Package 'fastglm'

October 13, 2022

Type Package

Title Fast and Stable Fitting of Generalized Linear Models using 'RcppEigen'

Version 0.0.3

Maintainer Jared Huling <jaredhuling@gmail.com>

Description Fits generalized linear models efficiently using 'RcppEigen'. The iteratively reweighted least squares implementation utilizes the step-halving approach of Marschner (2011) <doi:10.32614/RJ-2011-012> to help safeguard against convergence issues.

BugReports https://github.com/jaredhuling/fastglm/issues

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp (>= 0.12.13), methods

Depends bigmemory

LinkingTo Rcpp, RcppEigen, BH, bigmemory

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, glm2

VignetteBuilder knitr

NeedsCompilation yes

Author Jared Huling [aut, cre],
Douglas Bates [cph],
Dirk Eddelbuettel [cph],
Romain Francois [cph],
Yixuan Qiu [cph]

Repository CRAN

Date/Publication 2022-05-23 16:50:02 UTC

R topics documented:

deviance.fastglm . . . 2
family.fastglm . . . 3
fastglm . . . 3
fastglmPure . . . 5
logLik.fastglm . . . 8
predict.fastglm . . . 8
print.fastglm . . . 9
residuals.fastglm . . . 9
summary.fastglm . . . 10
%*%,big.matrix,vector-method . . . 11

Index **12**

`deviance.fastglm` *deviance method for fastglm fitted objects*

Description

deviance method for fastglm fitted objects

Usage

```
## S3 method for class 'fastglm'
deviance(object, ...)
```

Arguments

`object`	fastglm fitted object
`...`	not used

Value

The value of the deviance extracted from the object

family.fastglm *family method for fastglm fitted objects*

Description

family method for fastglm fitted objects

Usage

```
## S3 method for class 'fastglm'
family(object, ...)
```

Arguments

object	fastglm fitted object
...	not used

Value

returns the family of the fitted object

fastglm *fast generalized linear model fitting*

Description

fast generalized linear model fitting

bigLm default

Usage

```
fastglm(x, ...)

## Default S3 method:
fastglm(
  x,
  y,
  family = gaussian(),
  weights = NULL,
  offset = NULL,
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  method = 0L,
  tol = 1e-08,
```

```
  maxit = 100L,
  ...
)
```

Arguments

x	input model matrix. Must be a matrix object
...	not used
y	numeric response vector of length nobs.
family	a description of the error distribution and link function to be used in the model. For fastglm this can be a character string naming a family function, a family function or the result of a call to a family function. For fastglmPure only the third option is supported. (See family for details of family functions.)
weights	an optional vector of 'prior weights' to be used in the fitting process. Should be a numeric vector.
offset	this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be a numeric vector of length equal to the number of cases
start	starting values for the parameters in the linear predictor.
etastart	starting values for the linear predictor.
mustart	values for the vector of means.
method	an integer scalar with value 0 for the column-pivoted QR decomposition, 1 for the unpivoted QR decomposition, 2 for the LLT Cholesky, or 3 for the LDLT Cholesky
tol	threshold tolerance for convergence. Should be a positive real number
maxit	maximum number of IRLS iterations. Should be an integer

Value

A list with the elements

coefficients	a vector of coefficients
se	a vector of the standard errors of the coefficient estimates
rank	a scalar denoting the computed rank of the model matrix
df.residual	a scalar denoting the degrees of freedom in the model
residuals	the vector of residuals
s	a numeric scalar - the root mean square for residuals
fitted.values	the vector of fitted values

