Package 'bestridge'

October 12, 2022

Type Package

Title A Comprehensive R Package for Best Subset Selection

Version 1.0.7

Date 2021-10-10

Maintainer Liyuan Hu <huly5@mail2.sysu.edu.cn>

Description The bestridge package is designed to provide a one-stand service for users to successfully carry out best ridge regression in various complex situations via the primal dual active set algorithm proposed by Wen, C., Zhang, A., Quan, S. and Wang, X. (2020) <doi:10.18637/jss.v094.i04>. This package allows users to perform the regression, classification, count regression and censored regression for (ultra) high dimensional data, and it also supports advanced usages like group variable selection and nuisance variable selection.

License GPL-3

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.3), Matrix(>= 1.2-6), MASS, pheatmap, survival

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Liyuan Hu [aut, cre] (<https://orcid.org/0000-0003-2064-8990>),
Jin Zhu [aut] (<https://orcid.org/0000-0001-8550-5822>),
Junxian Zhu [aut],
Kangkang Jiang [aut],
Yanhang Zhang [aut],
Xueqin Wang [aut] (<https://orcid.org/0000-0001-5205-9950>),
Canhong Wen [aut]

Repository CRAN

Date/Publication 2021-10-10 11:40:02 UTC

R topics documented:

`bestridge-package` *bestridge: A Comprehensive R Package for Best Subset Selection*

Description

The bestridge package is designed to provide a one-stand service for users to successfully carry out best ridge regression in various complex situations via the primal dual active set algorithm proposed by Wen, C., Zhang, A., Quan, S. and Wang, X. (2020) <doi:10.18637/jss.v094.i04>. This package allows users to perform the regression, classification, count regression and censored regression for (ultra) high dimensional data, and it also supports advanced usages like group variable selection and nuisance variable selection.

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu and Xueqin Wang, Canhong Wen.

References

Wen, C., Zhang, A., Quan, S. and Wang, X. (2020). BeSS: An R Package for Best Subset Selection in Linear, Logistic and Cox Proportional Hazards Models, *Journal of Statistical Software*, Vol. 94(4). doi:10.18637/jss.v094.i04.

bsrr *Best subset ridge regression*

Description

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

Usage

```
bsrr(
  x,
  y,
  family = c("gaussian", "binomial", "poisson", "cox"),
  method = c("pgsection", "sequential", "psequential"),
  tune = c("gic", "ebic", "bic", "aic", "cv"),
  s.list,
  lambda.list = 0,
  s.min,
  s.max,
  lambda.min = 0.001,
  lambda.max = 100,
  nlambda = 100,
  always.include = NULL,
  screening.num = NULL,
  normalize = NULL,
  weight = NULL,
  max.iter = 20,
  warm.start = TRUE,
  nfolds = 5,
  group.index = NULL,
  seed = NULL
)
```

Arguments

x	Input matrix, of dimension $n \times p$; each row is an observation vector and each column is a predictor/feature/variable.
y	The response variable, of n observations. For family = "binomial" should be a factor with two levels. For family="poisson", y should be a vector with positive integer. For family = "cox", y should be a two-column matrix with columns named time and status.
family	One of the following models: "gaussian", "binomial", "poisson", or "cox". Depending on the response. Any unambiguous substring can be given.
method	The method to be used to select the optimal model size and L_2 shrinkage. For method = "sequential", we solve the best subset ridge regression problem for each s in 1,2,...,s.max and λ in lambda.list. For method = "pgsection"

and "psequential", the Powell method is used to solve the best subset ridge regression problem. Any unambiguous substring can be given.

tune — The criterion for choosing the model size and L_2 shrinkage parameters. Available options are "gic", "ebic", "bic", "aic" and "cv". Default is "gic". "cv" is recommanded for BSRR.

s.list — An increasing list of sequential values representing the model sizes. Only used for method = "sequential". Default is 1:min(p,round(n/log(n))).

lambda.list — A lambda sequence for "bsrr". Only used for method = "sequential". Default is exp(seq(log(100), log(0.01), length.out = 100)).

