Package 'robustsubsets'

June 1, 2020

Type Package

Title Robust subset selection in linear regression

Version 1.0.0

Author Ryan Thompson

Maintainer Ryan Thompson <ryan.thompson@monash.edu>

Description Provides functionality for robust subset selection in linear regression.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports stats,
foreach,
doParallel,
abind,
gurobi,
Matrix,
ggplot2,
Rcpp

LinkingTo Rcpp,
RcppArmadillo

R topics documented:

`bss` *Best subset selection*

Description

Fits a sequence of best subset selection models. This function is just a wrapper for the `rss` function. The function solves the robust subset selection problem with `h=n`, using nonrobust measures of location and scale to standardise, as well as a nonrobust measure of prediction error in cross-validation.

Usage

```
bss(X, y, k = (!int):min(nrow(X) - int, ncol(X), 20), int = T, mio = T, ...)
```

Arguments

`X`	a matrix of predictors
`y`	a vector of the response
`k`	the number of predictors to minimise sum of squares over; by default a sequence from 0 to 20
`int`	a logical indicating whether to include an intercept
`mio`	a logical indicating whether to run the mixed-integer solver
`...`	any other arguments

Value

See documentation for the `rss` function.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

coef.rss *Coefficient function for rss object*

Description

Extracts coefficients for a given parameter pair (k,h).

Usage

```
## S3 method for class 'rss'
coef(object, k = "min.k", h = "min.h", ...)
```

Arguments

object	an object of class rss
k	the number of predictors indexing the desired fit; 'min.k' uses best k from cross-validation
h	the number of observations indexing the desired fit; 'min.h' uses best h from cross-validation
...	any other arguments

Value

A vector of coefficients.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

coef.rss.fit *Coefficient function for rss.fit object*

Description

Extracts coefficients for a given parameter pair (k,h).

Usage

```
## S3 method for class 'rss.fit'
coef(object, k, h, ...)
```

Arguments

object	an object of class rss.fit
k	the number of predictors indexing the desired fit
h	the number of observations indexing the desired fit
...	any other arguments

Value

A vector of coefficients.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

---

plot.rss *Plot function for rss object*

Description

Plot the cross-validation results or coefficient profiles from robust subset selection.

Usage

```
## S3 method for class 'rss'
plot(x, type = "cv", ...)
```

Arguments

x	an object of class rss
type	one of 'cv' or 'profile'
...	any other arguments

Value

A plot of the cross-validation results or coefficient profiles.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

plot.rss.cv *Plot function for rss.cv object*

Description

Plot the cross-validation results from robust subset selection.

Usage

```
## S3 method for class 'rss.cv'
plot(x, ...)
```

Arguments

x	an object of class rss.cv
...	any other arguments

Value

A plot of the cross-validation results.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

plot.rss.fit *Plot function for rss.fit object*

Description

Plot the coefficient profiles from robust subset selection.

Usage

```
## S3 method for class 'rss.fit'
plot(x, ...)
```

Arguments

x	an object of class rss.fit
...	any other arguments

Value

A plot of the coefficient profiles.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

`predict.rss` *Predict function for rss object*

Description

Generate predictions given new data using a given parameter pair `(k,h)`.

Usage

```
## S3 method for class 'rss'
predict(object, X.new, k = "min.k", h = "min.h", ...)
```

Arguments

`object`	an object of class `rss`
`X.new`	a matrix of new values for the predictors
`k`	the number of predictors indexing the desired fit; 'min.k' uses best `k` from cross-validation
`h`	the number of observations indexing the desired fit; 'min.h' uses best `h` from cross-validation
`...`	any other arguments

Value

A vector of predictions.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

predict.rss.fit *Predict function for rss.fit object*

Description

Generate predictions for new data using a given parameter pair (k,h).

