# Package ‘nnlasso’

March 10, 2016

**Type** Package

**Title** Non-Negative Lasso and Elastic Net Penalized Generalized Linear Models

**Version** 0.3

**Date** 2016-03-09

**Author** Baidya Nath Mandal & Jun Ma

**Maintainer** Baidya Nath Mandal

**Depends** R (>= 3.2.2)

**Description** Estimates of coefficients of lasso penalized linear regression and generalized linear models subject to non-negativity constraints on the parameters using multiplicative iterative algorithm. Entire regularization path for a sequence of lambda values can be obtained. Functions are available for creating plots of regularization path, cross validation and estimating coefficients at a given lambda value. There is also provision for obtaining standard error of coefficient estimates.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-03-10 08:00:33

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>car</td>
<td>2</td>
</tr>
<tr>
<td>coef.nnlasso</td>
<td>3</td>
</tr>
<tr>
<td>cv.nnlasso</td>
<td>4</td>
</tr>
<tr>
<td>nnlasso</td>
<td>5</td>
</tr>
<tr>
<td>plot.nnlasso</td>
<td>8</td>
</tr>
<tr>
<td>predict.nnlasso</td>
<td>9</td>
</tr>
</tbody>
</table>

**Index** 11
The car data

Description

Car data has 105 observations and 12 variables. All variables except the 12th are standardized such that mean of each of them is 0 and standard deviation is 1. First 10 variables are various characteristics of the cars. The 11th variable y is the price. The 12th variable is a binary variable.

Usage

data("car")

Format

A data frame with 105 observations on the following 12 variables.

- Weight: weights of the cars
- Length: overall length
- Wheel.base: length of wheelbase
- Width: width of car
- Frt.Leg.Room: maximum front leg room
- Front.Hd: distance between the car's head-liner and the head of a 5 ft. 9 in. front seat passenger
- Turning: the radius of the turning circle
- Disp: engine displacement
- HP: net horsepower
- Tank: fuel refill capacity
- y: price
- y1: High or low price

Details

The data is created from car90 data of rpart package with selected 11 variables. The selected variables are Weight, Length, Wheel.base, Width, Frt.Leg.Room, Front.Hd, Turning, Disp, HP, Tank, Price. All these variables are standardized such that each of them has mean 0 and standard deviation 1. Price variable has been renamed as y. The variable y1 is a dichotomous variable created from that the data such that if price >=25000, then y1=1 else y1=0. Only complete cases are considered, so the data has 105 observations in place of 111 observations in car90 data set.

Source


Examples

data(car)
coef.nnlasso

Extract coefficients from a fitted nnlasso object

Description
The function returns the coefficients from a fitted nnlasso object.

Usage

## S3 method for class 'nnlasso'
coef(object,...)

Arguments

object     A 'nnlasso' object obtained using 'nnlasso' function.
...         Not used

Value

Estimated coefficients for different lambdas starting from maximum value of lambda to minimum value of lambda.

Author(s)
Baidya Nath Mandal and Jun Ma

References

Mandal, B.N. and Ma, J. (2016). L1 regularized multiplicative iterative path algorithm for non-negative generalized linear models

Examples

data(car)
attach(car)
x=car[,1:10]
g1=nnlasso(x,y1,family="binomial")
coef(g1)
g1=nnlasso(x,y,family="normal")
coef(g1)
detach(car)
cv.nnlasso  

k-fold cross validation for penalized generalized linear models for normal/binomial/poisson family

Description

The function does k-fold cross validation for selecting best value of regularization parameter.

Usage

```r
cv.nnlasso(x, y, family = c("binomial", "normal", "poisson"), k = 5, nlambda = 50, tau = 1, plot = TRUE, errorbars = TRUE)
```

Arguments

- **x**: x is matrix of order n x p where n is number of observations and p is number of predictor variables. Rows should represent observations and columns should represent predictor variables.
- **y**: y is a vector of response variable of order n x 1.
- **family**: family is either "normal" or "binomial" or "poisson".
- **k**: Number of folds for cross validation. Default is k=5.
- **nlambda**: Number of lambda values to be used for cross validation. Default is nlambda=50.
- **tau**: Elastic net parameter, 0 ≤ τ ≤ 1 in elastic net penalty \( \lambda \tau |\beta|_1 + (1 - \tau)|\beta|_2^2 \). Default tau=1 corresponds to LASSO penalty.
- **plot**: if TRUE, produces a plot of cross validated prediction mean squared errors/deviances against lambda. Default is TRUE.
- **errorbars**: If TRUE, error bars are drawn in the plot. Default is TRUE.