Examples

```
x <- matrix(rnorm(10000 * 100), ncol = 100)
y <- 1 * (0.25 * x[,1] - 0.25 * x[,3] > rnorm(10000))

system.time(gl1 <- glm.fit(x, y, family = binomial()))

system.time(gf1 <- fastglm(x, y, family = binomial()))

system.time(gf2 <- fastglm(x, y, family = binomial(), method = 1))

system.time(gf3 <- fastglm(x, y, family = binomial(), method = 2))

system.time(gf4 <- fastglm(x, y, family = binomial(), method = 3))

max(abs(coef(gl1) - gf1$coef))
max(abs(coef(gl1) - gf2$coef))
max(abs(coef(gl1) - gf3$coef))
max(abs(coef(gl1) - gf4$coef))


## Not run:
nrows <- 50000
ncols <- 50
bkFile <- "bigmat2.bk"
descFile <- "bigmatk2.desc"
bigmat <- filebacked.big.matrix(nrow=nrows, ncol=ncols, type="double",
                                backingfile=bkFile, backingpath=".",
                                descriptorfile=descFile,
                                dimnames=c(NULL,NULL))
for (i in 1:ncols) bigmat[,i] = rnorm(nrows)*i
y <- 1*(rnorm(nrows) + bigmat[,1] > 0)

system.time(gfb1 <- fastglm(bigmat, y, family = binomial(), method = 3))

## End(Not run)
```

`fastglmPure` *fast generalized linear model fitting*

Description

fast generalized linear model fitting

Usage

```
fastglmPure(
  x,
```

```
  y,
  family = gaussian(),
  weights = rep(1, NROW(y)),
  offset = rep(0, NROW(y)),
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  method = 0L,
  tol = 1e-07,
  maxit = 100L
)
```

Arguments

x	input model matrix. Must be a matrix object
y	numeric response vector of length nobs.
family	a description of the error distribution and link function to be used in the model. For fastglmPure this can only be the result of a call to a family function. (See family for details of family functions.)
weights	an optional vector of 'prior weights' to be used in the fitting process. Should be a numeric vector.
offset	this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be a numeric vector of length equal to the number of cases
start	starting values for the parameters in the linear predictor.
etastart	starting values for the linear predictor.
mustart	values for the vector of means.
method	an integer scalar with value 0 for the column-pivoted QR decomposition, 1 for the unpivoted QR decomposition, 2 for the LLT Cholesky, 3 for the LDLT Cholesky, 4 for the full pivoted QR decomposition, 5 for the Bidiagonal Divide and Conquer SVD
tol	threshold tolerance for convergence. Should be a positive real number
maxit	maximum number of IRLS iterations. Should be an integer

Value

A list with the elements

coefficients	a vector of coefficients
se	a vector of the standard errors of the coefficient estimates
rank	a scalar denoting the computed rank of the model matrix
df.residual	a scalar denoting the degrees of freedom in the model
residuals	the vector of residuals
s	a numeric scalar - the root mean square for residuals
fitted.values	the vector of fitted values

Examples

```
set.seed(1)
x <- matrix(rnorm(1000 * 25), ncol = 25)
eta <- 0.1 + 0.25 * x[,1] - 0.25 * x[,3] + 0.75 * x[,5] -0.35 * x[,6] #0.25 * x[,1] - 0.25 * x[,3]
y <- 1 * (eta > rnorm(1000))

yp <- rpois(1000, eta ^ 2)
yg <- rgamma(1000, exp(eta) * 1.75, 1.75)

# binomial
system.time(gl1 <- glm.fit(x, y, family = binomial()))

system.time(gf1 <- fastglmPure(x, y, family = binomial(), tol = 1e-8))

system.time(gf2 <- fastglmPure(x, y, family = binomial(), method = 1, tol = 1e-8))

system.time(gf3 <- fastglmPure(x, y, family = binomial(), method = 2, tol = 1e-8))

system.time(gf4 <- fastglmPure(x, y, family = binomial(), method = 3, tol = 1e-8))

max(abs(coef(gl1) - gf1$coef))
max(abs(coef(gl1) - gf2$coef))
max(abs(coef(gl1) - gf3$coef))
max(abs(coef(gl1) - gf4$coef))

# poisson
system.time(gl1 <- glm.fit(x, yp, family = poisson(link = "log")))

system.time(gf1 <- fastglmPure(x, yp, family = poisson(link = "log"), tol = 1e-8))

system.time(gf2 <- fastglmPure(x, yp, family = poisson(link = "log"), method = 1, tol = 1e-8))

system.time(gf3 <- fastglmPure(x, yp, family = poisson(link = "log"), method = 2, tol = 1e-8))

system.time(gf4 <- fastglmPure(x, yp, family = poisson(link = "log"), method = 3, tol = 1e-8))

max(abs(coef(gl1) - gf1$coef))
max(abs(coef(gl1) - gf2$coef))
max(abs(coef(gl1) - gf3$coef))
max(abs(coef(gl1) - gf4$coef))

# gamma
system.time(gl1 <- glm.fit(x, yg, family = Gamma(link = "log")))

system.time(gf1 <- fastglmPure(x, yg, family = Gamma(link = "log"), tol = 1e-8))

system.time(gf2 <- fastglmPure(x, yg, family = Gamma(link = "log"), method = 1, tol = 1e-8))

system.time(gf3 <- fastglmPure(x, yg, family = Gamma(link = "log"), method = 2, tol = 1e-8))

system.time(gf4 <- fastglmPure(x, yg, family = Gamma(link = "log"), method = 3, tol = 1e-8))
```

```
max(abs(coef(gl1) - gf1$coef))
max(abs(coef(gl1) - gf2$coef))
max(abs(coef(gl1) - gf3$coef))
max(abs(coef(gl1) - gf4$coef))
```

