

# Package ‘BeSS’

June 29, 2017

**Type** Package

**Title** Best Subset Selection for Sparse Generalized Linear Model and Cox Model

**Version** 1.0.2

**Date** 2017-6-20

**Author** Canhong Wen, Aijun Zhang, Shijie Quan, Xueqin Wang

**Maintainer** Canhong Wen <wencanhong@gmail.com>

**Description** An implementation of best subset selection in generalized linear model and Cox proportional hazard model via the primal dual active set algorithm. The algorithm formulates coefficient parameters and residuals as primal and dual variables and utilizes efficient active set selection strategies based on the complementarity of the primal and dual variables.

**License** GPL-3

**Depends** R (>= 2.0.0)

**Imports** Rcpp, Matrix, glmnet, survival

**LinkingTo** Rcpp, RcppEigen

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2017-06-29 05:11:54 UTC

## R topics documented:

bess	2
bess.one	5
coef.bess	7
gen.data	8
plot.bess	10
predict.bess	11
summary.bess	12

<b>Index</b>	<b>14</b>
--------------	-----------

bess

*Best subset selection***Description**

Best subset selection for generalized linear model and Cox's proportional model.

**Usage**

```
bess(x, y, family = c("gaussian", "binomial", "cox"), method = "gsection",
     s.min = 1, s.max, s.list, K.max = 20, max.steps = 15, glm.max = 1e6, cox.max = 20,
     epsilon = 1e-4, normalize = TRUE)
```

**Arguments**

x	Input matrix, of dimension $n \times p$ ; each row is an observation vector.
y	Response variable, of length $n$ . For family="binomial" should be a factor with two levels. For family="cox", y should be a two-column matrix with columns named 'time' and 'status'.
family	One of the GLM or Cox models. Either "gaussian", "binomial", or "cox", depending on the response.
method	Methods to be used to select the optimal model size. For method = "sequential", we solve the best subset selection problem for each $s$ in $1, 2, \dots, s_{max}$ . At each model size $s$ , we run the bess function with a warm start from the last solution with model size $s - 1$ . For method = "gsection", we solve the best subset selection problem with a range non-continuous model sizes.
s.min	The minimum value of model sizes. Only used for method = "gsection". Default is 1.
s.max	The maximum value of model sizes. Only used for method = "gsection". Default is $\min p, n / \log(n)$ .
s.list	A list of sequential value representing the model sizes. Only used for method = "sequential". Default is $(1, \min p, n / \log(n))$ .
K.max	The maximum iterations used for method = "gsection"
max.steps	The maximum number of iterations in bess function. In linear regression, only a few steps can guarantee the convergence. Default is 15.
glm.max	The maximum number of iterations for solving the maximum likelihood problem on the active set at each step in the primal dual active set algorithm. Only used in the logistic regression for family="binomial". Default is 1e6.
cox.max	The maximum number of iterations for solving the maximum partial likelihood problem on the active set at each step in the primal dual active set algorithm. Only used in Cox's model for family="cox". Default is 20.
epsilon	The tolerance for an early stopping rule in the method "sequential". The early stopping rule is defined as $\ Y - X\beta\ /n \leq \epsilon$ .
normalize	whether to normalize x or not. Default is TRUE.

## Details

The best subset selection problem with model size  $s$  is

$$\min_{\beta} -2\log L(\beta) \quad \text{s.t.} \quad \|\beta\|_0 \leq s.$$

In the GLM case,  $\log L(\beta)$  is the log-likelihood function; In the Cox model,  $\log L(\beta)$  is the log partial likelihood function.

For each candidate model size, the best subset selection problem is solved by the primal dual active set (PDAS) algorithm, see Wen et al (2017) for details. This algorithm utilizes an active set updating strategy via primal and dual variables and fits the sub-model by exploiting the fact that their support set are non-overlap and complementary. For the case of method = "sequential", we run the PDAS algorithm for a list of sequential model sizes and use the estimate from last iteration as a warm start. For the case of method = "gsection", a golden section search technique is adopted to efficiently determine the optimal model size.

## Value

A list with class attribute 'bess' and named components:

family	Types of the model: "bess_gaussian" for linear model, "bess_binomial" for logistic model and "bess_cox" for Cox model.
beta	The best fitting coefficients of size $s = 0, 1, \dots, p$ with the smallest loss function.
lambda	The lambda value in the Lagrangian form of the best subset selection problem with model size of $s$ .
deviance	The value of $-2 \times \log L$ .
nulldeviance	The value of $-2 \times \log L$ for null model.
AIC	The value of $-2 \times \log L + 2\ \beta\ _0$ .
BIC	The value of $-2 \times \log L + \log(n)\ \beta\ _0$ .
EBIC	The value of $-2 \times \log L + (\log(n) + 2 \times \log(p))\ \beta\ _0$ .

## Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang.

## References

Wen, C., Zhang, A., Quan, S., and Wang, X. (2017) BeSS: A R package for best subset selection in GLM and CoxPH Models, Technical reports.

## See Also

[bess.one](#), [plot.bess](#), [predict.bess](#).

## Examples

```
#-----linear model-----#
# Generate simulated data
n <- 500
p <- 20
K <- 10
sigma <- 1
rho <- 0.2
data <- gen.data(n, p, family = "gaussian", K, rho, sigma)

# Best subset selection
fit <- bess(data$x, data$y, family = "gaussian")
summary(fit)
coef(fit, sparse=TRUE) # The estimated coefficients

# Plot solution path and the loss function
plot(fit, type = "both", breaks = TRUE)

#-----logistic model-----#

# Generate simulated data
data <- gen.data(n, p, family="binomial", 5, rho, sigma)

# Best subset selection
fit2 <- bess(data$x, data$y, s.list = 1:15, method = "sequential",
             family = "binomial", epsilon = 0)
summary(fit2)
coef(fit2, sparse = TRUE)

# Plot solution path and the loss function
plot(fit2, type = "both", breaks = TRUE, K = 5)

#-----cox model-----#

# Generate simulated data
data <- gen.data(n, p, K, rho, sigma, c = 10, family = "cox", scal = 10)

# Best subset selection
fit3 <- bess(data$x, data$y, s.list = 1:15, method = "sequential",
             family = "cox")
coef(fit3, sparse = TRUE)
summary(fit3)

# Plot solution path and the loss function
plot(fit3, type = "both", breaks = TRUE, K = 10)

#-----High dimensional linear models-----#

p <- 1000
data <- gen.data(n, p, family = "gaussian", K, rho, sigma)
```

```
# Best subset selection
fit <- bess(data$x, data$y, method="sequential", family = "gaussian", epsilon = 1e-12)

# Plot solution path
plot(fit, type = "both", breaks = TRUE, K = 10)
```

---

bess.one	<i>Best subset selection with a specified model size</i>
----------	--

---

### Description

Best subset selection with a specified model size for generalized linear models and Cox's proportional hazard model.

### Usage

```
bess.one(x, y, family = c("gaussian", "binomial", "cox"), s = 1, max.steps = 15,
         glm.max = 1e+6, cox.max = 20, normalize = TRUE)
```

### Arguments

<code>x</code>	Input matrix, of dimension $n \times p$ ; each row is an observation vector.
<code>y</code>	Response variable, of length $n$ . For family = "gaussian", <code>y</code> should be a vector with continuous values. For family = "binomial", <code>y</code> should be a factor with two levels. For family = "cox", <code>y</code> should be a two-column matrix with columns named 'time' and 'status'.
<code>s</code>	Size of the selected model. It controls number of nonzero coefficients to be allowed in the model.
<code>family</code>	One of the distribution function for GLM or Cox models. Either "gaussian", "binomial", or "cox", depending on the response.
<code>max.steps</code>	The maximum number of iterations in the primal dual active set algorithm. In most cases, only a few steps can guarantee the convergence. Default is 15.
<code>glm.max</code>	The maximum number of iterations for solving the maximum likelihood problem on the active set. It occurs at each step in the primal dual active set algorithm. Only used in the logistic regression for family = "binomial". Default is $1e + 6$ .
<code>cox.max</code>	The maximum number of iterations for solving the maximum partial likelihood problem on the active set. It occurs at each step in the primal dual active set algorithm. Only used in Cox model for family = "cox". Default is 20.
<code>normalize</code>	Whether to normalize <code>x</code> or not. Default is TRUE.

## Details

Given a model size  $s$ , we consider the following best subset selection problem:

$$\min_{\beta} -2\log L(\beta); s.t. \|\beta\|_0 = s.$$

In the GLM case,  $\log L(\beta)$  is the log-likelihood function; In the Cox model,  $\log L(\beta)$  is the log partial likelihood function.

The best subset selection problem is solved by the primal dual active set algorithm, see Wen et al. (2017) for details. This algorithm utilizes an active set updating strategy via primal and dual variables and fits the sub-model by exploiting the fact that their support set are non-overlap and complementary.

## Value

A list with class attribute 'bess.one' and named components:

type	Types of the model: "bess_gaussian" for linear model, "bess_binomial" for logistic model and "bess_cox" for Cox model
beta	The best fitting coefficients with the smallest loss function given the model size $s$ .
lambda	The estimated lambda value in the Lagrangian form of the best subset selection problem with model size $s$ .
deviance	The value of $-2 * \log L(\beta)$ .
nulldeviance	The value of $-2 * \log L(\beta)$ for null model.

## Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang.

