Package ‘SSLASSO’

September 21, 2017

Version 0.0-1
Date 2017-09-21
Title The Spike-and-Slab LASSO
Author Veronika Rockova [aut,cre], Gemma Moran [aut]
Maintainer Veronika Rockova <veronika.rockova@chicagobooth.edu>
Description Efficient algorithms for fitting regularization paths for linear models penalized by Spike-and-Slab LASSO.
URL http://faculty.chicagobooth.edu/veronika.rockova/ssl.pdf
Depends
License GPL-3
RoxygenNote 6.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-09-21 20:43:31 UTC

R topics documented:

plot.SSLASSO .............................................. 1
SSLASSO ...................................................... 3

Index

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.
...

Author(s)

Veronika Rockova <Veronika.Rockova@chicagobooth.edu>

References


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))
lambda1<-0.1
lambda0<-seq(lambda1,100,length=50)
theta<-0.5

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

plot(result)
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("separable", "adaptive"), variance = c("known", "unknown"), lambda1, lambda0, theta = 0.5, sigma = 1, a = 1, b = p, nu = 0, xi = 0, 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 (mean zero and variance $n$).
- **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 $\theta$), "adaptive" (with a random $\theta$, where $\theta \sim B(a, p)$).
- **variance**: Whether the error variance is also estimated. Either "known" (with a fixed $\sigma$) or "unknown" (with a random $\sigma$, where $\sigma \sim IG(nu, xi)$).
- **lambda1**: Slab variance parameter. Needs to be greater than $\lambda_0$.
- **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.
- **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 "known" 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 $b=p$).
- **nu**: Hyperparameter of the inverse-gamma prior $IG(nu, xi)$ for the unknown variance case (default $nu = 0$).
- **xi**: Hyperparameter of the inverse-gamma prior $IG(nu, xi)$ for the unknown variance case (default $xi = 0$).
- **eps**: Optional. Convergence criterion: converged when difference in regression coefficients is less than $\epsilon$ (default $\epsilon = 0.001$).
- **max.iter**: Optional. Maximum number of iterations. Default is 1000.
counter

Optional. 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 Breheyn (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 $\lambda_0$).

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 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>

References


See Also

plot.SSLASSO
Examples

## Linear regression, where \( p > n \)

```r
library(SSLASSO)

p <- 1000
n <- 100

lambda1 <- 1 # slab penalty for Spike-and-Slab LASSO
lambda0 <- seq(lambda1, 50, length=20) # slab penalties for Spike-and-Slab LASSO
l <- length(lambda0)
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", variance="known",
                   lambda1 = lambda1, lambda0 = lambda0,
                   theta = 3/p)
plot(result1)

# Adaptive SSLASSO with known variance
result2 <- sslasso(X, y, penalty="adaptive",
                   lambda1 = lambda1, lambda0 = lambda0,
                   theta = 0.5, a = 1, b = p, counter = 10)
plot(result2)

# Adaptive SSLASSO with unknown variance
result3 <- sslasso(X, y, penalty="adaptive", variance="unknown",
                   lambda1 = lambda1, lambda0 = lambda0)
plot(result3)
result3$sigmas[l]
```
Index

* Topic models
  plot.SSLASSO, 1

* Topic regression
  plot.SSLASSO, 1

plot.SSLASSO, 1, 4

SSLASSO, 2, 3