Package ‘plus’

February 20, 2015

Type Package

Version 1.0

Date 2012-05-28

Title Penalized Linear Unbiased Selection

Author Cun-Hui Zhang and Ofer Melnik, with contributions from Yi Yu and Stephanie Zhang

Maintainer Cun-Hui Zhang <czhang@stat.rutgers.edu>

Depends R (>= 2.10), grDevices, graphics, stats, utils

Description Efficient procedures for fitting an entire regression sequences with different model types.

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-05-28 18:23:56

NeedsCompilation no

R topics documented:

plot.plus *Plot method for plus objects*

Description

Produce a plot from a plus solution path.

Usage

```
## S3 method for class 'plus'
plot(x, xvar=c("lam","step"), yvar=c("coef","newy","lam","dim","R-sq"),
newx, step.interval, lam.interval, predictors, ...)
```

Arguments

`x`	a plus object
`xvar`	penalty level or plus step as the variable for the horizontal axis in the plot. Default is "lam".
`yvar`	paths of coefficients, predictions, penalty level, the number of nonzero coefficients or R-square as the variable for the vertical axis in the plot. Default is "coef"
`newx`	x values at which the fit is required. If newx is not set and yvar is "newy", no plot is produced.
`step.interval`	lower and upper bounds of the x-axis in the plot when xvar is "step". Defult covers all steps in the computed path.
`lam.interval`	lower and upper bounds of the x-axis in the plot when xvar is "lam". Default covers all penalty levels in the computed path.
`predictors`	a subset of predictors for which coefficients are plotted. Default is the entire set of predictors.
`...`	Additonal arguments for generic methods

Details

The fitted coefficients and penalty levels are linear between two consecutive turning points in the plus path so that exact values of "coef", "newy" and "lam" are ploted when xvar is set as "step". For concave penalties, the solution path is not necessarily monotone in penalty level. Since the extracted coefficients for a particular given penalty level is defined as the first point at which the solution path hits the given penalty level, the "coef" and "newy" plotted as approximations as the linear interpolation of their actual values at specifiec lam when xvar is set as "lam".

Value

NULL

Author(s)

Cun-Hui Zhang and Ofer Melnik

References

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. Annals of Statistics 38, 894-942.

Examples

```
data(sp500)
attach(sp500)

x <- sp500.percent[,3: (dim(sp500.percent)[2])]
y <- sp500.percent[,1]

object <- plus(x,y,method="mc+")
plot(object)
detach(sp500)
```

`plus` *Fits linear regression with a quadratic spline penalty, including the Lasso, MC+ and SCAD.*

Description

The algorithm generates a piecewise linear path of coefficients and penalty levels as critical points of a penalized loss in linear regression, starting with zero coefficients for infinity penalty and ending with a least squares fit for zero penalty. It is an extension of the LARS algorithm from the absolute value penalty to quadratic spline penalties.

Usage

```
plus(x,y, method = c("lasso", "mc+", "scad", "general"), m=2, gamma,v,t,
   monitor=FALSE, normalize = TRUE, intercept = TRUE,
   Gram, use.Gram = FALSE, eps=1e-15, max.steps=500, lam)
```

Arguments

`x`	predictors, an n by p matrix with n > 1 and p > 1.
`y`	response, an n-vector with n > 1.
`method`	c("lasso", "mc+", "scad", "general"); the LASSO penalty is specified by m = 1, MC+ is specified by m = 2 and gamma > 0, SCAD by m = 3 and gamma > 1. A general quadratic penalty is specified by m-vectors v and t.
`m`	number of knots with a quadratic spline penalty: m = 1 for Lasso, m = 2 for MC+, m = 3 for SCAD. Default is m = 2.
`gamma`	the largest knot of a quadratic spline penalty, say rho(.); gamma = 0 for lasso.

`v`	m-vector giving the negative second derivative rho(.) of the penalty between two knots or beyond gamma.
`t`	m-vector giving the discontinuities of the derivatives of the penalty function rho(.) as knots, including 0 as a knot.
`monitor`	If TRUE, plus prints out its progress when variables move in and out of the active set. Default is FALSE.
`normalize`	If TRUE, each variable is standardized to have unit mean squares, otherwise it is left alone. Default is TRUE.
`intercept`	If TRUE, an intercept is included in the model (and not penalized), otherwise no intercept is included. Default is TRUE.
`Gram`	The X'X matrix; useful for repeated runs (e.g. bootstrap) where a large X'X stays the same.
`use.Gram`	When p is very large, you may not want PLUS to precompute the entire Gram matrix. Default is FALSE.
`eps`	An effective zero.
`max.steps`	Limit the number of steps taken. Default is 500. There can be many more steps than n or p since variables can be removed and added as the algorithm proceeds. Users should check if the desired penalty level is reached if PLUS ends in the maximum step.
`lam`	A decreasing sequence of nonnegative numbers as penalty levels for which penalized estimates of coefficients are generated. Default is the vector of ordered penalty levels at the turning points of the computed path. If lam is set, the computation stops when the path first hits the minimum of lam. The scale of lam is determined by the penalized loss sum((y - x

Details

PLUS is described in detail in Zhang (2007). It computes a complete path of crititcal points of a penalised squared loss emcompassing from zero for infinite penalty to a lease squares fit for zero penalty, including possible multiple local minima for each penalty level.