s.min — The minimum value of model sizes. Only used for "psequential" and "pgsection". Default is 1.

s.max — The maximum value of model sizes. Only used for "psequential" and "pgsection". Default is min(p, round(n/log(n))).

lambda.min — The minimum value of lambda. Only used for method = "psequential" and "pgsection". Default is 0.001.

lambda.max — The maximum value of lambda. Only used for method = "psequential" and "pgsection". Default is 100.

nlambda — The number of λs for the Powell path with sequential line search method. Only valid for method = "psequential".

always.include — An integer vector containing the indexes of variables that should always be included in the model.

screening.num — Users can pre-exclude some irrelevant variables according to maximum marginal likelihood estimators before fitting a model by passing an integer to screening.num and the sure independence screening will choose a set of variables of this size. Then the active set updates are restricted on this subset.

normalize — Options for normalization. normalize = 0 for no normalization. Setting normalize = 1 will only subtract the mean of columns of x. normalize = 2 for scaling the columns of x to have $\sqrt{n}$ norm. normalize = 3 for subtracting the means of the columns of x and y, and also normalizing the columns of x to have $\sqrt{n}$ norm. If normalize = NULL, by default, normalize will be set 1 for "gaussian", 2 for "binomial" and "poisson", 3 for "cox".

weight — Observation weights. Default is 1 for each observation.

max.iter — The maximum number of iterations in the bsrr function. In most of the case, only a few steps can guarantee the convergence. Default is 20.

warm.start — Whether to use the last solution as a warm start. Default is TRUE.

nfolds — The number of folds in cross-validation. Default is 5.

group.index — A vector of integers indicating the which group each variable is in. For variables in the same group, they should be located in adjacent columns of x and their corresponding index in group.index should be the same. Denote the first group as 1, the second 2, etc. If you do not fit a model with a group structure, please set group.index = NULL. Default is NULL.

seed — Seed to be used to divide the sample into K cross-validation folds. Default is NULL.

Details

The best ridge regression problem with model size s and the shrinkage parameter λ is

$$\min_{\beta} -2\log L(\beta) + \lambda\|\beta\|_2^2 \text{ s.t. } \|\beta\|_0 \leq s.$$

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.

The best subset selection problem is a special case of the best ridge regression problem with the shrinkage $\lambda = 0$.

For each candidate model size and λ, the best subset ridge regression problems are solved by the L_2 penalized primal-dual active set (PDAS) algorithm, see Wen et al (2020) 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 sets are non-overlap and complementary. For the case of `method = "sequential"` if `warm.start = "TRUE"`, we run the PDAS algorithm for a list of sequential model sizes and use the estimate from the last iteration as a warm start. For the case of `method = "psequential"` and `method = "pgsection"`, the Powell method using a sequential line search method or a golden section search technique is used for parameters determination.

Value

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

`beta`	The best fitting coefficients.
`coef0`	The best fitting intercept.
`loss`	The training loss of the best fitting model.
`ic`	The information criterion of the best fitting model when model selection is based on a certain information criterion.
`cvm`	The mean cross-validated error for the best fitting model when model selection is based on the cross-validation.
`lambda`	The lambda chosen for the best fitting model
`beta.all`	For `bsrr` objects obtained by `gsection`, `pgsection` and `psequential`, `beta.all` is a matrix with each column be the coefficients of the model in each iterative step in the tuning path. For `bsrr` objects obtained by `sequential` method, A list of the best fitting coefficients of size `s=0,1,...,p` and λ in `lambda.list` with the smallest loss function. For `"bsrr"` objects of `"bsrr"` type, the fitting coefficients of the $i^{th}\lambda$ and the j^{th} `s` are at the i^{th} list component's j^{th} column.
`coef0.all`	For `bsrr` objects obtained from `gsection`, `pgsection` and `psequential`, `coef0.all` contains the intercept for the model in each iterative step in the tuning path. For `bsrr` objects obtained from `sequential` path, `coef0.all` contains the best fitting intercepts of size $s = 0, 1, \ldots, p$ and λ in `lambda.list` with the smallest loss function.
`loss.all`	For `bsrr` objects obtained from `gsection`, `pgsection` and `psequential`, `loss.all` contains the training loss of the model in each iterative step in the tuning path. For `bsrr` objects obtained from `sequential` path, this is a list of the training loss of the best fitting intercepts of model size $s = 0, 1, \ldots, p$ and λ in `lambda.list`. For `"bsrr"` object obtained by `"bsrr"`, the training loss of the $i^{th}\lambda$ and the j^{th} `s` is at the i^{th} list component's j^{th} entry.

