Package 'SSLASSO'

November 4, 2025

Version 1.2.3

Date 2025-10-16

Title The Spike-and-Slab LASSO

Maintainer Gemma Moran <gm845@stat.rutgers.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 https://doi.org/10.1080/01621459.2016.1260469

Imports stats, graphics

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation yes

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

Repository CRAN

Date/Publication 2025-11-04 19:00:13 UTC

Contents

---

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

Description

Produces a coefficient profile plot of the SSLASSO solution path.

Usage

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

Arguments

x	An object of class "SSLASSO", usually, a result of a call to SSLASSO.
...	Other arguments to be passed to plot.

Value

No return value, called for side effects (produces a plot of the coefficient paths).

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,
  beta.init = numeric(ncol(X)),
  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 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)').
variance	Specifies whether the error variance is "fixed" or "unknown".
lambda1	The slab penalty parameter. Must be smaller than the smallest 'lambda0'.
lambda0	A sequence of spike penalty parameters. Must be monotone increasing. If not specified, a default sequence is generated.
beta.init	Initial values for the coefficients. Defaults to a vector of zeros.
nlambda	The number of 'lambda0' values to use in the default sequence. Defaults to 100.
theta	The initial mixing proportion for the spike component. Defaults to 0.5.
sigma	The initial value for the error standard deviation. Defaults to 1.
a	Hyperparameter for the Beta prior on theta. Defaults to 1.
b	Hyperparameter for the Beta prior on theta. Defaults to the number of predictors, p.
eps	Convergence tolerance. The algorithm stops when the maximum change in coefficients is less than 'eps'. Defaults to 0.001.
max.iter	The maximum number of iterations. Defaults to 500.
counter	The number of iterations between updates of the adaptive penalty parameters. Defaults to 10.
warn	A logical value indicating whether to issue a warning if the algorithm fails to converge. Defaults to 'FALSE'.

Value

An object with S3 class "SSLASSO". The object contains:

beta	A p x L matrix of estimated coefficients, where L is the number of regularization parameter values.
intercept	A vector of length L containing the intercept terms.
iter	The number of iterations for each value of 'lambda0'.
lambda0	The sequence of 'lambda0' values used.

lambda1	The 'lambda1' value used.
penalty	The penalty type used.
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.
n	The number of observations.

Author(s)

Veronika Rockova <Veronika.Rockova@chicagobooth.edu>, Gemma Moran <gm845@stat.rutgers.edu>

References

Ročková, V., & George, E. I. (2018). The spike-and-slab lasso. Journal of the American Statistical Association, 113(521), 431-444.

Moran, G. E., Ročková, V., & George, E. I. (2019). Variance prior forms for high-dimensional bayesian variable selection. Bayesian Analysis, 14(4), 1091-1119.

See Also

[plot.SSLASSO()]

Examples

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

library(SSLASSO)

p <- 100
n <- 50

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 'standard' 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 'standard' uses the root-mean-square, or population, standard deviation 'sqrt(mean(sum(x^2)))', and (2) 'standard' 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

A standardized matrix with attributes for center, scale, and non-singular columns.

Index