Package ‘constrlasso’

March 6, 2025

Title Contrained lasso regression and solution path in R as in [Gaines et al. (2018)]

(http //hua-zhou.github.io/SparseReg/). as in [Gaines et al. (2018)](http://hua-zhou.github.io/SparseReg/)

Version 0.1.0

Description
This package provides functions for fitting the constrained lasso regression and computing the solution path. The constrained lasso regression is a generalization of the lasso regression that allows for linear equality constraints on the coefficients. The solution path is computed using the algorithm proposed by Gaines et al. (2018) <doi:10.1080/10618600.2017.1302882>.

License GPL-2

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports stats, MASS, lpSolve, glmnet, ROI, ROI.plugin.qpoases, Rdpack

RdMacros Rdpack

NeedsCompilation no

Author Antoniya Shivarova [aut, cre]

Maintainer Antoniya Shivarova <an.shivarova@gmail.com>

Contents

`constrlasso` *The function constrlasso*

Description

This function fits a linearly constrained lasso regression, using a predictor matrix X, a response y and a tuning parameter value lambda. It results in a vector of coefficient estimates. The function corresponds to lsq_constrsparsereg of the SparseReg MATLAB-toolbox by Zhou and Gaines, see the project page. Only the Quadratic Programming Algorithm for ENET is implemented. For more information, see Gaines et al. (2018).

Usage

```
constrlasso(
  X,
  y,
  lambda,
  Aeq = NULL,
  beq = NULL,
  A = NULL,
  b = NULL,
  penidx = NULL,
  method = "QP"
)
```

Arguments

X	an nxp matrix of p regressors with n observations.
y	an nx1 response vector with n observations.
lambda	a tuning parameter value for the lasso penalty. Default value is lambda=0.
Aeq	a cxp equality constraint matrix, containing c constraints for p regressors. Default value is Aeq=NULL, no equality constraints.
beq	a cx1 equality constraint vector. Default value is beq=NULL, no equality constraints.
A	a cxp inequality constraint matrix, containing c constraints for p regressors. Default value is A=NULL, no inequality constraints.
b	a cx1 inequality constraint vector. Default value is b=NULL, no inequality constraints.
penidx	a logical px1 vector, indicating which coefficients are to be penalized. Default value is penidx=NULL and imposes penalty on all p coefficients.
method	a character string, the method to be used. Possible values are "QP" (default) for Quadratic Programming and "CD" for Coordinate Descent. "CD" uses the glmnet package and only works without equality and inequality constraints.

Value

betahat a px1 vector of estimated coefficients.

dual_eq equality duals. The function returns an empty vector for no equality constraints.

dual_neq inequality duals. The function returns an empty vector for no inequality constraints.

Details

The Constrained Lasso as in Gaines et al. (2018) minimizes

$$0.5||y - X\beta||_2^2 + \lambda||\beta||_1,$$

subject to $Aeq\beta = beq$ and $A\beta \leq b$.

References

Gaines BR, Kim J, Zhou H (2018). "Algorithms for fitting the constrained lasso." *Journal of Computational and Graphical Statistics*, **27**(4), 861–871.

Examples

```
set.seed(1234)
n <- 200 # number of observations
p <- 150 # number of regressors
real_p <- 50 # number of true predictors
Xmat <- matrix(rnorm(n * p), nrow = n, ncol = p)
yvec <- apply(Xmat[, 1:real_p], 1, sum) + rnorm(n)
results_reg_no_constr <- constrlasso(Xmat, yvec, lambda = 0)
```

---

constrlasso_path *The function constrlasso_path*

Description

This function performs a Constrained Lasso Solution Path as in Gaines et al. (2018). It computes the solution path for the constrained lasso problem, using a predictor matrix X and a response y. The constrained lasso solves the standard lasso Tibshirani (1996) subject to the linear equality constraints $Aeq\beta = beq$ and linear inequality constraints $A\beta \leq b$. The result lambda_path contains the values of the tuning parameter along the solution path and beta_path - the estimated regression coefficients for each value of lambda_path. The function corresponds to lsq_classopath of the SparseReg MATLAB-toolbox of by Zhou and Gaines, see the project page. For more information, see Gaines et al. (2018).

Usage

```
constrlasso_path(
  X,
  y,
  Aeq = NULL,
  beq = NULL,
  A = NULL,
  b = NULL,
  penidx = NULL,
  init_method = "QP",
  epsilon = 1e-04,
  stop_lambda_tol = 1e-07,
  ceiling_tol = 1e-10,
  zeros_tol = 1e-20,
  verbose = FALSE
)
```

Arguments

X	an nxp matrix with p regressors with n observations.
y	an nx1 response vector with n observations.
Aeq	a cxp equality constraint matrix, containing c constraints for p regressors. Default value is Aeq=NULL, no equality constraints.
beq	a cx1 equality constraint vector. Default value is beq=NULL, no equality constraints.

A	a cxp inequality constraint matrix, containing c constraints for p regressors. Default value is A=NULL, no inequality constraints.
b	a cx1 inequality constraint vector. Default value is b=NULL, no inequality constraints.
penidx	a logical px1 vector, indicating which coefficients are to be penalized. Default value is penidx=NULL and allows all p coefficients to be penalized.
init_method	a character string, the initializing method to be used. Possible values are "QP" (default) for Quadratic Programming and "LP" for Linear Programming. "LP" is recommended only when it's reasonable to assume that all coefficient estimates initialize at zero.
epsilon	a tuning parameter for ridge penalty in case of high-dimensional (n>p) regressors matrix X. Default value is 1e-4.
stop_lambda_tol	a tolerance value for the tuning lasso parameter. The algorithm stops when hitting this tolerance. Default value is 1e-7.
ceiling_tol	a tolerance value for the change in subgradients. Default value is 1e-10.
zeros_tol	a tolerance value for the zero equality of coefficients. Default value is 1e-20.
verbose	a logical parameter. TRUE prints along the constraint lasso solution path. Default value is FALSE.

Value

lambda_path a vector of the tuning parameter values along the solution path.

beta_path a matrix with estimated regression coefficients for each value of lambda_path

df_path a vector with degrees of freedom along the solution path

objval_path a vector with values of the objective function for each value of lambda_path

Details

The Constrained Lasso as in Gaines et al. (2018) minimizes

$$0.5||y - X\beta||_2^2 + \lambda||\beta||_1,$$

subject to $Aeq\beta = beq$ and $A\beta \leq b$.

References

Gaines BR, Kim J, Zhou H (2018). "Algorithms for fitting the constrained lasso." *Journal of Computational and Graphical Statistics*, **27**(4), 861–871.

Tibshirani R (1996). "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1), 267–288.

Examples

```
set.seed(1234)
n <- 200 # number of observations
p <- 150 # number of regressors
real_p <- 50 # number of true predictors
Xmat <- matrix(rnorm(n * p), nrow = n, ncol = p)
yvec <- apply(Xmat[, 1:real_p], 1, sum) + rnorm(n)
results_path <- constrlasso_path(Xmat, yvec)
```

Index