Package 'bigstep'

July 22, 2025

Type Package

Title Stepwise Selection for Large Data Sets

Version 1.1.2

Date 2025-2-10

Description Selecting linear and generalized linear models for large data sets using modified stepwise procedure and modern selection criteria (like modifications of Bayesian Information Criterion). Selection can be performed on data which exceed RAM capacity. Bogdan et al., (2004) <doi:10.1534/genetics.103.021683>.

License GPL-3

URL https://github.com/pmszulc/bigstep

BugReports https://github.com/pmszulc/bigstep/issues

Depends R (>= 3.5.0)

Imports bigmemory, magrittr, matrixStats, R.utils, RcppEigen, speedglm, stats, utils

Suggests devtools, knitr, rmarkdown, testthat

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

Author Piotr Szulc [aut, cre]

Maintainer Piotr Szulc <piotr.michal.szulc@gmail.com>

Repository CRAN

Date/Publication 2025-03-10 10:00:02 UTC

Contents

aic 2
backward 3

bic 4
bigstep 4
fast_forward 6
forward 7
get_model 8
maic 9
maic2 9
mbic 10
mbic2 11
multi_backward 11
prepare_data 12
reduce_matrix 13
stepwise 14
summary.big 15

Index **16**

`aic` *AIC*

Description

Calculate AIC (Akaike Information Criterion).

Usage

```
aic(loglik, k)
```

Arguments

`loglik`	A numeric, the log-likelihood.
`k`	An integer >= 0, the number of selected variables.

Value

A number, a value of AIC.

Examples

```
aic(10, 5)
```

backward *Backward step*

Description

Remove the worst variable from a model according to the given criterion.

Usage

```
backward(data, crit = mbic, ...)
```

Arguments

data	an object of class big.
crit	a function defining the model selection criterion. You can use your own function or one of these: bic, mbic, mbic2, aic, maic, maic2.
...	optional arguments to crit.

Details

Type browseVignettes("bigstep") for more details.

Value

An object of class big.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
d %>%
  fast_forward(crit = aic) %>%
  backward() %>%
  backward()
```

bic *BIC*

Description

Calculate BIC (Bayesian Information Criterion).

Usage

```
bic(loglik, k, n)
```

Arguments

loglik	A numeric, the log-likelihood.
k	An integer >= 0, the number of selected variables.
n	An integer > 0, the number of observations.

Value

A number, a value of BIC.

Examples

```
bic(10, 5, 100)
```

bigstep *Model selection*

Description

Model selection using the stepwise procedure and the chosen criterion.

Details

The main goal of the package `bigstep` is to allow you to select a regression model using the stepwise procedure when data is very big, potentially larger than available RAM in your computer. What is more, the package gives you a lot of control over how this procedure should look like. At this moment, you can use one of these functions: `stepwise`, `forward`, `backward`, `fast_forward`, `multi_backward` and combinations of them. They can be treated as blocks from which the whole procedure of finding the best model is built.

When your data is larger than RAM you have in your computer, it is impossible to read it in a normal way. Fortunately, in a process of building a regression model it is not necessary to have access to all predictors at the same time. Instead, you can read only a part of the matrix `X`, check all variables from that part and then read another one. To do that with this package, you only need to read the matrix `X` using `read.big.matrix` from `bigmemory` package. The `prepare_data` function

has a parameter `maxp` which represents the maximum size (that is the number of elements) of one part. If `X` is bigger, it will be split. It will be done even if your matrix is big but you have enough RAM to read it in a normal way. It may seem unnecessary, but it is worth to do because R is not very efficient in dealing with big matrices.

Another problem with a large number of predictors is choosing an appropriate criterion. Classical ones like AIC or BIC are bad choice because they will almost certainly select a model with two many variables [1]. You can use modifications of them like mBIC [2], mBIC2 [3], mAIC or mAIC2. In brief, these criteria have much heavier penalty for the number of parameters, so they prefer smaller models than their classic versions.

If you want to read more, type `browseVignettes("bigstep")`

Author(s)

Piotr Szulc

References

[1] M. Bogdan, J.K. Ghosh, M. Zak-Szatkowska. Selecting explanatory variables with the modified version of Bayesian Information Criterion. Quality and Reliability Engineering International, 24:989-999, 2008.

[2] M. Bogdan, J.K. Ghosh, R.W. Doerge. Modifying the Schwarz Bayesian Information Criterion to locate multiple interacting quantitative trait loci. Genetics, 167:989-999, 2004.

[3] F. Frommlet, A. Chakrabarti, M. Murawska, M. Bogdan. Asymptotic Bayes optimality under sparsity for general distributions under the alternative, Technical report, arXiv:1005.4753v2, 2011.

Examples

```
## Not run:
library(bigstep)