## References

Wen, C., Zhang, A., Quan, S., and Wang, X. (2017) BeSS: A R package for best subset selection in GLM and CoxPH Models, Technical reports.

## See Also

[bess](#), [plot.bess](#), [predict.bess](#).

## Examples

```
#-----linear model-----#
# Generate simulated data

n <- 500
p <- 20
K <- 10
sigma <- 1
rho <- 0.2
```

```

data <- gen.data(n, p, family = "gaussian", K, rho, sigma)

# Best subset selection
fit1 <- bess.one(data$x, data$y, s = 10, family = "gaussian", normalize = TRUE)
coef(fit1, sparse=TRUE)

#-----logistic model-----#

# Generate simulated data
data <- gen.data(n, p, family = "binomial", K, rho, sigma)

# Best subset selection
fit2 <- bess.one(data$x, data$y, family = "binomial", s = 10, normalize = TRUE)

#-----cox model-----#

# Generate simulated data
data <- gen.data(n, p, K, rho, sigma, c=10, family="cox", scal=10)

# Best subset selection
fit3 <- bess.one(data$x, data$y, s = 10, family = "cox", normalize = TRUE)

#-----High dimensional linear models-----#

p <- 1000
data <- gen.data(n, p, family = "gaussian", K, rho, sigma)

# Best subset selection
fit <- bess.one(data$x, data$y, s=10, family = "gaussian", normalize = TRUE)

```

coef.bess

*Provides estimated coefficients from a fitted "bess" object.***Description**

Similar to other prediction methods, this function provides estimated coefficients from a fitted "bess" object or a fitted "bess.one" object.

**Usage**

```

## S3 method for class 'bess'
coef(object, sparse=TRUE, type = c("ALL", "AIC", "BIC", "EBIC"),...)
## S3 method for class 'bess.one'
coef(object, sparse = TRUE , ...)

```

Arguments

object	A "bess" project or a "bess.one" project.
sparse	Logical or NULL, specifying whether the coefficients should be presented as sparse matrix or not.
type	Types of coefficients returned. type = "AIC" cooresponds to the coefficient with optimal AIC value; type = "BIC" cooresponds to the coefficient with optimal BIC value; type = "EBIC" cooresponds to the coefficient with optimal EBIC value; type = "ALL" cooresponds to all coefficients in the bess object. Default is ALL.
...	Other arguments.

Author(s)

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang.

References

Wen, C., Zhang, A., Quan, S., and Wang, X. (2017) BeSS: A R package for best subset selection in GLM and CoxPH Models, Technical reports.

See Also

[bess](#), [bess.one](#)

Examples

```
data <- gen.data(500, 20, family = "gaussian", 10, 0.2, 1)
fit <- bess(data$x, data$y, family = "gaussian")
coef(fit, sparse=TRUE) # The estimated coefficients
```

---

gen.data	<i>Generate simulated data</i>
----------	--------------------------------

---

Description

Generate data for simulations under the generalized linear model and Cox model.

Usage

```
gen.data(n, p, family, K, rho = 0, sigma = 1, beta = NULL, censoring = TRUE,
         c = 1, scal)
```



**Arguments**

n	The number of observations.
p	The number of predictors of interest.
family	The distribution of the simulated data. "gaussian" for gaussian data."binomial" for binary data. "cox" for survival data
K	The number of nonzero coefficients in the underlying regression model.
rho	A parameter used to characterize the pairwise correlation in predictors. Default is 0.
sigma	A parameter used to control the signal-to-noise ratio. For linear regression, it is the error variance $\sigma^2$ . For logistic regression and Cox's model, the larger the value of sigma, the higher the signal-to-noise ratio.
beta	The coefficient values in the underlying regression model.
censoring	Whether data is censored or not. Default is TRUE
c	The censoring rate. Default is 1.
scal	A parameter in generating survival time based on the Weibull distribution. Only used for the "cox" family.

**Details**

For the design matrix  $X$ , we first generate an  $n \times p$  random Gaussian matrix  $\bar{X}$  whose entries are i.i.d.  $\sim N(0, 1)$  and then normalize its columns to the  $\sqrt{n}$  length. Then the design matrix  $X$  is generated with  $X_j = \bar{X}_j + \rho(\bar{X}_{j+1} + \bar{X}_{j-1})$  for  $j = 2, \dots, p-1$ .

For "gaussian" family, the data model is

$$Y = X\beta + \epsilon, \text{ where } \epsilon \sim N(0, \sigma^2).$$

The underlying regression coefficient  $\beta$  has uniform distribution  $[m, 100m]$ ,  $m = 5\sqrt{2\log(p)/n}$ .

