Package ‘rqPen’

March 15, 2015

Type Package
Title Penalized Quantile Regression
Version 1.1
Date 2014-03-15
Author Ben Sherwood
Depends R (>= 3.0.0),quantreg
Maintainer Ben Sherwood <bsherwo2@jhu.edu>
Description Performs penalized quantile regression for LASSO, SCAD and MCP functions. Provides a function that automatically generates lambdas and evaluates different models with cross validation or BIC, including a large p version of BIC.
ByteCompile TRUE
License MIT + file LICENSE
NeedsCompilation no
Repository CRAN
Date/Publication 2015-03-15 17:47:08

R topics documented:

beta_plots ................................................................. 2
check ................................................................. 3
cv.rq.pen ............................................................. 4
cv_plots ............................................................... 5
lasso ................................................................. 6
mcp ................................................................. 7
mcp_deriv ............................................................ 8
model_eval ............................................................ 9
pos_part ............................................................... 10
predict.cv.rq.pen ................................................... 10
predict.rq.pen ....................................................... 11
qbic ............................................................... 12
randomly_assign .................................................. 13
rq.lasso.fit .......................................................... 13
Plots how the beta estimates changes with the different values of lambda.

Usage

```
beta_plots(model, voi=NULL, logLambda=FALSE, loi=NULL,...)
```

Arguments

- `model`: "cv.rq.pen" object.
- `voi`: Index of betas to be plotted. Default is all of the lambdas from "cv.rq.pen" object.
- `logLambda`: Plot of lambdas is on the log scale.
- `loi`: Index of lambdas to be plotted. Default is all of the lambdas from "cv.rq.pen" object.
- `...`: Additional arguments to be sent to plot.

Value

Plot of how beta estimates change with lambda.

Author(s)

Ben Sherwood

Examples

```r
x <- matrix(rnorm(800),nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
lassoModels <- cv.rq.pen(x,y)
b_plot <- beta_plots(lassoModels)
```
check  

Quantile check function

Description

Evaluates the check function for quantile tau at value x. Check function is the objective function defined in Koenker and Bassett (1978).

Usage

check(x, tau)

Arguments

x  Number to be evaluated.

tau  Number between 0 and 1 for quantile of interest.

Value

x*(tau-I(x < 0))

Author(s)

Ben Sherwood

References


Examples

check(2,.5)
check(-2,.5)
check(2,.2)
check(2,.8)
cv.rq.pen

Cross Validated quantile regression

Description

Produces penalized quantile regression models for a range of lambdas and penalty of choice. If lambda is unselected than an iterative algorithm is used to find a maximum lambda such that the penalty is large enough to produce an intercept only model. Then range of lambdas goes from the maximum lambda found to "eps" on the log scale. For non-convex penalties local linear approximation approach used by Wang, Wu and Li to extend LLA as proposed by Zou and Li (2008) to the quantile regression setting.

Usage

cv.rq.pen(x,y,tau=.5,lambda=NULL,weights=NULL,penalty="LASSO", intercept=TRUE,criteria="CV",cvFunc="check",nfolds=10,
foldid=NULL,nlambda=100,eps=.0001,init.lambda=1,...)

Arguments

x     Matrix of predictors.
y     Vector of response values.
tau  Conditional quantile being modeled.
lambda Vector of lambdas. Default is for lambdas to be automatically generated.
weights Weights for the objective function.
penalty Type of penalty: "LASSO", "SCAD" or "MCP".
intercept Whether model should include an intercept. Constant does not need to be included in "x".
criteria How models will be evaluated. Either cross-validation "CV", BIC "BIC" or large P BIC "PBIC".
cvFunc If cross-validation is used how errors are evaluated. Check function "check", "SqErr" (Squared Error) or "AE" (Absolute Value).
nfolds K for K-folds cross-validation.
foldid Group id for cross-validation. Function will randomly generate groups if not specified.
nlambda Number of lambdas for which models are fit.
eps Smallest lambda used.
init.lambda Initial lambda used to find the maximum lambda. Not needed if lambda values are set.
... Additional arguments to be sent to rq.lasso.fit or rq.nc.fit.
cv_plots

Value
Returns the following:

models       List of penalized models fit. Number of models will match number of lambdas and correspond to cv$lambda.
cv           Data frame with "lambda" and second column is the evaluation based on the criteria selected.
lambda.min   Lambda which provides the smallest statistic for the selected criteria.
penalty      Penalty selected.

