Package ‘SSLASSO’

December 14, 2019

Version 1.2-2

Date 2019-12-13

Title The Spike-and-Slab LASSO

Author Veronika Rockova [aut,cre], Gemma Moran [aut]

Maintainer Gemma Moran <gm2918@columbia.edu>

Description Efficient coordinate ascent algorithm for fitting regularization paths for linear models penalized by Spike-and-Slab LASSO of Rockova and George (2018) <doi:10.1080/01621459.2016.1260469>.

URL http://faculty.chicagobooth.edu/veronika.rockova/ssl.pdf

Depends

Imports methods

License GPL-3

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-12-13 23:20:18 UTC

R topics documented:

plot.SSLASSO *Plot coefficients from a "SSLASSO" object*

Description

Produces a plot of the coefficient paths for a fitted "SSLASSO" object.

Usage

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

Arguments

x	Fitted "SSLASSO" model.
...	Other graphical parameters to plot.

Author(s)

Veronika Rockova <Veronika.Rockova@chicagobooth.edu>

References

Rockova, V. and George, E.I. (2018) The Spike-and-Slab LASSO. Journal of the American Statistical Association.

See Also

SSLASSO

Examples

```

## Linear regression, where p>n
library(SSLASSO)

n=100
p=1000
X=matrix(rnorm(n*p), n, p)
beta=c(1,2,3,rep(0,p-3))
Y=X[,1]*beta[1]+X[,2]*beta[2]+X[,3]*beta[3]+rnorm(n)
lambda1<-0.1
lambda0<-seq(lambda1,100,length=50)
theta<-0.5


# Separable penalty with fixed theta

result<-SSLASSO(X, Y,penalty="separable", variance = "fixed",
lambda1 = lambda1, lambda0 = lambda0,theta=theta)

plot(result)
```

SSLASSO *The Spike-and-Slab LASSO*

Description

Spike-and-Slab LASSO is a spike-and-slab refinement of the LASSO procedure, using a mixture of Laplace priors indexed by `lambda0` (spike) and `lambda1` (slab).

The `SSLASSO` procedure fits coefficients paths for Spike-and-Slab LASSO-penalized linear regression models over a grid of values for the regularization parameter `lambda0`. The code has been adapted from the `ncvreg` package (Breheny and Huang, 2011).

Usage

```
SSLASSO(X, y, penalty = c("adaptive", "separable"), variance = c("fixed", "unknown"),
lambda1, lambda0, nlambda = 100, theta = 0.5, sigma, a = 1, b,
eps = 0.001, max.iter = 500,  counter = 10, warn = FALSE)
```

Arguments

`X` The design matrix (n x p), without an intercept. `SSLASSO` standardizes the data by default.

`y` Vector of continuous responses (n x 1). The responses will be centered by default.

`penalty` The penalty to be applied to the model. Either "separable" (with a fixed `theta`) or "adaptive" (with a random `theta`, where `theta ~ B(a,p)`). The default is `"adaptive"`.

`variance` Whether the error variance is also estimated. Either "fixed" (with a fixed `sigma`) or "unknown" (with a random `sigma`, where `p(sigma) ~ 1/sigma`). The default is `"fixed"`.

`lambda1` Slab variance parameter. Needs to be less than `lambda0`. The default is `lambda0 = 1`.

`lambda0` Spike penalty parameters (L x 1). Either a numeric value for a single run (L=1) or a sequence of increasing values for dynamic posterior exploration. The default is `lambda0 = seq(1,nrow(X),length.out = 100)`.

`nlambda` The number of `lambda0` values. Default is 100.

`theta` Prior mixing proportion. For "separable" penalty, this value is fixed. For "adaptive" penalty, this value is used as a starting value.

`sigma` Error variance. For "fixed" variance, this value is fixed. For "unknown" variance, this value is used as a starting value.

`a` Hyperparameter of the beta prior `B(a,b)` for the adaptive penalty (default `a = 1`).

`b` Hyperparameter of the beta prior `B(a,b)` for the adaptive penalty (default `b = ncol(X)`).

`eps` Convergence criterion: converged when difference in regression coefficients is less than `eps` (default `eps = 0.001`).