For "binomial" family, the data model is

$$\text{Prob}(Y = 1) = \exp(X\beta) / (1 + \exp(X\beta))$$

The underlying regression coefficient  $\beta$  has uniform distribution  $[2m, 10m]$ ,  $m = 5\sigma\sqrt{2\log(p)/n}$ .

For "cox" family, the data model is

$$T = (-\log(S(t))/\exp(X\beta))^{(1/\text{scal})},$$

The centering time  $C$  is generated from uniform distribution  $[0, c]$ , then we define the censor status as  $\delta = IT \leq C$ ,  $R = \min T, C$ . The underlying regression coefficient  $\beta$  has uniform distribution  $[2m, 10m]$ ,  $m = 5\sigma\sqrt{2\log(p)/n}$ .

**Value**

A list with the following components: x, y, Tbeta.

x	Design matrix of predictors.
y	Response variable
Tbeta	The coefficients used in the underlying regression model.

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang.

**References**

Wen, C., Zhang, A., Quan, S., and Wang, X. (2017) BeSS: A R package for best subset selection in GLM and CoxPH Models, Technical reports.

**Examples**

```
# Generate simulated data
n <- 500
p <- 20
K <- 10
sigma <- 1
rho <- 0.2
data <- gen.data(n, p, family = "gaussian", K, rho, sigma)

# Best subset selection
fit <- bess(data$x, data$y, family = "gaussian")
```

---

plot.bess	<i>Produces a coefficient profile plot of the coefficient or loss function paths</i>
-----------	--

---

**Description**

Produces a coefficient profile plot of the coefficient or loss paths for a fitted "bess" object.

**Usage**

```
## S3 method for class 'bess'
plot(x, type=c("loss","coefficients","both"), breaks=TRUE, K=NULL, ...)
```

**Arguments**

x	a "bess" project
type	Either "both", "solutionPath" or "loss"
breaks	If TRUE, then vertical lines are drawn at each break point in the coefficient paths
K	which break point should the vertical lines drawn at
...	Other graphical parameters to plot

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang.

## References

Wen, C., Zhang, A., Quan, S., and Wang, X. (2017) BeSS: A R package for best subset selection in GLM and CoxPH Models, Technical reports.

## See Also

[bess](#), [bess.one](#)

## Examples

```
#-----linear model-----#

data <- gen.data(500, 20, family = "gaussian", 10, 0.2, 1)
fit <- bess(data$x, data$y, family = "gaussian")
plot(fit, type = "both")
```

---

predict.bess	<i>make predictions from a "bess" object.</i>
--------------	---

---

## Description

Similar to other predict methods, which returns predictions from a fitted "bess" object or a fitted "bess.one" object.

## Usage

```
## S3 method for class 'bess'
predict(object, newdata, type = c("ALL", "AIC", "BIC", "EBIC"),...)
## S3 method for class 'bess.one'
predict(object, newdata, ...)
```

## Arguments

object	Output from the bess function or the bess.one function.
newdata	New data used for prediction.
type	Types of coefficients returned. type = "AIC" cooresponds to the predictor with optimal AIC value; type = "BIC" cooresponds to the predictor with optimal BIC value; type = "EBIC" cooresponds to the predictor with optimal EBIC value; type = "ALL" cooresponds to all predictors in the bess object. Default is ALL.
...	Additional arguments affecting the predictions produced.

## Value

The object returned depends on the types of family.

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang.

**References**

Wen, C., Zhang, A., Quan, S., and Wang, X. (2017) BeSS: A R package for best subset selection in GLM and CoxPH Models, Technical reports.

**See Also**

[bess](#), [bess.one](#)

**Examples**

```
data <- gen.data(500, 20, family = "gaussian", 10, 0.2, 1)
fit <- bess(data$x, data$y, family = "gaussian")
pred=predict(fit, newdata = data$x)
```

---

summary.bess

*summary method for a "bess" object*


---

**Description**

Print a summary of the "bess" path at each step along the path.

**Usage**

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

**Arguments**

object	a "bess" project
...	additional summary arguments

**Author(s)**

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang.

**References**

Wen, C., Zhang, A., Quan, S., and Wang, X. (2017) BeSS: A R package for best subset selection in GLM and CoxPH Models, Technical reports.

### **See Also**

[bess](#), [bess.one](#)

### **Examples**

```
data <- gen.data(500, 20, family = "gaussian", 10, 0.2, 1)
fit <- bess(data$x, data$y, family = "gaussian")
summary.bess(fit)
```

# Index

bess, [2](#), [6](#), [8](#), [11–13](#)  
bess.one, [3](#), [5](#), [8](#), [11–13](#)  
coef.bess, [7](#)  
gen.data, [8](#)  
plot.bess, [3](#), [6](#), [10](#)  
predict.bess, [3](#), [6](#), [11](#)  
summary.bess, [12](#)