot(Animals$logbody, Animals$logbrain)
abline(lm1, col = "red")
abline(lm2, col = "green")
plot(outlierProbs(lm2))
outlierTest(lm2, cores  =  1)

# Forbes data on relationship between atmospheric pressure and boiling point of water
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(100*log10(pres)~bp, data = MASS::forbes, cores = 1)
summary(forbes.robustmix)
plot(outlierProbs(forbes.robustmix))
outlierTest(forbes.robustmix, cores  =  1)

# diabetes
diabdata.robustmix <- robmixglm(glyhb~age+gender+bmi+waisthip+frame,
    data = diabdata, cores = 1)
summary(diabdata.robustmix)
# this will take about 5-10 minutes
diabdata.step <- step(diabdata.robustmix, glyhb~age+gender+bmi+waisthip+frame)
summary(diabdata.step)
plot(outlierProbs(diabdata.step))
outlierTest(diabdata.step, cores  =  1)

# Hawkins' data
library(forward)
data(hawkins)
hawkins.robustmix <- robmixglm(y~x1+x2+x3+x4+x5+x6+x7+x8,
    cores = 1, data=hawkins)
summary(hawkins.robustmix)
plot(outlierProbs(hawkins.robustmix))
outlierTest(hawkins.robustmix, cores = 1)

# carrot damage
library(robustbase)
data(carrots)
carrots.robustmix <- robmixglm(cbind(success, total-success)~logdose+factor(block),
     family = "binomial", data = carrots, cores = 1)
summary(carrots.robustmix)
plot(outlierProbs(carrots.robustmix))
outlierTest(carrots.robustmix, cores  =  1)
```

```
# train derailment
library(forward)
data(derailme)
derailme$cYear <- derailme$Year-mean(derailme$Year)
derailme$TrainKm100 <- derailme$TrainKm*100.0
derailme.robustmix <- robmixglm(y~cYear+factor(Type), offset = log(TrainKm100),
    family = "truncpoisson", quadpoints = 51,  data = derailme, cores = 1)
summary(derailme.robustmix)
plot(outlierProbs(derailme.robustmix))
outlierTest(derailme.robustmix, cores  =  1)

# hospital costs
hospcosts.robustmix <- robmixglm(costs~adm+age+dest+ins+loglos+sex, family = "gamma",
    data = hospcosts, cores = 1)
summary(hospcosts.robustmix)
plot(outlierProbs(hospcosts.robustmix))
outlierTest(hospcosts.robustmix, cores  =  1)
```

`AIC` *AIC for robmixglm object*

Description

Returns AIC for a robmixglm object.

Usage

```
## S3 method for class 'robmixglm'
AIC(object,  ...,  k  =  2)
```

Arguments

`object`	robmixglm object
`...`	additional argument; currently none is used.
`k`	penalty per parameter

Value

AIC

Author(s)

Ken Beath

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
AIC(forbes.robustmix)
```

`BIC` *BIC for robmixglm object*

Description

Returns BIC for a robmixglm object.

Usage

```
## S3 method for class 'robmixglm'
BIC(object,  ...)
```

Arguments

`object`	robmixglm object
`...`	additional argument; currently none is used.

Value

BIC

Author(s)

Ken Beath

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
BIC(forbes.robustmix)
```

coef *Coefficients for a robmixglm object*

Description

Returns coefficients for a robmixglm object. Only the coefficients for the linear part of the model are returned. Additional coefficients may be obtained using summary().

Usage

```
## S3 method for class 'robmixglm'
coef(object,  ...)
```

Arguments

object	robmixglm object
...	additional argument; currently none is used.

Value

coef

Author(s)

Ken Beath

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
coef(forbes.robustmix)
```

diabdata *Diabetes data*

Description

Data from Heritier et al (2009), originally from Harrell (2001, p379). This data was from a study of the prevalence of cardiovascular risk factors such as obesity and diabetes for African Americans. (Willems et al, 19997) Data was available for 403 subjects screened for diabetes, reduced to 372 after removal of cases with missing data.