ic.all	For bsrr objects obtained from gsection, pgsection and psequential, ic.all contains the values of the chosen information criterion of the model in each iterative step in the tuning path. For bsrr objects obtained from sequential path, this is a matrix of the values of the chosen information criterion of model size $s = 0, 1, \ldots, p$ and λ in lambda.list with the smallest loss function. For "bsrr" object obtained by "bsrr", the training loss of the $i^{th}\lambda$ and the j^{th} s is at the i^{th} row j^{th} column. Only available when model selection is based on a certain information criterion.
cvm.all	For bsrr objects obtained from gsection, pgsection and psequential, cvm.all contains the mean cross-validation error of the model in each iterative step in the tuning path. For bsrr objects obtained from sequential path, this is a matrix of the mean cross-validation error of model size $s = 0, 1, \ldots, p$ and λ in lambda.list with the smallest loss function. For "bsrr" object obtained by "bsrr", the training loss of the $i^{th}\lambda$ and the j^{th} s is at the i^{th} row j^{th} column. Only available when model selection is based on the cross-validation.
lambda.all	The lambda chosen for each step in pgsection and psequential.
family	Type of the model.
s.list	The input s.list.
nsample	The sample size.
type	Either "bss" or "bsrr".
method	Method used for tuning parameters selection.
ic.type	The criterion of model selection.

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

plot.bsrr, summary.bsrr, coef.bsrr, predict.bsrr.

Examples

```
#------------------linear model----------------------#
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
```

```
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bsrr(x, y, method = "pgsection")
coef(lm.bsrr)
print(lm.bsrr)
summary(lm.bsrr)
pred.bsrr <- predict(lm.bsrr, newx = x_new)

# generate plots
plot(lm.bsrr)
#------------------logistic model----------------------#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "binomial", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
logi.bsrr <- bsrr(x, y, family = "binomial", lambda.list = lambda.list)
coef(logi.bsrr)
print(logi.bsrr)
summary(logi.bsrr)
pred.bsrr <- predict(logi.bsrr, newx = x_new)

# generate plots
plot(logi.bsrr)
#------------------poisson model----------------------#
Data <- gen.data(n, p, k, rho=0.3, family = "poisson", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
poi.bsrr <- bsrr(x, y, family = "poisson", lambda.list = lambda.list)
coef(poi.bsrr)
print(poi.bsrr)
summary(poi.bsrr)
pred.bsrr <- predict(poi.bsrr, newx = x_new)

# generate plots
plot(poi.bsrr)
#------------------coxph model----------------------#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "cox", scal = 10, beta = Tbeta)

x <- Data$x[1:140, ]
y <- Data$y[1:140, ]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200, ]
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
cox.bsrr <- bsrr(x, y, family = "cox", lambda.list = lambda.list)
coef(cox.bsrr)
```

```
print(cox.bsrr)
summary(cox.bsrr)
pred.bsrr <- predict(cox.bsrr, newx = x_new)

# generate plots
plot(cox.bsrr)

#----------------------High dimensional linear models--------------------#
## Not run:
data <- gen.data(n, p = 1000, k, family = "gaussian", seed = seed)

# Best subset selection with SIS screening
lm.high <- bsrr(data$x, data$y, screening.num = 100)

## End(Not run)

#-------------------group selection----------------------#
beta <- rep(c(rep(1,2),rep(0,3)), 4)
Data <- gen.data(200, 20, 5, rho=0.4, beta = beta, seed =10)
x <- Data$x
y <- Data$y

group.index <- c(rep(1, 2), rep(2, 3), rep(3, 2), rep(4, 3),
                 rep(5, 2), rep(6, 3), rep(7, 2), rep(8, 3))
lm.groupbsrr <- bsrr(x, y, s.min = 1, s.max = 8, group.index = group.index)
coef(lm.groupbsrr)
print(lm.groupbsrr)
summary(lm.groupbsrr)
pred.groupl0l2 <- predict(lm.groupbsrr, newx = x_new)
#-------------------include specified variables----------------------#
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lm.bsrr <- bsrr(Data$x, Data$y, always.include = 2)
```