### small data
set.seed(1)
n <- 200
p <- 20
X <- matrix(rnorm(n * p), ncol = p)
colnames(X) <- paste0("X", 1:p)
y <- 1 + 0.4 * rowSums(X[, c(5, 10, 15, 20)]) + rnorm(n)

data <- prepare_data(y, X)
results <- stepwise(data, crit = aic)
results$model
summary(results)

### bigger data
set.seed(1)
n <- 1e3
p <- 1e4
X <- matrix(rnorm(p * n), ncol = p)
colnames(X) <- paste0("X", 1:p)
Xadd <- matrix(rnorm(5 * n), n, 5)  # additional variables
```

```
colnames(Xadd) <- paste0("Xadd", 1:5)
y <- 0.2 * rowSums(X[, 1000 * (1:10)]) + Xadd[, 1] - 0.1 * Xadd[, 3] + rnorm(n)

data <- prepare_data(y, X, Xadd = Xadd)
data %>%
  reduce_matrix(minpv = 0.15) %>%
  stepwise(mbic) ->
  results
summary(results)

### big data
Xbig <- read.big.matrix("X.txt", sep = " ", header = TRUE,
                        backingfile = "X.bin", descriptorfile = "X.desc")
# Xbig <- attach.big.matrix("X.desc") # much faster
y <- read.table("y.txt")
# data <- prepare_data(y, Xbig) # slow because of checking NA
data <- prepare_data(y, Xbig, na = FALSE) # set if you know that you do not have NA
m <- data %>%
  reduce_matrix(minpv = 0.001) %>%
  fast_forward(crit = bic, maxf = 50) %>%
  multi_backward(crit = mbic) %>%
  stepwise(crit = mbic)
summary(m)

# more examples: type browseVignettes("bigstep")

## End(Not run)
```

`fast_forward` *Fast-forward step*

Description

Add variables to a model as long as they reduce the given criterion. Variables are searched according to `candidates` and every one which reduces the criterion is added (not necessarily the best one).

Usage

```
fast_forward(data, crit = bic, ..., maxf = 70)
```

Arguments

`data`	an object of class `big`.
`crit`	a function defining the model selection criterion. You can use your own function or one of these: `bic`, `mbic`, `mbic2`, `aic`, `maic`, `maic2`.
`...`	optional arguments to `crit`.
`maxf`	a numeric, a maximal number of variables in the final model.

Details

Type `browseVignettes("bigstep")` for more details.

Value

An object of class `big`.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
fast_forward(d)
```

`forward` *Forward step*

Description

Add the best variable to a model according to the given criterion.

Usage

```
forward(data, crit = mbic, ...)
```

Arguments

`data`	an object of class `big`.
`crit`	a function defining the model selection criterion. You can use your own function or one of these: `bic`, `mbic`, `mbic2`, `aic`, `maic`, `maic2`.
`...`	optional arguments to `crit`.

Details

Type `browseVignettes("bigstep")` for more details.

Value

An object of class `big`.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
forward(d, crit = bic)
d %>%
  forward() %>%
  forward() %>%
  forward()
```

`get_model` *Get the model*

Description

Extract the model (an object of class `lm`) from an object of class `big`.

Usage

```
get_model(object, ...)
```

Arguments

`object`	an object of class `big`.
`...`	Further arguments to be passed to or from other methods. They are ignored in this function.

Value

An object of class `lm`.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
m <- stepwise(d)
get_model(m)
```

maic *mAIC*

Description

Calculate mAIC (modified Akaike Information Criterion).

Usage

```
maic(loglik, k, p, const = 0.5)
```

Arguments

loglik	A numeric, the log-likelihood.
k	An integer >= 0, the number of selected variables.
p	An integer > 0, the number of all variables or a weight.
const	A numeric > 0, the expected number of significant variables.

Value

A number, a value of mAIC.

Examples

```
maic(10, 5, 100, 50)
```

maic2 *mAIC2*

Description

Calculate mAIC2 (the second version of modified Akaike Information Criterion).

Usage

```
maic2(loglik, k, n, p, const = 0.5)
```

Arguments

loglik	A numeric, the log-likelihood.
k	An integer >= 0, the number of selected variables.
n	An integer > 0, the number of observations, used only to check if k is not too large.
p	An integer > 0, the number of all variables or a weight.
const	A numeric > 0, the expected number of significant variables.

Value

A number, a value of mAIC2.

Examples

```
maic2(10, 5, 100, 50)
```

`mbic` *mBIC*

Description

Calculate mBIC (modified Bayesian Information Criterion).

Usage

```
mbic(loglik, k, n, p, const = 4)
```

Arguments

`loglik`	A numeric, the log-likelihood.
`k`	An integer >= 0, the number of selected variables.
`n`	An integer > 0, the number of observations.
`p`	An integer > 0, the number of all variables or a weight.
`const`	A numeric > 0, the expected number of significant variables.