Author(s)
Ben Sherwood

References

Examples
x <- matrix(rnorm(800),nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
cv_model <- cv.rq.pen(x,y)

______________________________
cv_plots                       Plots of Cross-validation results
______________________________

Description
Slightly misnamed as user could choose BIC as a criteria for "cv.rq.pen" object. Function is able to discern between the two types of evaluation criterias and provides appropriate labels for the plot.

Usage
cv_plots(model,logLambda=TRUE,loi=NULL,...)

Arguments
model       "cv.rq.pen" object.
logLambda   Plot of lambdas is on the log scale.
loi         Index of lambdas to be plotted. Default is all of the lambdas from "cv.rq.pen" object.
...         Additional items to be sent to plot function.
Value
Plot of how cross validation statistic changes with lambda.

Author(s)
Ben Sherwood

Examples
```r
x <- matrix(rnorm(800), nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
lassoModels <- cv.rq.pen(x, y)
cv_plot <- cv_plots(lassoModels)
```

---

lasso \textit{Lasso}

Description
LASSO penalty function.

Usage
```r
lasso(x, lambda)
```

Arguments
- `x`: Number to be evaluated
- `lambda`: Tuning parameter lambda

Value
`lambda * abs(x)`

Author(s)
Ben Sherwood

References

Examples
```r
lasso(3, 1)
lasso(-3, 1)
lasso(-3, 2)
```
Description

MCP function as described in Fan and Li (2001).

Usage

mcp(x, lambda, a)

Arguments

x Number to be evaluated
lambda Tuning parameter lambda
a Tuning parameter a

Value

MCP function with tuning parameters lambda and "a" evaluated at "x".

Author(s)

Ben Sherwood

References


Examples

mcp(3, 1)
mcp(-3, 1)
mcp(-0.01, 2)
Description

Derivative of MCP function as described in Fan and Li (2001).

Usage

mcp_deriv(x, lambda, a)

Arguments

x  Number to be evaluated
lambda  Tuning parameter lambda
a  Tuning parameter a

Value

Derivative of MCP function with tuning parameters lambda and "a" evaluated at "x".

Author(s)

Ben Sherwood

References


Examples

mcp(3, 1)
mcp(-3, 1)
mcp(.001, 2)
**model_eval**

**Model Evaluation**

**Description**

Used for cross-validation. For a model of class "rqPen" it provides the average prediction error given the evaluation function of choice.

**Usage**

```r
model_eval(model, test_x, test_y, func="check",...)
```

**Arguments**

- **model**: Model of class "rqPen".
- **test_x**: Covariates used for prediction.
- **test_y**: Response to compare predictions against.
- **func**: Function used for evaluation. Options: "check" (Quantile Check), "SqErr" (Squared Error), "AE" (Absolute Value)
- **...**: Additional arguments to be sent to evaluation function. For instance check requires tau which is not part of the model_eval function.

**Value**

Mean of prediction errors using the choosen function.

**Author(s)**

Ben Sherwood

**Examples**

```r
x <- matrix(rnorm(800),ncol=8)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
l_model <- rq.lasso.fit(x,y,lambda=1)
nc_model <- rq.nc.fit(x,y,lambda=1)
newx <- matrix(rnorm(16),ncol=8)
newy <- 1 + newx[,1] - 3*newx[,5] + rnorm(2)
model_eval(l_model, newx, newy)
model_eval(l_model, newx, newy, func="SqErr")
model_eval(nc_model, newx, newy)
```
pos_part  

**Description**

Returns min(0, x)

**Usage**

```r
pos_part(x)
```

**Arguments**

- `x` Number to be evaluated

**Value**

min(0, x)

**Author(s)**

Ben Sherwood

**Examples**

```r
pos_part(5)
pos_part(-5)
```

---

predict.cv.rq.pen  

**Prediction from a cv quantile regression penalized model**

**Description**

Returns predicted values from "rqPen" model associated with lambda for "newx" covariates.