`coef.bsrr` *Provides estimated coefficients from a fitted "bsrr" object.*

Description

This function provides estimated coefficients from a fitted "bsrr" object.

Usage

```
## S3 method for class 'bsrr'
coef(object, sparse = TRUE, ...)
```

Arguments

`object`	A "bsrr" project.

sparse	Logical or NULL, specifying whether the coefficients should be presented as sparse matrix or not.
...	Other arguments.

Value

If sparse == FALSE, a vector containing the estimated coefficients from a fitted "bsrr" object is returned. If sparse == TRUE, a dgCMatrix containing the estimated coefficients is returned.

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

bsrr, print.bsrr.

Examples

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bsrr(Data$x, Data$y, method = "pgsection")
coef(lm.bsrr)
```

---

deviance.bsrr *Extract the deviance from a "bsrr.one" object.*

Description

Similar to other deviance methods, which returns deviance from a fitted "bsrr.one" object.

Usage

```
## S3 method for class 'bsrr'
deviance(object, best.model = TRUE, ...)
```

Arguments

`object`	A "`bsrr`" object.
`best.model`	Whether only return the loglikelihood of the best model. Default is `TRUE`. If `best.model = FALSE`, the loglikelihood of the best models with model size and λ in the original `s.list` and `lambda.list` (for `method = "sequential"`) or in the iteration path (for `method = "gsection"`, `method = "pgsection"`, and `method = "psequential"`) is returned.
`...`	additional arguments

Value

A matrix or vector containing the deviance for each model is returned. For `bsrr` object fitted by `sequantial` method, values in each row in the returned matrix corresponding to the model size in `s.list`, and each column the shrinkage parameters in `lambda.list`.

For `bsrr` object fitted by `gsection`, `pgsection` and `psequential`, the returned vector contains deviance for fitted models in each iteration. The coefficients of those model can be extracted from `beta.all` and `coef0.all` in the `bsrr` object.

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

`bsrr`, `summary.bsrr`.

Examples

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", seed = seed)
lm.bsrr <- bsrr(Data$x, Data$y, method = "sequential")

deviance(lm.bsrr)
deviance(lm.bsrr, best.model = FALSE)
```

duke *Duke breast cancer data*

Description

This data set details microarray experiment for breast cancer patients.

Format

A data frame with 46 rows and 7130 variables, where the first variable is the label of estrogen receptor-positive/negative, and the remaining 7129 variables are 7129 gene.

Details

The binary variable Status is used to classify the patients into estrogen receptor-positive (y = 0) and estrogen receptor-negative (y = 1). The other variables contain the expression level of the considered genes.