Usage

```
## S3 method for class 'rss.fit'
predict(object, X.new, k, h, ...)
```

Arguments

object	an object of class rss.fit
X.new	a matrix of new values for the predictors
k	the number of predictors indexing the desired fit
h	the number of observations indexing the desired fit
...	any other arguments

Value

A vector of predictions.

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

rss *Robust subset selection*

Description

Fits a sequence of robust subset selection models and cross-validates the prediction error from these models.

Usage

```
rss(
  X,
  y,
  k = (!int):min(nrow(X) - int, ncol(X), 20),
  h = floor(seq(0.75, 1, 0.05) * nrow(X)),
  int = T,
  mio = T,
  ...
)
```

Arguments

X	a matrix of predictors
y	a vector of the response
k	the number of predictors to minimise sum of squares over; by default a sequence from 0 to 20
h	the number of observations to minimise sum of squares over; by default a sequence from 75 to 100 percent of sample size (in increments of 5 percent)
int	a logical indicating whether to include an intercept
mio	a logical indicating whether to run the mixed-integer solver
...	any other arguments (see rss.fit and rss.cv)

Details

This function fits a sequence of models and cross-validates the prediction error associated with these models. In the interest of speed, these steps are carried out using heuristic optimisation methods. The parameters that produce the lowest cv error are run through the mixed-integer solver which (given sufficient time) will find a global minimiser.
See rss.fit and rss.cv for further options controlling the model fit and cross-validation.

Value

An object of class rss; a list with the following components:

cv	the output from rss.cv; see documentation
fit	the output from rss.fit; see documentation

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

Examples

```
# Set simulation parameters
set.seed(1)
n <- 100
p <- 10
p0 <- 5
n.c <- 10

# Generate training data with mixture error
beta <- c(rep(1, p0), rep(0, p - p0))
X <- matrix(rnorm(n * p), n, p)
e <- rnorm(n, c(rep(10, n.c), rep(0, n - n.c)))
y <- X %*% beta + e

# Fit best/robust subset selection models
fit.bss <- bss(X, y, n.core = 1)
fit.rss <- rss(X, y, n.core = 1)
```

```
# Extract model coefficients
bss.beta <- coef(fit.bss)
rss.beta <- coef(fit.rss)

# Check estimation error
ee.bss <- norm(bss.beta - c(0, beta), '2')
ee.rss <- norm(rss.beta - c(0, beta), '2')
cat('Best subsets estimation error:', ee.bss, '\n')
cat('Robust subsets estimation error:', ee.rss, '\n')

# Plot coefficient profiles
plot(fit.rss, type = 'profile')
# Each facet corresponds to a different value of h

# Plot cross-validation results
plot(fit.rss, type = 'cv')
# Each line corresponds to a different value of h

# Generate test data
X.test <- matrix(rnorm(n * p), n, p)
e.test <- rnorm(n)
y.test <- X.test %*% beta + e.test

# Make model predictions (using best parameters from cv)
pred.bss <- predict(fit.bss, X.test)
pred.rss <- predict(fit.rss, X.test)

# Compute prediction error
pe.bss <- 1 / n * norm(y.test - pred.bss, '2') ^ 2
pe.rss <- 1 / n * norm(y.test - pred.rss, '2') ^ 2
cat('Best subsets prediction error:', pe.bss, '\n')
cat('Robust subsets prediction error:', pe.rss, '\n')
```

`rss.cv` *Cross-validation for robust subset selection*

Description

Does (repeated) K-fold cross-validation for robust subset selection in parallel. To achieve good run time, only uses the heuristics (by default).