`max.iter` Maximum number of iterations. Default is 500.

`counter` Applicable only for the adaptive penalty. Determines how often the parameter `theta` is updated throughout the cycles of coordinate ascent. Default is 10.

`warn` TRUE if warnings should be printed; FALSE by default

Details

The sequence of models indexed by the regularization parameter `lambda0` is fitted using a coordinate descent algorithm. The algorithm uses screening rules for discarding irrelevant predictors along the lines of Breheny (2011).

Value

An object with S3 class `"SSLASSO"` containing:

`beta` The fitted matrix of coefficients (p x L). The number of rows is equal to the number of coefficients `p`, and the number of columns is equal to `L` (the length of `lambda0`).

`intercept` A vector of length `L` containing the intercept for each value of `lambda0`. The intercept is `intercept = mean(y) -crossprod(XX,beta)`, where `XX` is the centered design matrix.

`iter` A vector of length `L` containing the number of iterations until convergence at each value of `lambda0`.

`lambda0` The sequence of regularization parameter values in the path.

`penalty` Same as above.

`thetas` A vector of length `L` containing the hyper-parameter values `theta` (the same as `theta` for "separable" penalty).

`sigmas` A vector of length `L` containing the values `sigma` (the same as the initial `sigma` for "known" variance).

`select` A (p x L) binary matrix indicating which variables were selected along the solution path.

`model` A single model chosen after the stabilization of the regularization path.

Author(s)

Veronika Rockova <Veronika.Rockova@chicagobooth.edu>, Gemma Moran <gmoran@wharton.upenn.edu>

References

Rockova, V. and George, E.I. (2018) The Spike-and-Slab LASSO. Journal of the American Statistical Association.

Moran, G., Rockova, V. and George, E.I. (2018) On variance estimation for Bayesian variable selection. <https://arxiv.org/abs/1801.03019>

See Also

plot.SSLASSO

Examples

```
## Linear regression, where p > n

library(SSLASSO)

p <- 1000
n <- 100

X <- matrix(rnorm(n*p), nrow = n, ncol = p)
beta <- c(1, 2, 3, rep(0, p-3))
y = X[,1] * beta[1] + X[,2] * beta[2] + X[,3] * beta[3] + rnorm(n)

# Oracle SSLASSO with known variance

result1 <- SSLASSO(X, y, penalty = "separable", theta = 3/p)
plot(result1)

# Adaptive SSLASSO with known variance

result2 <- SSLASSO(X, y)
plot(result2)

# Adaptive SSLASSO with unknown variance

result3 <- SSLASSO(X, y, variance = "unknown")
plot(result3)
```

standard *Standardizes a design matrix*

Description

The function std accepts a design matrix and returns a standardized version of that matrix (i.e., each column will have mean 0 and mean sum of squares equal to 1). The code has been adapted from the ncvreg package (Breheny and Huang, 2011).

Usage

```
standard(X)
```

Arguments

X	A matrix (or object that can be coerced to a matrix, such as a data frame).

Details

This function centers and scales each column of `X` so that

$$\sum_{i=1}^{n} x_{ij} = 0$$

and

$$\sum_{i=1}^{n} x_{ij}^2 = n$$

for all j. This is usually not necessary to call directly, as `SSLASSO` internally standardizes the design matrix, but inspection of the standardized design matrix can sometimes be useful. This differs from the base R function `scale` in two ways: (1) `scale` uses the sample standard deviation `sqrt(sum(x^2)/(n-1))`, while `std` uses the root-mean-square, or population, standard deviation `sqrt(mean(sum(x^2)))`, and (2) `std` is faster. The reason for using the population standard deviation is that `SSLASSO` assumes that the columns of the design matrix have been scaled to have norm `sqrt(n)`.

Value

The standardized design matrix, with attribues "center" and "scale" corresponding to the mean and (population) standard deviation used to scale the columns.

Author(s)

Patrick Breheny

Examples

```
X <- matrix(rnorm(50), 10, 5)
S <- standard(X)
apply(S, 2, sum)
apply(S, 2, function(x) mean(x^2))
```

Index