References

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.A. Olson, Jr., J.R. Marks and Joseph R. Nevins (2001) <doi:10.1073/pnas.201162998> Predicting the clinical status of human breast cancer by using gene expression profiles, Proceedings of the National Academy of Sciences of the USA, Vol 98(20), 11462-11467.

gen.data *Generate simulated data*

Description

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

Usage

```
gen.data(
  n,
  p,
  k = NULL,
  rho = 0,
  family = c("gaussian", "binomial", "poisson", "cox"),
  beta = NULL,
  cortype = 1,
  snr = 10,
  censoring = TRUE,
  c = 1,
  scal,
```

```
  sigma = 1,
  seed = 1
)
```

Arguments

n	The number of observations.
p	The number of predictors of interest.
k	The number of nonzero coefficients in the underlying regression model. Can be omitted if beta is supplied.
rho	A parameter used to characterize the pairwise correlation in predictors. Default is 0.
family	The distribution of the simulated data. "gaussian" for gaussian data."binomial" for binary data. "poisson" for count data. "cox" for survival data.
beta	The coefficient values in the underlying regression model.
cortype	The correlation structure. cortype = 1 denotes the exponential structure, where the covariance matrix has (i, j) entry equals $rho^{\|i-j\|}$. codecortype = 2 denotes the constant structure, where the (i, j) entry of covariance matrix is rho for every $i \neq j$ and 1 elsewhere. cortype = 3 denotes the moving average structure. Details can be found below.
snr	A numerical value controlling the signal-to-noise ratio (SNR). The SNR is defined as as the variance of $x\beta$ divided by the variance of a gaussian noise: $\frac{Var(x\beta)}{\sigma^2}$. The gaussian noise ϵ is set with mean 0 and variance. The noise is added to the linear predictor $\eta = x\beta$. Default is snr = 10. This option is invalid for cortype = 3.
censoring	Whether data is censored or not. Valid only for family = "cox". 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.
sigma	A parameter used to control the signal-to-noise ratio. For linear regression, it is the error variance σ^2. For logistic regression and Cox's model, the larger the value of sigma, the higher the signal-to-noise ratio. Valid only for cortype = 3.
seed	seed to be used in generating the random numbers.

Details

We generate an $n \times p$ random Gaussian matrix X with mean 0 and a covariance matrix with an exponential structure or a constant structure. For the exponential structure, the covariance matrix has (i, j) entry equals $rho^{|i-j|}$. For the constant structure, the (i, j) entry of the covariance matrix is rho for every $i \neq j$ and 1 elsewhere. For the moving average structure, 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 $j = 2, \ldots, p-1$.

For `family = "gaussian"` , the data model is

$$Y = X\beta + \epsilon.$$

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

For `family= "binomial"`, the data model is

$$Prob(Y = 1) = \exp(X\beta + \epsilon)/(1 + \exp(X\beta + \epsilon)).$$

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

For `family = "poisson"` , the data is modeled to have an exponential distribution:

$$Y = Exp(\exp(X\beta + \epsilon)).$$

For `family = "cox"`, the data model is

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

The centering time is generated from uniform distribution $[0, c]$, then we define the censor status as $\delta = I\{T \leq C\}, R = min\{T, C\}$. The underlying regression coefficient β has uniform distribution [2m, 10m], $m = 5\sigma\sqrt{2log(p)/n}$. In the above models, $\epsilon \sim N(0, \sigma^2)$, where σ^2 is determined by the `snr`.

Value

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

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

`bsrr`, `predict.bsrr`.

Examples

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
SNR <- 10
cortype <- 1
seed <- 10
Data <- gen.data(n, p, k, rho, family = "gaussian", cortype = cortype, snr = SNR, seed = seed)
x <- Data$x[1:140, ]
y <- Data$y[1:140]
```

```
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bsrr(x, y, method = "pgsection")
```

`gravier` *breast cancer data set*

Description

Gravier et al. (2010) have considered small, invasive ductal carcinomas without axillary lymph node involvement (T1T2N0) to predict metastasis of small node-negative breast carcinoma. Using comparative genomic hybridization arrays, they examined 168 patients over a five-year period. The 111 patients with no event after diagnosis were labelled good, and the 57 patients with early metastasis were labelled poor.

Format

A list containing the design matrix X and response matrix y

Value

No return value

Source

https://github.com/ramhiser

References

Eleonore Gravier., Gaelle Pierron., and Anne Vincent-Salomon. (2010). A prognostic DNA signature for T1T2 node-negative breast cancer patients.

`logLik.bsrr` *Extract the log-likelihood from a "bsrr.one" object.*

Description

This function returns the log-likelihood for the fitted models.

Usage