Usage

```
rss.cv(
  X,
  y,
  k = (!int):min(nrow(X) - int, ncol(X), 20),
  h = floor(seq(0.75, 1, 0.05) * nrow(X)),
```

```
  int = T,
  n.fold = 10,
  n.cv = 1,
  n.cores = parallel::detectCores(),
  cv.objective = tmspe,
  ...
)
```

Arguments

X	a matrix of predictors
y	a vector of the response
k	the number of predictors to minimise sum of squares over; by default a sequence from 0 to 20
h	the number of observations to minimise sum of squares over; by default a sequence from 75 to 100 percent of sample size (in increments of 5 percent)
int	a logical indicating whether to include an intercept
n.fold	the number of folds to use in cross-validation
n.cv	the number of times to repeat cross-validation; the results are averaged
n.cores	the number of cores to use in cross-validation; by default all cores are used
cv.objective	the cross-validation objective function; by default trimmed mean square prediction error with 25 percent trimming
...	any other arguments

Value

An object of class rss.cv; a list with the following components:

mean.cv	a matrix with the cross-validated values of cv.objective; each row corresponds to a value of k and each column to a value of h
min.k	the k yielding the lowest cross-validated cv.objective
min.h	the h yielding the lowest cross-validated cv.objective
k	the value of k that was passed in
h	the value of h that was passed in

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

`rss.fit` *Robust subset selection*

Description

Fits a sequence of robust subset selection models using a combination of heuristics and mixed-integer optimisation (mio).

Usage

```
rss.fit(
  X,
  y,
  k = (!int):min(nrow(X) - int, ncol(X), 20),
  h = floor(seq(0.75, 1, 0.05) * nrow(X)),
  int = T,
  k.mio = NA,
  h.mio = NA,
  time = 300,
  tau = 2,
  focus = 0,
  log = F,
  output = T,
  robust = T,
  max.iter.ns = 100,
  max.iter.gd = 1e+05,
  tol = 1e-04,
  ...
)
```

Arguments

`X`	a matrix of predictors
`y`	a vector of the response
`k`	the number of predictors to minimise sum of squares over (i.e. the model sparsity); by default a sequence from 0 to 20
`h`	the number of observations to minimise sum of squares over; by default a sequence from 75 to 100 percent of sample size (in increments of 5 percent)
`int`	a logical indicating whether to include an intercept
`k.mio`	the subset of `k` for which the mixed-integer solver should be run
`h.mio`	the subset of `h` for which the mixed-integer solver should be run
`time`	a time limit in seconds on each call to the mixed-integer solver
`tau`	a positive number greater than 1 used to tighten variable bounds in the mixed-integer formulation; small values give quicker run times but can also exclude the optimal solution

focus	an integer in {0,1,2,3} used to tune the high level strategy of the mixed-integer solver
log	a logical indicating whether to save the mixed-integer solver output
output	a logical indicating whether to print status updates
robust	a logical indicating whether to standardise the data robustly; median/mad for true and mean/sd for false
max.iter.ns	the maximum number of neighbourhood search iterations to perform; if output is true then the number of iterations required for convergence will be printed
max.iter.gd	the maximum number of gradient descent iterations to perform
tol	a numerical tolerance parameter used to declare convergence
...	any other arguments

Details

The function first computes solutions over all combinations of k and h using heuristics. The solutions can then be refined further using the mixed-integer solver. The values that the solver operates on are specified by the k.mio and h.mio parameters, which must be subsets of k and h.
The focus parameter tells the mio solver whether to focus on improving the upper bound, lower bound, or to balance both goals. See https://www.gurobi.com/documentation/9.0/refman/mipfocus.html.
If robust is set to true and the median of any predictor is zero, then the data cannot be standardised (the median absolute deviation is undefined) and an error message will be returned.

Value

An object of class rss.fit; a list with the following components:

beta	a 3d array of estimated regression coefficients; each column of regression coefficients corresponds to fixed value of k and each matrix to fixed value of h
eta	a 3d array of estimated residual outliers; each column of residual outliers corresponds to a fixed value of k and each matrix to fixed value of h
objval	a matrix with the objective function values; each row corresponds to a value for different k and each column to a value for different h
k	the value of k that was passed in
h	the value of h that was passed in
int	whether an intercept was included

Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

Index