**Usage**

```r
## S3 method for class 'cv.rq.pen'
predict(object, newx, lambda,...)
```

**Arguments**

- `object` "cv.rq.pen" object.
- `newx` Matrix of covariates used for prediction.
- `lambda` Lambda associated with the model from which predictions should be made. Default is to use the lambda that provides the minimum criteria (CV or BIC) that was selected by cv.rq.pen.
- `...` Needed for consistency with generic predict.
**predict.rq.pen**

**Value**

Returns predicted values from the model for the selected lambda.

**Author(s)**

Ben Sherwood

**Examples**

```r
x <- matrix(rnorm(800), nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
cv_model <- cv.rq.pen(x, y)
n2 <- matrix(rnorm(100), nrow=10)
preds <- predict(cv_model, n2)
```

---

**predict.rq.pen**  
*Prediction from a quantile regression penalized model*

**Description**

Returns predicted values from "rq.pen" model for "newx" covariates.

**Usage**

```r
## S3 method for class 'rq.pen'
predict(object, newx,...)
```

**Arguments**

- **object**  
  "rq.pen" model.
- **newx**  
  Matrix of covariates used for prediction.
- **...**  
  Needed for consistency with generic predict.

**Value**

Returns predicted values from the model.

**Author(s)**

Ben Sherwood

**Examples**

```r
x <- matrix(rnorm(800), nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
lassoModel <- rq.lasso.fit(x, y, lambda=1)
n2 <- matrix(rnorm(100), nrow=10)
preds <- predict(lassoModel, n2)
```
**Description**

Quantile regression BIC with large p alternative as described in Lee, Noh and Park (2013).

**Usage**

\[
\text{qbic}(\text{model}, \text{largeP}=\text{FALSE})
\]

**Arguments**

- **model**: Model of class "rqPen".
- **largeP**: Large P version using an additional penalty factor of \(\log(s)\) where "s" is the total number of covariates considered.

**Value**

Numeric value representing BIC of selected model.

**Author(s)**

Ben Sherwood

**References**


**Examples**

```r
x <- matrix(rnorm(800), nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
l_model <- rq.lasso.fit(x,y, lambda=1)
nc_model <- rq.nc.fit(x,y, lambda=1)
qbic(l_model)
qbic(nc_model)
qbic(l_model, largeP=TRUE)
qbic(nc_model, largeP=TRUE)
```
randomly_assign  Randomly Assign

Description
Randomly assign n samples into k groups

Usage
randomly_assign(n,k)

Arguments
n  Number of samples.
k  Number of groups.

Value
A vector of length n having entries of 1,...,k.

Author(s)
Ben Sherwood

Examples
randomly_assign(37,5)
randomly_assign(11,3)

rq.lasso.fit  Quantile Regression with LASSO penalty

Description
Fits a quantile regression model with the LASSO penalty. Algorithm is similar to LASSO code presented in Koenker and Mizera (2014).

Usage
rq.lasso.fit(x,y,tau=.5,lambda=NULL,weights=NULL,
intercept=TRUE,coef.cutoff=.0000001,...)
Arguments

- **x**: Matrix of predictors.
- **y**: Vector of response values.
- **tau**: Conditional quantile being modeled.
- **lambda**: Tuning parameter.
- **weights**: Weights for the objective function.
- **intercept**: Whether model should include an intercept. Constant does not need to be included in "x".
- **coef.cutoff**: Coefficients below this value will be set to zero.
- **...**: Additional items to be sent to `rq`. Note this will have to be done carefully as `rq` is run on the augmented data to account for penalization and could provide strange results if this is not taken into account.

Value

Returns the following:

- **coefficients**: Coefficients from the penalized model.
- **PenRho**: Penalized objective function value.
- **residuals**: Residuals from the model.
- **rho**: Objective function evaluation without the penalty.
- **tau**: Conditional quantile being modeled.
- **n**: Sample size.