`logLik.fastglm` *logLik method for fastglm fitted objects*

Description

logLik method for fastglm fitted objects

Usage

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

Arguments

`object`	fastglm fitted object
`...`	not used

Value

Returns an object of class `logLik`

`predict.fastglm` *Obtains predictions and optionally estimates standard errors of those predictions from a fitted generalized linear model object.*

Description

Obtains predictions and optionally estimates standard errors of those predictions from a fitted generalized linear model object.

Usage

```
## S3 method for class 'fastglm'
predict(
  object,
  newdata = NULL,
  type = c("link", "response"),
  se.fit = FALSE,
  dispersion = NULL,
  ...
)
```

Arguments

object	a fitted object of class inheriting from "fastglm".
newdata	a matrix to be used for prediction
type	the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities. The "terms" option returns a matrix giving the fitted values of each term in the model formula on the linear predictor scale. The value of this argument can be abbreviated.
se.fit	logical switch indicating if standard errors are required.
dispersion	the dispersion of the GLM fit to be assumed in computing the standard errors. If omitted, that returned by summary applied to the object is used.
...	further arguments passed to or from other methods.

---

print.fastglm *print method for fastglm objects*

Description

print method for fastglm objects

Usage

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

Arguments

x	object to print
...	not used

---

residuals.fastglm *residuals method for fastglm fitted objects*

Description

residuals method for fastglm fitted objects

Usage

```
## S3 method for class 'fastglm'
residuals(
  object,
  type = c("deviance", "pearson", "working", "response", "partial"),
  ...
)
```

Arguments

object	fastglm fitted object
type	type of residual to be returned
...	not used

Value

a vector of residuals

---

summary.fastglm *summary method for fastglm fitted objects*

Description

summary method for fastglm fitted objects

Usage

```
## S3 method for class 'fastglm'
summary(object, dispersion = NULL, ...)
```

Arguments

object	fastglm fitted object
dispersion	the dispersion parameter for the family used. Either a single numerical value or NULL (the default), when it is inferred from object.
...	not used

Value

a summary.fastglm object

Examples

```
x <- matrix(rnorm(10000 * 10), ncol = 10)
y <- 1 * (0.25 * x[,1] - 0.25 * x[,3] > rnorm(10000))

fit <- fastglm(x, y, family = binomial())

summary(fit)
```

%*%,big.matrix,vector-method
big.matrix prod

Description

big.matrix prod

big.matrix prod

Usage

```
## S4 method for signature 'big.matrix,vector'
x %*% y

## S4 method for signature 'vector,big.matrix'
x %*% y
```

Arguments

x	big.matrix
y	numeric vector

Index

%*%,vector,big.matrix-method
        (%*%,big.matrix,vector-method), 11
%*%,big.matrix,vector-method, 11

deviance.fastglm, 2

family, *4*, *6*
family.fastglm, 3
fastglm, 3
fastglmPure, 5

logLik.fastglm, 8

predict.fastglm, 8
print.fastglm, 9

residuals.fastglm, 9

summary.fastglm, 10