Value

A "plus" object is returned, for which print, predict, coef and plot methods exist. In addition to arguments x, y, max.steps, and the used values of method, gamma and lam, the object contains the following items:

Some significant components of the object are:

`v`	matrix with rows as p-vectors indicating the parallelepipeds in which the computed path lives
`beta.path`	Tmatrix with rows as p-vectors of regression coefficients at the turning points of the solution path
`lam.path`	penalty levels at the turning points of the computed path. When the penalty function is concave, lam.path may not be a decreasing sequence but always takes nonnegative values.

beta	matrix with rows as p-vector of coefficients when the solution path first hits lam
lam	the specified penalty levels hit by lam.path. This may not be the same as argument lam if the minimum of the argument is not reached by the computed solution path.
dim	the number of nonzero beta
r.square	R-square values for beta
total.hits	length of output lam
total.steps	total number of steps executed, the same as the total number of segments in the computed solution path. With zero as the first coefficient vector, beta.path contains one more vector than total.steps.
full.path	TRUE if zero penalty is reached.
forced.stop	TRUE if PLUS is forced to stop due to reasons other than reaching max.steps or the minimum of argument lam.
singular.Q	TRUE if PLUS is forced to stop when a matrix is not invertible.

Author(s)

Cun-Hui Zhang and Ofer Melnik

References

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. Annals of Statistics 38, 894-942.

See Also

print, plot, and predict methods

Examples

```
data(sp500)
attach(sp500)
x <- sp500.percent[,3: (dim(sp500.percent)[2])]
y <- sp500.percent[,1]

par(mfrow=c(2,3))
object <- plus(x,y,method="lasso")
plot(object)
plot(object, yvar="dim")
plot(object, yvar="R-sq")
object <- plus(x,y,method="mc+")
plot(object)
plot(object, yvar="dim")
plot(object, yvar="R-sq")
detach(sp500)
```

`predict.plus` *Extract coefficients and make predictions from a plus solution path.*

Description

While plus() produces turning points in a path of critical points of a penalized squared loss, predict.plus extracts coefficients and make predictions at particular penalty levels by linear interpolation. The extracted coefficients are the same as these produced by plus() when the input lam is the same.

Usage

```
## S3 method for class 'plus'
predict(object, lam, newx, ...)
## S3 method for class 'plus'
coef(object, lam, ...)
```

Arguments

`object`	A fitted plus object
`lam`	A decreasing sequence of nonnegative numbers as penalty levels at which the coefficients are extracted and predictions are required. Default is the ordered values of lam.path generated by plus().
`newx`	x values at which the fit is required. If newx is not set, coefficients are extacted but predictions are not produced.
`...`	Additonal arguments for generic methods

Value

A list containing the following items:

`lambda`	penalty levels at which the coefficients and predicted values are extracted.
`coefficients`	extracted coefficients.
`dimension`	number of nonzero coefficients.
`r.square`	R-square as the ratio of the total centered residual sum of squares and the total centered sum of squares.
`step`	number of plus steps required to compute the coefficients.
`method`	including the LASSO, MC+, and SCAD.
`newy`	extracted predictions at newx; not produced if newx is not supplied.
`...`	Additonal arguments for generic methods

Author(s)

Cun-Hui Zhang and Ofer Melnik

References

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. Annals of Statistics 38, 894-942.

See Also

print, plot, plus

Examples

```
data(sp500)
attach(sp500)

x <- sp500.percent[,3: (dim(sp500.percent)[2])]
y <- sp500.percent[,1]

object <- plus(x,y,method="mc+")
## extract coefficients for the first 10 values of lam.path at values in x
extracted.values <- predict(object, lam = sort(object$lam.path[1:10],decreasing=TRUE), newx=object$x)
extracted.coef <- coef(object, lam = sort(object$lam.path[1:10],decreasing=TRUE))
detach(sp500)
```

---

print.plus *Print plus() steps when predictors are added or removed.*

Description

Print plus steps when predictors are added or removed, beginning with a description of the method such as LASSO, MC+ or SCAD.

Usage

```
## S3 method for class 'plus'
print(x, print.moves = 20, ...)
```

Arguments

`x`	a plus object
`print.moves`	the number of lines printed. Default is 20.
`...`	Additonal arguments for generic methods

Details

When a predictor is added or removed in the solution path, print one line providing the step number, the action, and the predicor involved.

Value

NULL

Author(s)

Cun-Hui Zhang and Ofer Melnik

References

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. Annals of Statistics 38, 894-942.

Examples

```
data(sp500)
attach(sp500)

x <- sp500.percent[,3: (dim(sp500.percent)[2])]
y <- sp500.percent[,1]

object <- plus(x,y,method="mc+")
print(object, print.moves=30)
detach(sp500)
```

---

sp500 *sp500*

Description

The sp500 datafile contains a year's worth of close-of-day data for most of the Standard and Poors 500 stocks. The data is in reverse chronological order, with the top row being Dec 31st, 2008.

Usage

```
sp500
```

Format

This data file contains the following items:

sp500.2008 The raw close-of-day data. The first column is of the DJIA index, the second is the S&P 500 index, the rest are individual labeled stocks.

sp500.diff The daily difference.

sp500.percent The daily percentage change.

Details

The goal can be to estimate either index using the individual stocks.

Source

This database was generated using data available http://finance.yahoo.com .

Index