Package ‘SSLASSO’

August 28, 2018

Version 1.2-1

Date 2018-08-28

Title The Spike-and-Slab LASSO

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

Maintainer Gemma Moran <gmoran@wharton.upenn.edu>


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

Depends

License GPL-3

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-08-28 17:54:27 UTC

R topics documented:

plot.SSLASSO .................................................. 2
SSLASSO ..................................................... 3
standard ....................................................... 5

Index 7
plot.SSLASSO

Plot coefficients from a "SSLASSO" object

Description

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

Usage

```r
## 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


See Also

SSLASSO

Examples

```r
## 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))
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",
```

```r
```
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 \( \lambda_0 \) (spike) and \( \lambda_1 \) (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 \( \lambda_0 \). 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 = 1, a = 1, b,
eps = 0.001, max.iter = 500, counter = 10, warn = FALSE)
```

Arguments

- **x**: The design matrix \((n \times p)\), without an intercept. SSLASSO standardizes the data by default.
- **y**: Vector of continuous responses \((n \times 1)\). 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 \( \theta \sim \text{Beta}(a, p) \)). 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 \( p(\sigma) \sim 1/\sigma \)). The default is "fixed".
- **lambda1**: Slab variance parameter. Needs to be less than \( \lambda_0 \). The default is \( \lambda_0 = 1 \).
- **lambda0**: Spike penalty parameters \((L \times 1)\). Either a numeric value for a single run \((L=1)\) or a sequence of increasing values for dynamic posterior exploration. The default is \( \lambda_0 = \text{seq}(1, nrow(x), length.out = 100) \).
- **nlambda**: The number of \( \lambda_0 \) 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 \( \text{Beta}(a, b) \) 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 $\text{eps}$ (default $\text{eps} = 0.001$).

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 $\lambda_0$ 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 ($p \times L$). The number of rows is equal to the number of coefficients $p$, and the number of columns is equal to $L$ (the length of $\lambda_0$).
- **intercept**: A vector of length $L$ containing the intercept for each value of $\lambda_0$. The intercept is $\text{intercept} = \text{mean}(y) - \text{crossprod}(XX, beta)$, where $XX$ is the centered design matrix.
- **iter**: A vector of length $L$ containing the number of iterations until convergence at each value of $\lambda_0$.
- **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 ($p \times L$) 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


**standard**

**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).

---

---

**See Also**

`plot.SSLASSO`

**Examples**

```r
## 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))

# 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)
```
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 \( \sqrt{\text{sum}(x^2)/(n-1)} \), while `std` uses the root-mean-square, or population, standard deviation \( \sqrt{\text{mean}(\text{sum}(x^2))} \), 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 attributes "center" and "scale" corresponding to the mean and (population) standard deviation used to scale the columns.

Author(s)

Patrick Breheny

Examples

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

*Topic models
plot.SSLASSO, 2

*Topic regression
plot.SSLASSO, 2

plot.SSLASSO, 2, 5

scale, 6
SSLASSO, 2, 3
standard, 5