constrLasso

The package constrLasso includes a function for constrained lasso regression and a solution path algorithm as in Gaines et al. (2018).

Installation

You can install this version of the package from Github with:

```
install.packages("devtools")
library(devtools)
install_github("antshi/constrLasso")
library(constrLasso)
```

Constrained Lasso Regression

These are basic examples which show you how to use the function constrLassoReg. First, let's prepare with

```
# include the package
library(constrLasso)

# generate some data
library(MASS)
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)
```

Example 1

No constraints and no penalty.

```
resultsReg <- constrLassoReg(Xmat, yvec, lambda=0)
```

Example 2

Included constraints and penalty.

```
# build constraints

# equality constraints (sum constraint)
Aeq <- matrix(1, 1, p)
beq <- matrix(1, 1, 1)

# inequality constraints (upper and lower bounds set to 0.5 and -0.5, respectively)
A1 <- matrix(diag(1,p), p, p)
b1 <- matrix(0.5, p, 1)
A2 <- matrix(-diag(1,p), p, p)
b2 <- matrix(0.5, p, 1)
A <- rbind(A1,A2)
b <- rbind(b1,b2)

resultsReg_ConstrPen <- constrLassoReg(Xmat, yvec, Aeq=Aeq, beq=beq, lambda=3)
```

```r
sum(resultsReg_ConstrPen[[1]]) #should be equal to 1

resultsReg_ConstrPen2 <- constrLassoReg(Xmat, yvec, Aeq=Aeq, beq=beq, A=A, b=b, lambda=2)
sum(resultsReg_ConstrPen2[[1]]) #should be equal to 1
which(abs(resultsReg_ConstrPen2[[1]])>0.5) #should not contain entries
```

Constrained Lasso Solution Path

These are basic examples which show you how to use the function constrLassoPath. First, let's prepare with

```r
# include the package
library(constrLasso)

# generate some data
library(MASS)
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)
```

Example 1

No constraints

```r
resultsPath <- constrLassoPath(Xmat, yvec)
```

Example 2

Included constraints

```r
# build constraints

# equality constraints (sum constraint)
Aeq <- matrix(1, 1, p)
beq <- matrix(1, 1, 1)

# inequality constraints (upper and lower bounds set to 0.5 and -0.5, respectively)
A1 <- matrix(diag(1,p), p, p)
b1 <- matrix(0.5, p, 1)
A2 <- matrix(-diag(1,p), p, p)
b2 <- matrix(0.5, p, 1)
A <- rbind(A1,A2)
b <- rbind(b1,b2)

resultsPath_Constr <- constrLassoPath(Xmat, yvec, Aeq=Aeq, beq=beq)
apply(resultsPath_Constr[[1]], 2, sum) #should be equal to 1 along the path


resultsPath_Constr2 <- constrLassoPath(Xmat, yvec, Aeq=Aeq, beq=beq, A=A, b=b)
apply(resultsPath_Constr2[[1]], 2, sum)  #should be equal to 1 along the path
apply(resultsPath_Constr2[[1]], 2, function(x) which(abs(round(x,10))>0.5)) #should not contain entries
```