Usage

```
diabdata
```

Format

A data frame with 372 observations on the following 8 variables.

`glyhb` Glycosated haemoglobin (values above 7.0 are usually taken as a positive diagnosis of diabetes)

`age` age in years

`gender` male or female

`bmi` body mass index in kg/m^2

`waisthip` ratio of waist to hip measurement

`frame` body frame, small, medium or large

`stab.glu` glucose

`location` location, Buckingham or Louisa

Source

Heritier et al (2009)

References

Harrell, F.E. (2001). Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression and Survival Analysis. Springer.

Heritier, S., Cantoni, E., Copt, S. and Victoria-Feser, M-P (2009). Robust Methods in Biostatistics. Wiley.

Willems, J.P., Saunders, J.T., Hunt, D.E. and Schorling, J.B. (1997) Prevalence of coronary heart disease risk factors among rural blacks: A community-based study. Southern Medical Journal, 90:814-820.

Examples

```
diabdata.robustmix <- robmixglm(glyhb~age+gender+bmi+waisthip+frame+location,
         data = diabdata, cores = 1)
summary(diabdata.robustmix)

diabdata.step <- step(diabdata.robustmix, glyhb~age+gender+bmi+waisthip+frame+location, cores = 1)
summary(diabdata.step)
```

`extractAIC` *Extract AIC from a Fitted Model*

Description

Computes the (generalized) AIC for a fitted `robmixglm` model. Used in `step`, otherwise use `AIC`.

Usage

```
## S3 method for class 'robmixglm'
extractAIC(fit, scale, k = 2, ...)
```

Arguments

fit	fitted robmixglm model.
scale	ignored.
k	numeric specifying the 'weight' of the *equivalent degrees of freedom* (≡ edf) part in the AIC formula.
...	further arguments (currently unused).

Author(s)

Ken Beath

See Also

extractAIC, step

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = MASS::forbes, cores = 1)
extractAIC(forbes.robustmix)
```

---

fitted.robmixglm *Fitted values.*

Description

Calculates the fitted values.

Usage

```
## S3 method for class 'robmixglm'
fitted(object, ...)
```

Arguments

object	A robmixglm object with a mixture (robust) random effects distribution.
...	Other parameters. (not used)

Value

A vector of the fitted values.

Author(s)

Ken Beath <ken.beath@mq.edu.au>

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
BIC(forbes.robustmix)
plot(fitted(forbes.robustmix), residuals(forbes.robustmix))
```

---

hospcosts *Hospital Costs data*

Description

Data for the analysis in Beath (2018), previously analysed in Marazzi and Yohai (2004), Cantoni and Ronchetti (2006) and Heritier et al (2009). The data is for 100 patients hospitalised at the Centre Hospitalier Universitaire Vaudois in Lausanne, Switzerland for "medical back problems" (APDRG 243).

Usage

```
hospcosts
```

Format

A data frame with 100 observations on the following 9 variables.

`id` patient id

`costs` cost of stay in Swiss francs

`los` length of stay in days

`adm` admission type, 0 = planned, 1 = emergency

`ins` insurance type, 0 = regular, 1 = private

`age` age in years

`sex` sex, 0 = female, 1 = male

`dest` discharge destination, 0 = another health institution, 1 = home

`loglos` log of length of stay

Source

Heritier et al (2009)

References

Cantoni, E., & Ronchetti, E. (2006). A robust approach for skewed and heavy-tailed outcomes in the analysis of health care expenditures. Journal of Health Economics, 25(2), 198213. http://doi.org/10.1016/j.jhealeco.2005.04.0

Heritier, S., Cantoni, E., Copt, S. and Victoria-Feser, M-P (2009). Robust Methods in Biostatistics. Wiley.

Marazzi, A., & Yohai, V. J. (2004). Adaptively truncated maximum likelihood regression with asymmetric errors. Journal of Statistical Planning and Inference, 122(12), 271291. http://doi.org/10.1016/j.jspi.2003.06.011

Examples

```
hospcosts.robustmix <- robmixglm(costs~adm+age+dest+ins+loglos+sex, family = "gamma",
    data = hospcosts, cores = 1)
summary(hospcosts.robustmix)
```

---

logLik *log Likelikelihood for robmixglm object*

Description

Returns log Likelihood for a robmixglm object.