```
## S3 method for class 'bsrr'
logLik(object, best.model = TRUE, ...)
```

Arguments

object	A "bsrr" object.
best.model	Whether only return the log-likelihood of the best model. Default is TRUE. If best.model = FALSE, the log-likelihood of the best models with model size and λ in the original s.list and lambda.list (for method = "sequential") or in the iteration path (for method = "gsection", method = "pgsection", and method = "psequential") is returned.
...	additional arguments

Details

The log-likelihood for the best model chosen by a certain information criterion or cross-validation corresponding to the call in bsrr or the best models with model size and λ in the original s.list and lambda.list (or the in the iteration path) can be returned. For "lm" fits it is assumed that the scale has been estimated (by maximum likelihood or REML), and all the constants in the log-likelihood are included.

Value

A matrix or vector containing the log-likelihood for each model is returned. For bsrr objects fitted by sequantial method, values in each row in the returned matrix corresponding to the model size in s.list, and each column the shrinkage parameters in lambda.list.

For bsrr objects fitted by gsection, pgsection and psequential, the returned vector contains log-likelihood for fitted models in each iteration. The coefficients of those model can be extracted from beta.all and coef0.all in the bsrr object.

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

bsrr, summary.bsrr.

Examples

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
SNR <- 10
cortype <- 1
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", cortype = cortype, snr = SNR, seed = seed)
lm.bsrr <- bsrr(Data$x, Data$y, method = "sequential")
```

```
logLik(lm.bsrr, best.model = FALSE)
```

`patient.data` *Lymphoma patients data set*

Description

Lymphoma patients data set

Format

patient.data A list with survival times, staus and covariates from patients.

Details

A subset of the data set of lymphoma patients used in the study of Alizadeh et al. (2000) and also Simon et al. (2011).

References

Alizadeh, A. A., et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature, 403(6769), p.503 Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2011). Regularization paths for Cox's proportional hazards model via coordinate descent. Journal of statistical software, 39(5), 1.

`plot.bsrr` *Produces a coefficient profile plot of the coefficient or loss function paths*

Description

Produces a coefficient profile plot of the coefficient or loss function paths

Usage

```
## S3 method for class 'bsrr'
plot(
  x,
  type = c("tune", "coefficients"),
  lambda = NULL,
  sign.lambda = 0,
  breaks = T,
  K = NULL,
  ...
)
```

Arguments

x	A "bsrr" object.
type	One of "tune", "coefficients", "both". For "bsrr" with L_2 shrinkage: If (type = "tune"), the path of corresponding information criterion or cross-validation loss is provided; If type = "coefficients", a lambda should be provided and and this funciton provides a coefficient profile plot of the coefficient; For "bsrr" object without L_2 shrinkage: If type = "tune", a path of lcorresponding information criterion or cross-validation loss is provided. If type = "coefficients", it provides a coefficient profile plot of the coefficient.
lambda	For "bsrr" with L_2 shrinkage: To plot the change of coefficients with lambda equals this value for type = "coefficients" or type = "both".
sign.lambda	For "bsrr" with L_2 shrinkage: A logical value indicating whether to show lambda on log scale. Default is 0.
breaks	For "bsrr" object without L_2 shrinkage: If TRUE, a vertical line is drawn at a specified break point in the coefficient paths.
K	For "bsrr" object without L_2 shrinkage: Which break point should the vertical line be drawn at. Default is the optimal model size.
...	Other graphical parameters to plot

Value

No return value, called for plots generation

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

bsrr.

Examples

```
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bsrr(Data$x, Data$y, method = "pgsection")

# generate plots
plot(lm.bsrr)
```

predict.bsrr *make predictions from a "bsrr" object.*

Description

Returns predictions from a fitted "bsrr" object.

