

Package ‘bigFastlm’

March 12, 2017

Type Package

Title Fast Linear Models for Objects from the 'bigmemory' Package

Version 0.0.2

Date 2017-03-11

Maintainer Jared Huling <jaredhuling@gmail.com>

Description A reimplementation of the fastLm() functionality of 'RcppEigen' for big.matrix objects for fast out-of-memory linear model fitting.

License GPL (>= 2)

LazyData TRUE

Depends R (>= 3.2.0), bigmemory

Imports Rcpp (>= 0.12.3), methods

LinkingTo Rcpp, RcppEigen, BH, bigmemory

RoxygenNote 5.0.1

NeedsCompilation yes

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

Repository CRAN

Date/Publication 2017-03-12 18:53:34

R topics documented:

bigLm	2
bigLmPure	3
predict.bigLm	4
print.bigLm	5
summary.bigLm	6
%*%,big.matrix,vector-method	7

Index	8
--------------	----------

bigLm

*fast and memory efficient linear model fitting***Description**

fast and memory efficient linear model fitting

bigLm default

Usage

bigLm(X, ...)

Default S3 method:

bigLm(X, y, method = 0L, gigs = 2, nslices = NULL, ...)

Arguments

X	input model matrix. must be a big.matrix object (type = 8 for double)
...	not used
y	numeric response vector of length nobs.
method	an integer scalar with value 0 for the LLT Cholesky or 1 for the LDLT Cholesky
gigs	double scalar. maximum number of gigs of memory available. Used to figure out how to break up calculations involving the design matrix X
nslices	integer scalar, defaults to NULL, which defers to the gigs argument to determine the number of slices required. If specified, nslices determines the number of slices to break up computation of X'X into.

Value

A list object with S3 class "bigLm" 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

```

library(bigmemory)

nrows <- 50000
ncols <- 50
bkFile <- "bigmat.bk"
descFile <- "bigmatk.desc"
bigmat <- filebacked.big.matrix(nrow=nrows, ncol=ncols, type="double",
                               backingfile=bkFile, backingpath=".",
                               descriptorfile=descFile,
                               dimnames=c(NULL, NULL))

# Each column value will be the column number multiplied by
# samples from a standard normal distribution.
set.seed(123)
for (i in 1:ncols) bigmat[,i] = rnorm(nrows)*i

y <- rnorm(nrows) + bigmat[,1]

system.time(lmr1 <- bigLm(bigmat, y))

system.time(lmr2 <- lm.fit(x = bigmat[,], y = y))

max(abs(coef(lmr1) - coef(lmr2)))

```

bigLmPure

fast and memory efficient linear model fitting

Description

fast and memory efficient linear model fitting

Usage

```
bigLmPure(X, y, method = 0L, gigs = 2, nslices = NULL)
```

Arguments

X	input model matrix. must be a big.matrix object (type = 8 for double)
y	numeric response vector of length nobs.
method	an integer scalar with value 0 for the LLT Cholesky or 1 for the LDLT Cholesky
gigs	double scalar. maximum number of gigs of memory available. Used to figure out how to break up calculations involving the design matrix X
nslices	integer scalar, defaults to NULL, which defers to the gigs argument to determine the number of slices required. If specified, nslices determines the number of slices to break up computation of $X'X$ into.

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

```
library(bigmemory)

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))

# Each column value will be the column number multiplied by
# samples from a standard normal distribution.
set.seed(123)
for (i in 1:ncols) bigmat[,i] = rnorm(nrows)*i

y <- rnorm(nrows) + bigmat[,1]

system.time(lmr1 <- bigLmPure(bigmat, y))

system.time(lmr2 <- lm.fit(x = bigmat[,], y = y))

max(abs(coef(lmr1) - coef(lmr2)))
```

predict.bigLm

Prediction method for bigLm fitted objects

Description

Prediction method for bigLm fitted objects

Usage

```
## S3 method for class 'bigLm'
predict(object, newdata = NULL, ...)
```

Arguments

object	fitted "bigLm" model object
newdata	big.matrix object. If NULL, then fitted values are returned
...	not used

Value

An object depending on the type argument

Examples

```
library(bigmemory)

nrows <- 50000
ncols <- 50
bkFile <- "bigmat3.bk"
descFile <- "bigmatk3.desc"
bigmat <- filebacked.big.matrix(nrow=nrows, ncol=ncols, type="double",
                               backingfile=bkFile, backingpath=".",
                               descriptorfile=descFile,
                               dimnames=c(NULL, NULL))

# Each column value will be the column number multiplied by
# samples from a standard normal distribution.
set.seed(123)
for (i in 1:ncols) bigmat[,i] = rnorm(nrows)*i

y <- rnorm(nrows) + bigmat[,1]

system.time(lmr1 <- bigLm(bigmat, y))

preds <- predict(lmr1, newdata = bigmat)
```

print.bigLm

print method for bigLm objects

Description

print method for bigLm objects
print method for summary.bigLm objects

Usage

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

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

Arguments

```
x          a "summary.bigLm" object
...        not used
```

```
summary.bigLm      summary method for bigLm fitted objects
```

Description

summary method for bigLm fitted objects

Usage

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

Arguments

```
object      bigLm fitted object
...         not used
```

Value

a summary.bigLm object

Examples

```
library(bigmemory)

nrows <- 50000
ncols <- 15
bkFile <- "bigmat4.bk"
descFile <- "bigmatk4.desc"
bigmat <- filebacked.big.matrix(nrow=nrows, ncol=ncols, type="double",
                               backingfile=bkFile, backingpath=".",
                               descriptorfile=descFile,
                               dimnames=c(NULL, NULL))

# Each column value will be the column number multiplied by
```

```
# samples from a standard normal distribution.
set.seed(123)
for (i in 1:ncols) bigmat[,i] = rnorm(nrows)*i

y <- rnorm(nrows) + bigmat[,1] - bigmat[,2]

system.time(lmr1 <- bigLm(bigmat, y))

summary(lmr1)
```

%*%,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),
 7
%%, big.matrix, vector-method, 7

bigLm, 2
bigLmPure, 3

predict.bigLm, 4
print.bigLm, 5
print.summary.bigLm (print.bigLm), 5

summary.bigLm, 6