Usage

```
## S3 method for class 'robmixglm'
logLik(object,  ...)
```

Arguments

`object`	robmixglm object
`...`	additional argument; currently none is used.

Value

The loglikelihood.

Author(s)

Ken Beath

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
logLik(forbes.robustmix)
```

outlierProbs *Calculate outlier probabilities for each observation.*

Description

For the normal mixture random effect calculates the probability that each observation is an outlier based on the posterior probability of it being an outlier.

Usage

```
outlierProbs(object)
```

Arguments

object	A metaplus object with a mixture (robust) random effects distribution.

Details

The outlier probabilities are obtained as the posterior probabilities of each observation being an outlier based on the fitted mixture model.

Value

outlier.prob	Posterior probability that each observation is an outlier

Author(s)

Ken Beath <ken.beath@mq.edu.au>

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
outlierProbs(forbes.robustmix)
```

outlierTest *Test for the presence of outliers.*

Description

Uses the parametric bootstrap to test for the presence of outliers.

Usage

```
outlierTest(object, R = 999, cores = max(detectCores() %/% 2, 1))
```

Arguments

object	A robmixglm object with a mixture (robust) random effects distribution.
R	number of bootstrap replications
cores	Number of cores to be used in parallel. Default is one less than available.

Details

Performs a parametric bootstrap to compare models with and without outliers.

Value

An outlierTest object which is the object of class "boot" returned by the call to boot.

Author(s)

Ken Beath <ken.beath@mq.edu.au>

Examples

```
hospcosts.robustmix <- robmixglm(costs~adm+age+dest+ins+loglos+sex, family = "gamma",
    data = hospcosts, cores = 1)
summary(hospcosts.robustmix)
summary(outlierTest(hospcosts.robustmix,  cores  =  1))
```

---

plot.outlierProbs *Plot outlier probabilities.*

Description

Plots the outlier probability for each observation, from an outlierProbs object.

Usage

```
## S3 method for class 'outlierProbs'
plot(x,  ...)
```

Arguments

x	outlierProbs object to be plotted
...	additional parameters to plot

Value

Plot

Author(s)

Ken Beath <ken.beath@mq.edu.au>

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
plot(outlierProbs(forbes.robustmix))
```

`predict.robmixglm` *Predict Method for robmixglm*

Description

Obtains predictions from a fitted robust mixture generalized linear model object.

Usage

```
## S3 method for class 'robmixglm'
predict(object, newdata = NULL,
            type = c("link", "response"), ...)
```

Arguments

`object`	a fitted object of class inheriting from `robmixglm`.
`newdata`	optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
`type`	the type of prediction required. The default `link` is on the scale of the linear predictors, while the alternative `response` is on the scale of the response variable.
`...`	Other parameters. (not used)

Details

If `newdata` is omitted the predictions are based on the data used for the fit. In that case how cases with missing values in the original fit is determined by the `na.action` argument of that fit. If `na.action = na.omit` omitted cases will not appear in the residuals, whereas if `na.action = na.exclude` they will appear (in predictions and standard errors), with residual value `NA`. See also `napredict`.

Value

A vector predicted linear predictors or response. For `binomial` the resonse is the predicted proportion.

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(100*log10(pres)~bp, data = forbes, cores = 1)
plot(forbes$bp, forbes$pres)
preddata <- data.frame(bp = seq(from = min(forbes$bp), to = max(forbes$bp), by = 0.01))
# convert to original scale
preddata$predpres <-10^(predict(forbes.robustmix, newdata = preddata)/100)
lines(preddata$bp, preddata$predpres, col = "red")
```

`print.outlierTest` *Print an outlierTest object*

Description

Print an outlierTest object.