Usage

```
## S3 method for class 'bsrr'
predict(object, newx, type = c("link", "response"), ...)
```

Arguments

object	Output from the bsrr function.
newx	New data used for prediction. If omitted, the fitted linear predictors are used.
type	type = "link" gives the linear predictors for "binomial", "poisson" or "cox" models; for "gaussian" models it gives the fitted values. type = "response" gives the fitted probabilities for "binomial", fitted mean for "poisson" and the fitted relative-risk for "cox"; for "gaussian", type = "response" is equivalent to type = "link"
...	Additional arguments affecting the predictions produced.

Value

The object returned depends on the types of family.

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

bsrr.

Examples

```
#-------------------linear model----------------------#
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
```

```
x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bsrr(x, y, method = "pgsection")

pred.bsrr <- predict(lm.bsrr, newx = x_new)

#------------------logistic model---------------------#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "binomial", beta = Tbeta, seed = seed)

x <- Data$x[1:140, ]
y <- Data$y[1:140]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200]
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
logi.bsrr <- bsrr(x, y, tune="cv",
                 family = "binomial", lambda.list = lambda.list, method = "sequential")

pred.bsrr <- predict(logi.bsrr, newx = x_new)

#------------------coxph model---------------------#
#Generate simulated data
Data <- gen.data(n, p, k, rho, family = "cox", beta = Tbeta, scal = 10)

x <- Data$x[1:140, ]
y <- Data$y[1:140, ]
x_new <- Data$x[141:200, ]
y_new <- Data$y[141:200, ]
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
cox.bsrr <- bsrr(x, y, family = "cox", lambda.list = lambda.list)

pred.bsrr <- predict(cox.bsrr, newx = x_new)

#------------------group selection---------------------#
beta <- rep(c(rep(1,2),rep(0,3)), 4)
Data <- gen.data(200, 20, 5, rho=0.4, beta = beta, seed =10)
x <- Data$x
y <- Data$y

group.index <- c(rep(1, 2), rep(2, 3), rep(3, 2), rep(4, 3),
                 rep(5, 2), rep(6, 3), rep(7, 2), rep(8, 3))
lm.groupbsrr <- bsrr(x, y, s.min = 1, s.max = 8, group.index = group.index)

pred.groupbsrr <- predict(lm.groupbsrr, newx = x_new)
```

`print.bsrr` *print method for a "bsrr" object*

Description

Print the primary elements of the "bsrr" object.

Usage

```
## S3 method for class 'bsrr'
print(x, digits = max(5, getOption("digits") - 5), nonzero = FALSE, ...)
```

Arguments

x	A "bsrr" object.
digits	Minimum number of significant digits to be used.
nonzero	Whether the output should only contain the non-zero coefficients.
...	additional print arguments

Details

prints the fitted model and returns it invisibly.

Value

No return value, called for side effect

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

bsrr, coef.bsrr.

Examples

```
# Generate simulated data
n = 200
p = 20
k = 5
rho = 0.4
seed = 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data = gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed=seed)
lambda.list = exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr = bsrr(Data$x, Data$y, lambda.list = lambda.list, method = "sequential")

print(lm.bsrr)
```

prostate *Factors associated with prostate specific antigen*

Description

Data from a study by by Stamey et al. (1989) to examine the association between prostate specific antigen (PSA) and several clinical measures that are potentially associated with PSA in men who were about to receive a radical prostatectomy. The variables are as follows:

- lcavol: Log cancer volume
- lweight: Log prostate weight
- age: The man's age
- lbph: Log of the amount of benign hyperplasia
- svi: Seminal vesicle invasion; 1=Yes, 0=No
- lcp: Log of capsular penetration
- gleason: Gleason score
- pgg45: Percent of Gleason scores 4 or 5
- lpsa: Log PSA

Format

A data frame with 97 observations on 9 variables

Value

No return value

References

Stamey, T., Kabalin, J., McNeal, J., Johnstone, I., Freiha, F., Redwine, E. and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate II. Radical prostatectomy treated patients, Journal of Urology 16: 1076-1083.