Value

Produces a plot and returns a list with following components:

- **lambda**: Value of lambda for which average cross validation error is minimum
- **pmse**: A vector of average cross validation errors for various lambda values
- **lambdas**: A vector of lambda values used in cross validation
- **se**: A vector containing standard errors of cross validation errors

Note

This function uses prediction means squared errors for normal family and deviance for binomial and poisson family.

Author(s)

Baidya Nath Mandal and Jun Ma
References


Examples

#normal family
data(car)
attach(car)
x=as.matrix(car[,1:10])
## not run: cv.nnlasso(x,y,family="normal",k=5)
#binomial family with same data
## not run: cv.nnlasso(x,y,family="binomial",k=5)
detach(car)

#poisson family
x=matrix(rnorm(100*30),100,30)
y=sample(c(1:5),100,replace=TRUE)
## not run: cv.nnlasso(x,y,family="poisson",k=5)

---------
nnlasso

Entire regularization path of non-negative penalized generalized linear model for normal/binomial/poisson family using multiplicative iterative algorithm

Description

The function computes coefficients of a penalized generalized linear model subject to non-negativity constraints for normal/binomial/poisson family using multiplicative iterative algorithm for a sequence of lambda values. Currently lasso and elastic net penalty are supported.

Usage

nnlasso(x,y,family=c("normal","binomial","poisson"),lambda=NULL,
intercept=TRUE,normalize=TRUE,tau=1,tol=1e-6,maxiter=1e5,nstep=100,min.lambda=1e-4,
eps=1e-6,path=TRUE,SE=FALSE)

Arguments

x x is matrix of order n x p where n is number of observations and p is number of predictor variables. Rows should represent observations and columns should represent predictor variables.

y y is a vector of response variable of order n x 1. y should follow either normal/binomial/poisson distribution.

family family should be one of these: "normal","binomial","poisson"

lambda The value of lambda for which coefficients are desired. The value of path must be FALSE in this case.
**Value**

An object of class ‘nnlasso’ with following components:

- **beta0**: A vector of order nstep of intercept estimates. Each value denote an estimate for a particular lambda. Corresponding lambda values are available in ‘lambdas’ element of the ‘nnlasso’ object.
- **coef**: A matrix of order nstep x p of slope estimates. Each row denotes solution for a particular lambda. Corresponding lambda values are available in ‘lambdas’ element of the ‘nnlasso’ object. Here p is number of predictor variables.
- **lambdas**: Sequence of lambda values for which coefficients are obtained.
- **L1norm**: L1norm of the coefficients.
- **norm.frac**: Fractions of norm computed as L1 norm at current lambda divided by maximum L1 norm.
- **lambda.iter**: Number of iterations used for different lambdas.
- **of.value**: Objective function values.
- **normx**: Norm of x variables.
- **se**: The standard errors of coefficient estimates.

**Author(s)**

Baidya Nath Mandal and Jun Ma
References


Examples

```r
# Non-negative LASSO
data(car)
attach(car)
x <- as.matrix(car[,1:10])
g1 <- nlasso(x,y,family="normal")
plot(g1)
plot(g1,xvar="lambda")

# Non-negative Elastic net with same data
## Not run:
g2 <- nlasso(x,y,family="normal",tau=0.6)
plot(g2)
plot(g2,xvar="lambda")

## End(Not run)

# Non-negative Ridge regression with same data
## Not run:
g3 <- nlasso(x,y,family="normal",tau=0)
plot(g3)
plot(g3,xvar="lambda")

## End(Not run)

# Non-negative L1 penalized GLM for binomial family
## Not run:
g1 <- nlasso(x,y,lambda=0.01,family="binomial")
plot(g1)
plot(g1,xvar="lambda")

## End(Not run)

# Non-negative Elastic net with GLM with binomial family
## Not run:
g2 <- nlasso(x,y,lambda=0.01,family="binomial",tau=0.8)
plot(g2)
plot(g2,xvar="lambda")

## End(Not run)

# Coefficient estimates for a particular lambda for normal family
cof <- nlasso(x,y,lambda=0.01,family="normal",path=FALSE,SE=TRUE)
coef(g1)
round(g1$se,3)
```
# coefficient estimates for a particular lambda for binomial family
## Not run:
g2=nnlasso(x,y,lambda=0.01,family="binomial",path=FALSE,SE=TRUE)
coef(g2)
round(g2$se,3)
detach(car)

## End(Not run)

---

**plot.nnlasso**  
*Plot of regularization path*

**Description**

Produces a plot of entire regularization path from a 'nnlasso' object obtained using 'nnlasso' function.