Usage

```
## S3 method for class 'outlierTest'
print(x,  ...)
```

Arguments

`x`	outlierTest object
`...`	further arguments (not currently used)

Author(s)

Ken Beath

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
summary(forbes.robustmix)
print(outlierTest(forbes.robustmix,  cores  =  1))
```

residuals.robmixglm *Extract Model Residuals*

Description

Extracts model residuals from objects returned by modeling functions.

Usage

```
## S3 method for class 'robmixglm'
residuals(object, type = c("deviance", "pearson"), ...)
```

Arguments

object	an object for which the extraction of model residuals is meaningful.
type	Type of residual where valid types are deviance and pearson.
...	other arguments.

Value

Residuals extracted from the object object.

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
BIC(forbes.robustmix)
plot(fitted(forbes.robustmix), residuals(forbes.robustmix))
```

robmixglm *Fits a Robust Generalized Linear Model and Variants*

Description

Fits robust generalized linear models and variants described in Beath (2018).

Usage

```
robmixglm(formula, family = c("gaussian", "binomial", "poisson",
"gamma", "truncpoisson", "nbinom"), data, offset = NULL,
quadpoints = 21, notrials = 50, EMTol = 1.0e-4, cores = max(detectCores() %/% 2, 1),
verbose = FALSE)
```

Arguments

formula	Model formula
family	Distribution of response
data	Data frame from which variables are obtained
offset	Offset to be incorporated in the linear predictor.
quadpoints	Number of quadrature points used in the Gauss-Hermite integration.
notrials	Number of random starting values to be used for EM
EMTol	Relative change in likelihood for completion of EM algorithm before switching to quasi-Newton
cores	Number of cores to be used for parallel evaluation of starting values
verbose	Print out diagnostic information? This includes the likelihood and parameter estimates for each EM run.

Details

Fits robust generalized models assuming that data is a mixture of standard observations and outlier abservations, which belong to an overdispersed model (Beath, 2018). For binomial, Poisson, truncated Poisson and gamma, the overdispersed component achieved through including a random effect as part of the linear predictor, as described by Aitkin (1996). For gaussian and negative binomial data the outlier component is also a gaussian and negative binomial model, respectively but with a higher dispersion. For gaussian this corresponds to a higher value of σ^2 but for negative binomial this is a lower value of θ.

The method used is a generalised EM. Random starting values are determined by randomly allocating observations to either the standard or outlier class for the first iteration of the EM. The EM is then run to completion for all sets of starting values. The best set of starting values is then used to obtain the final results using a quasi-Newton method. Where the overdispersed data is obtained using a random effect, the likelihood is obtained by integrating out the random effect using Gauss-Hermite quadrature.

Value

robmixglm object. This contains

fit	Final model fit from quasi-Newton
prop	Posterior probability of observation in each class
logLik	final log likelihood
np	Number of parameters
nobs	Number of observations
coef.names	Coefficient names
call	Call to function
family	Family of model to be fitted
model	model
terms	terms

xlevels	Levels for factors.
quadpoints	Number of quadrature points used in the Gauss-Hermite integration.
notrials	Number of random starting values to be used for EM
EMTol	Relative change in likelihood for completion of EM algorithm before switching to quasi-Newton
verbose	Was verbose output requested?

Author(s)

Ken Beath

References

Beath, K. J. A mixture-based approach to robust analysis of generalised linear models, Journal of Applied Statistics, 45(12), 2256-2268 (2018) DOI: 10.1080/02664763.2017.1414164

Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models. Statistics and Computing, 6, 251262. DOI: 10.1007/BF00140869

Examples

```
if (requireNamespace("MASS", quietly = TRUE)) {
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(100*log10(pres)~bp, data = forbes, cores = 1)
}
```

---

summary.robmixglm *summaryficients for robmixglm object*

Description

Returns summary for a robmixglm object.

Usage

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

Arguments

object	robmixglm object
...	additional argument; currently none is used.

Value

summary

Author(s)

Ken Beath

Examples

```
library(MASS)
data(forbes)
forbes.robustmix <- robmixglm(bp~pres, data = forbes, cores = 1)
summary(forbes.robustmix)
```

Index