SAheart *Risk factors associated with heart disease*

Description

Data from a subset of the Coronary Risk-Factor Study baseline survey, carried out in rural South Africa.

Format

The variables are as follows:

- sbp: Systolic blood pressure
- tobacco: Cumulative tobacco consumption, in kg
- ldl: Low-density lipoprotein cholesterol
- adiposity: Adipose tissue concentration
- famhist: Family history of heart disease (1=Present, 0=Absent)
- typea: Score on test designed to measure type-A behavior
- obesity: Obesity
- alcohol: Current consumption of alcohol
- age: Age of subject
- chd: Coronary heart disease at baseline; 1=Yes 0=No

A data frame with 462 observations on 10 variables

Value

No return value

References

Rousseauw, J., du Plessis, J., Benade, A., Jordaan, P., Kotze, J. and Ferreira, J. (1983). Coronary risk factor screening in three rural communities. South African Medical Journal 64: 430-436.

`summary.bsrr` *summary method for a "bsrr" object*

Description

Print a summary of the "bsrr" object.

Usage

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

Arguments

`object`	A "bsrr" object.
`...`	additional print arguments

Value

No return value

Author(s)

Liyuan Hu, Kangkang Jiang, Yanhang Zhang, Jin Zhu, Canhong Wen and Xueqin Wang.

See Also

bsrr.

Examples

```
#------------------linear model---------------------#
# Generate simulated data
n <- 200
p <- 20
k <- 5
rho <- 0.4
seed <- 10
Tbeta <- rep(0, p)
Tbeta[1:k*floor(p/k):floor(p/k)] <- rep(1, k)
Data <- gen.data(n, p, k, rho, family = "gaussian", beta = Tbeta, seed = seed)
lambda.list <- exp(seq(log(5), log(0.1), length.out = 10))
lm.bsrr <- bsrr(Data$x, Data$y, method = "pgsection")

summary(lm.bsrr)

#------------------group selection---------------------#
beta <- rep(c(rep(1,2),rep(0,3)), 4)
Data <- gen.data(200, 20, 5, rho=0.4, beta = beta, snr = 100, seed =10)

group.index <- c(rep(1, 2), rep(2, 3), rep(3, 2), rep(4, 3),
                 rep(5, 2), rep(6, 3), rep(7, 2), rep(8, 3))
lm.groupbsrr <- bsrr(Data$x, Data$y, s.min = 1, s.max = 8, group.index = group.index)

summary(lm.groupbsrr)
```

---

trim32 *The Bardet-Biedl syndrome Gene expression data*

Description

Gene expression data (500 gene probes for 120 samples) from the microarray experiments of mammalianeye tissue samples of Scheetz et al. (2006).

Format

A data frame with 120 rows and 501 variables, where the first variable is the expression level of TRIM32 gene, and the remaining 500 variables are 500 gene probes.

Details

In this study, laboratory rats (Rattus norvegicus) were studied to learn about gene expression and regulation in the mammalian eye. Inbred rat strains were crossed and tissue extracted from the eyes of 120 animals from the F2 generation. Microarrays were used to measure levels of RNA expression in the isolated eye tissues of each subject. Of the 31,000 different probes, 18,976 were detected at a sufficient level to be considered expressed in the mammalian eye. For the purposes of this analysis, we treat one of those genes, Trim32, as the outcome. Trim32 is known to be linked with a genetic disorder called Bardet-Biedl Syndrome (BBS): the mutation (P130S) in Trim32 gives rise to BBS.

Note

This data set contains 120 samples with 500 predictors. The 500 predictors are features with maximum marginal correlation to Trim32 gene.

References

T. Scheetz, k. Kim, R. Swiderski, A. Philp, T. Braun, K. Knudtson, A. Dorrance, G. DiBona, J. Huang, T. Casavant, V. Sheffield, E. Stone. Regulation of gene expression in the mammalian eye and its relevance to eye disease. Proceedings of the National Academy of Sciences of the United States of America, 2006.

Index