Value

A number, a value of mBIC.

Examples

```
mbic(10, 5, 100, 50)
```

mbic2 *mBIC2*

Description

Calculate mBIC2 (the second version of modified Bayesian Information Criterion).

Usage

```
mbic2(loglik, k, n, p, const = 4)
```

Arguments

loglik	A numeric, the log-likelihood.
k	An integer >= 0, the number of selected variables.
n	An integer > 0, the number of observations.
p	An integer > 0, the number of all variables or a weight.
const	A numeric > 0, the expected number of significant variables.

Value

A number, a value of mBIC2.

Examples

```
mbic2(10, 5, 100, 50)
```

multi_backward *Multi-backward step*

Description

Remove the worst variables from a model as long as they reduce the given criterion (backward elimination).

Usage

```
multi_backward(data, crit = mbic, ...)
```

Arguments

data	an object of class big.
crit	a function defining the model selection criterion. You can use your own function or one of these: bic, mbic, mbic2, aic, maic, maic2.
...	optional arguments to crit.

Details

Type `browseVignettes("bigstep")` for more details.

Value

An object of class `big`.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
d %>%
  fast_forward(crit = aic) %>%
  multi_backward(crit = bic)
```

---

prepare_data *Data preparation*

Description

Create an object of class `big` which is needed to perform the selection procedure.

Usage

```
prepare_data(
  y,
  X,
  type = "linear",
  candidates = NULL,
  Xadd = NULL,
  na = NULL,
  maxp = 1e+06,
  verbose = TRUE
)
```

Arguments

`y`	a numeric vector of dependent (target) variable.
`X`	a numeric matrix or an object of class `big.matrix`. The columns of `X` should contain dependent variables (predictors).
`type`	a string, type of the regression model you want to fit. You can use one of these: `"linear"`, `"logistic"`, `"poisson"`.

candidates	a numeric vector, columns from X which will be used in the selection procedure. The order is important. If NULL, every column will be used.
Xadd	a numeric matrix, additional variables which will be included in the model selection procedure (they will not be removed in any step). If NULL, Xadd will contain only a column of ones (the intercept). If you specify Xadd, a column of ones will be automatically added (it is impossible to not include the intercept).
na	a logical. There are any missing values in X? If NULL, it will be checked (it can take some time if X is big, so it is reasonable to set it).
maxp	a numeric. The matrix X will be split into parts with maxp elements. It will not change results, but it is necessary if your computer does not have enough RAM. Set to a lower value if you still have problems.
verbose	a logical. Set FALSE if you do not want to see any information during the selection procedure.

Details

The function automatically removes observations which have missing values in y. Type browseVignettes("bigstep") for more details.

Value

An object of class big.

Examples

```
X <- matrix(rnorm(20), ncol = 4)
y <- X[, 2] + rnorm(5)
data <- prepare_data(y, X)
```

---

reduce_matrix *Reducing number of variables*

Description

Perform the Pearson correlation tests between a vector y and every variable from a matrix X (separately) and remove uncorrelated variables. The function is much faster when you do not have any missing values (set na = FALSE in prepare_data in that case).

Usage

```
reduce_matrix(data, minpv = 0.15)
```

Arguments

data	an object of class big.
minpv	a numeric. Variables with p-values for the Pearson correlation tests larger than minpv will be removed from candidates.

Details

Type `browseVignettes("bigstep")` for more details.

Value

An object of class `big`.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
reduce_matrix(d)
```

`stepwise` *Stepwise*

Description

Build a model according to the stepwise procedure (bidirectional) and the given criterion.

Usage

```
stepwise(data, crit = mbic, ...)
```

Arguments

`data`	an object of class `big`.
`crit`	a function defining the model selection criterion. You can use your own function or one of these: `bic`, `mbic`, `mbic2`, `aic`, `maic`, `maic2`.
`...`	optional arguments to `crit`.

Details

Type `browseVignettes("bigstep")` for more details.

Value

An object of class `big`.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
stepwise(d)
d %>%
  fast_forward(crit = aic) %>%
  stepwise(crit = bic)
```

---

summary.big *Summarizing model fit*

Description

summary method for class big.

Usage

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

Arguments

object	an object of class big.
...	Further arguments to be passed to or from other methods. They are ignored in this function.

Value

An object of class summary.lm.

Examples

```
set.seed(1)
n <- 30
p <- 10
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 2] + 2*X[, 3] - X[, 6] + rnorm(n)
d <- prepare_data(y, X)
m <- stepwise(d)
summary(m)
```

Index

aic, 2

backward, 3
bic, 4
bigstep, 4

fast_forward, 6
forward, 7

get_model, 8

maic, 9
maic2, 9
mbic, 10
mbic2, 11
multi_backward, 11

prepare_data, 12

reduce_matrix, 13

stepwise, 14
summary.big, 15