Package 'SLOPE'

November 3, 2015

Type Package

Title Sorted L1 Penalized Estimation (SLOPE)

Version 0.1.3

Date 2015-04-27

Author Malgorzata Bogdan, Ewout van den Berg, Chiara Sabatti, Weijie Su, Emmanuel Candes, Evan Patterson

Maintainer Evan Patterson <epatters@stanford.edu>

Description Efficient procedures for Sorted L1 Penalized Estimation (SLOPE). The sorted L1 norm is useful for statistical estimation and testing, particularly for variable selection in the linear model.

License GPL-3

LinkingTo Rcpp

Imports Rcpp

Suggests knitr, testthat, isotone, R.matlab

VignetteBuilder knitr

RoxygenNote 5.0.0

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-11-03 10:54:06

R topics documented:

create_lambda *Lambda sequences for SLOPE*

Description

Computes λ sequences for SLOPE according to several pre-defined methods.

Usage

```
create_lambda(n, p, fdr = 0.2, method = c("bhq", "gaussian"))
```

Arguments

n	number of observations
p	number of variables
fdr	target False Discovery Rate (FDR)
method	method to use for computing λ (see Details)

Details

The following methods for computing λ are supported:

- bhq: Computes sequence inspired by Benjamini-Hochberg (BHq) procedure
- gaussian: Computes modified BHq sequence inspired by Gaussian designs

prox_sorted_L1 *Prox for sorted L1 norm*

Description

Compute the prox for the sorted L1 norm. That is, given a vector x and a decreasing vector λ, compute the unique value of y minimizing

$$\frac{1}{2}\|x - y\|_2^2 + \sum_{i=1}^{n} \lambda_i |x|_{(i)}.$$

Usage

```
prox_sorted_L1(x, lambda, method = c("c", "isotone"))
```

Arguments

x	input vector
lambda	vector of λ's, sorted in decreasing order
method	underlying prox implementation, either 'c' or 'isotone' (see Details)

Details

At present, two methods for computing the sorted L1 prox are supported. By default, we use a fast custom C implementation. Since SLOPE can be viewed as an isotonic regression problem, the prox can also be computed using the `isotone` package. This option is provided primarily for testing.

`SLOPE` *SLOPE: Sorted L1 Penalized Estimation*

Description

Performs variable selection using SLOPE (Sorted L1 Penalized Estimation). Given a design matrix X and a response vector y, find the coefficient vector β minimizing

$$\frac{1}{2}\|X\beta - y\|_2^2 + \sigma \cdot \sum_{i=1}^{p} \lambda_i |\beta|_{(i)},$$

where the λ sequence is chosen to control the false discovery rate associated with nonzero components of β.

Usage

```
SLOPE(X, y, fdr = 0.2, lambda = "gaussian", sigma = NULL,
  normalize = TRUE, solver = c("default", "matlab"), ...)
```

Arguments

`X`	the n-by-p design matrix
`y`	response vector of length n
`fdr`	target FDR (false discovery rate)
`lambda`	specifcation of λ, either one of "bhq" or "gaussian", or a vector of length p, sorted in decreasing order (see `create_lambda`)
`sigma`	noise level. If omitted, estimated from the data (see Details).
`normalize`	whether to center the input data and re-scale the columns of the design matrix to have unit norm. Do not disable this unless you are certain that your data is appropriately pre-processed.
`solver`	which SLOPE solver to use (see Details)
`...`	additional arguments to pass to the solver (see the relevant solver)

Details

At present, two solvers for the SLOPE problem are supported. By default, we use SLOPE_solver, which is mostly written in R but uses a fast prox implemented in C. If you have MATLAB installed, it is also possible to use the TFOCS solver for SLOPE. This requires the MATLAB package TFOCS and the R package R.matlab.

If the noise level is unknown, it is estimated from the data using one of two methods. When n is large enough compared to p, the classical unbiased estimate of σ^2 is used. Otherwise, the *iterative SLOPE* algorithm is used, in which a decreasing sequence of σ^2 estimates is used until the set of selected variables stabilizes. For details, see Section 3.2.3 of the SLOPE paper.

Value

An object of class SLOPE.result. This object is a list containing at least the following components:

lambda	the λ sequence used
lambda_method	method of λ construction ("bhq", "gaussian", or "user")
sigma	(sequence of) noise level(s) used
beta	optimized coefficient vector β
selected	selected variables (variables i with $\beta_i > 0$)

See Also

SLOPE_solver

SLOPE_solver *Sorted L1 solver*

Description

Solves the sorted L1 penalized regression problem: given a matrix A, a vector b, and a decreasing vector λ, find the vector x minimizing

$$\frac{1}{2}\|Ax - b\|_2^2 + \sum_{i=1}^{p} \lambda_i |x|_{(i)}.$$

Usage

```
SLOPE_solver(A, b, lambda, initial = NULL, prox = prox_sorted_L1,
  max_iter = 10000, grad_iter = 20, opt_iter = 1, tol_infeas = 1e-06,
  tol_rel_gap = 1e-06)
```

Arguments

A	an n-by-p matrix
b	vector of length n
lambda	vector of length p, sorted in decreasing order
initial	initial guess for x
prox	function that computes the sorted L1 prox
max_iter	maximum number of iterations in the gradient descent
grad_iter	number of iterations between gradient updates
opt_iter	number of iterations between checks for optimality
tol_infeas	tolerance for infeasibility
tol_rel_gap	tolerance for relative gap between primal and dual problems

Details

This optimization problem is convex and is solved using an accelerated proximal gradient descent method.

Value

An object of class SLOPE_solver.result. This object is a list containing at least the following components:

x	solution vector x
optimal	logical: whether the solution is optimal
iter	number of iterations

Index