**Usage**

## S3 method for class 'nnlasso'

```r
plot(x,xvar=c("lambda","L1norm","fraction of norm"),...)
```

**Arguments**

- `x`: A 'nnlasso' object obtained using 'nnlasso' function.
- `xvar`: What should be on x-axis? `xvar="lambda"` produces a plot of regularization path with respect to lambda, `xvar="L1norm"` produces a plot of regularization path with respect to L1 norm of coefficients and `xvar="fraction of norm"` produces a plot of regularization path with respect to fraction of norm of coefficients. Default is `xvar="L1norm"`.
  - ...: Optional graphical parameters to matplot() function

**Value**

A plot of regularization path is produced.

**Author(s)**

Baidya Nath Mandal and Jun Ma

**References**

**Examples**

```r
data(car)
attach(car)
x <- as.matrix(car[, 1:10])
g1 <- nlasso(x, y, family = "normal")
plot(g1)
plot(g1, xvar = "lambda")
g1 <- nlasso(x, y1, family = "binomial")
plot(g1)
plot(g1, xvar = "lambda")
detach(car)
```

**predict.nnlasso**

*Prediction of coefficients of a penalized linear regression or generalized linear models*

**Description**

The function computes estimated coefficients value at a given lambda or L1 norm or fraction of norm using a ‘nnlasso’ object obtained using ‘nnlasso’ function.

**Usage**

```r
## S3 method for class 'nnlasso'
predict(object, mode = c("fraction", "norm", "lambda"), at = 0, ...)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>A ‘nnlasso’ object obtained using ‘nnlasso’ function.</td>
</tr>
<tr>
<td>mode</td>
<td>If mode=&quot;lambda&quot;, prediction is made for a given lambda, if mode=&quot;norm&quot;, prediction is made for a given L1 norm and if mode=&quot;fraction&quot;, prediction is made for a fraction of norm value. Default is mode=&quot;lambda&quot;</td>
</tr>
<tr>
<td>at</td>
<td>A value at which prediction is to be made. Default is at = 0.</td>
</tr>
<tr>
<td>...</td>
<td>Not used. Other arguments to predict.</td>
</tr>
</tbody>
</table>

**Value**

A vector of estimated coefficients of length p or p+1 at the given value of lambda or L1 norm or fraction of norm, depending on intercept=TRUE or FALSE in ‘nnlasso’ object. Here p is number of predictor variables.

**Author(s)**

Baidya Nath Mandal and Jun Ma
References


Examples

data(car)
attach(car)
x=as.matrix(car[,1:10])
g1=nnlasso(x,y,family="binomial")
predict(g1,mode="lambda",at=0.1)
predict(g1,mode="l1norm",at=1)
predict(g1,mode="fraction",at=0.5)
g1=nllasso(x,y,family="normal")
predict(g1,mode="lambda",at=0.09)
predict(g1,mode="l1norm",at=0.6)
predict(g1,mode="fraction",at=0.8)
detach(car)
Index

*Topic **Crossvalidation**
  cv.nnlasso, 4
*Topic **Elastic net**
  nnlasso, 5
  plot.nnlasso, 8
*Topic **Generalized linear models**
  plot.nnlasso, 8
*Topic **Generalized linear model**
  nnlasso, 5
*Topic **Gradient descent**
  plot.nnlasso, 8
*Topic **Jacobi**
  plot.nnlasso, 8
*Topic **L1 norm**
  plot.nnlasso, 8
*Topic **L1 penalty**
  nnlasso, 5
*Topic **LASSO**
  nnlasso, 5
  plot.nnlasso, 8
*Topic **Non-negative**
  nnlasso, 5
*Topic **Non-negativity**
  nnlasso, 5
  plot.nnlasso, 8
*Topic **Penalized**
  plot.nnlasso, 8
*Topic **Regression**
  nnlasso, 5
  plot.nnlasso, 8
*Topic **Regularization**
  nnlasso, 5
*Topic **Regularized**
  plot.nnlasso, 8
*Topic **Ridge regression**
  nnlasso, 5
*Topic **datasets**
  car, 2

car, 2