Author(s)

Ben Sherwood

References


Examples

```r
x <- matrix(rnorm(800),nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
lassoModel <- rq.lasso.fit(x,y,lambda=1)
```
Description

Produces penalized quantile regression models for a range of lambdas and penalty of choice. If lambda is unselected than an iterative algorithm is used to find a maximum lambda such that the penalty is large enough to produce an intercept only model. Then range of lambdas goes from the maximum lambda found to "eps" on the log scale. Local linear approximation approach used by Wang, Wu and Li to extend LLA as proposed by Zou and Li (2008) to the quantile regression setting.

Usage

rq.nc.fit(x,y,tau=.5,lambda=NULL,weights=NULL,intercept=TRUE, penalty="SCAD",a=3.7,iterations=10, converge_criteria=.0001,...)

Arguments

x          Matrix of predictors.
y          Vector of response values.
tau        Conditional quantile being modeled.
lambda     Tuning parameter.
weights     Weights for the objective function.
intercept   Whether model should include an intercept. Constant does not need to be included in "x".
penalty     MCP or SCAD.
a           Second tuning parameter.
iterations  Number of iterations to be done for iterative LLA algorithm.
converge_criteria  Difference in betas from iteration process that would satisfy convergence.
...         Additional items to be sent to rq.lasso.fit. LLA algorithm is used so rq.lasso.fit is used to approximate the non-convex objective function. See Wang, Wu and Li (2012) for details.

Value

Returns the following:

coefficients  Coefficients from the penalized model.
PenRho        Penalized objective function value.
residuals     Residuals from the model.
rho           Objective function evaluation without the penalty.
tau  Conditional quantile being modeled.
n  Sample size.
penalty  Penalty used, SCAD or MCP.

Author(s)
Ben Sherwood

References

Examples
```r
x <- matrix(rnorm(800), nrow=100)
y <- 1 + x[,1] - 3*x[,5] + rnorm(100)
scadModel <- rq.nc.fit(x,y,lambda=1)
```

scad  scad

Description
SCAD penalty function as described in Fan and Li (2001).

Usage
```r
scad(x,lambda,a)
```

Arguments
- **x**  Number to be evaluated
- **lambda**  Tuning parameter lambda
- **a**  Tuning parameter a

Value
SCAD penalty function with tuning parameters lambda and "a" evaluated at "x".

Author(s)
Ben Sherwood
scad_deriv

References

Examples
```r
scad(3,1)
scad(-3,1)
scad(.001,2)
```

---

### scad_deriv

**SCAD Derivative**

**Description**
Derivative of SCAD penalty function as described in Fan and Li (2001).

**Usage**
```r
scad_deriv(x, lambda, a)
```

**Arguments**
- `x`: Number to be evaluated
- `lambda`: Tuning parameter lambda
- `a`: Tuning parameter `a`. Default value of 3.7 as suggested in Fan and Li (2001)

**Value**
Derivative SCAD penalty function with tuning parameters lambda and "a" evaluated at "x".

**Author(s)**
Ben Sherwood

**References**

**Examples**
```r
scad_deriv(3,1)
scad_deriv(-3,1)
scad_deriv(.001,2)
```
square  

Square function

**Description**
Square value of a number

**Usage**
square(x)

**Arguments**

| x  | Number to be squared. |

**Value**

\[ x^2 \]

**Author(s)**
Ben Sherwood

**Examples**

\[
\begin{align*}
\text{square}(4) \\
\text{square}(-4) \\
\text{square}(2)
\end{align*}
\]
Index

beta_plots, 2
check, 3
cv.rq.pen, 4
cv_plots, 5
lasso, 6
mcp, 7
mcp_deriv, 8
model_eval, 9
pos_part, 10
predict.cv.rq.pen, 10
predict.rq.pen, 11
qbic, 12
randomly_assign, 13
rq.lasso.fit, 13
rq.nc.fit, 15
scad, 16
scad_deriv, 17
square, 18