R:Lab (Lasso)

Nithya Dev

BS2018

We need to install the **lars**(least Angle Regression,Lasso and Forward Stagewise) package to perform lasso in R.

Code:

```
>x1=1:100
>x2=3*x1+rnorm(100)/10
>x3=x1*x1
>tmp=c()
>for(i in 1:100){
 y=1+2*x1+0.05*x3+4*rnorm(100)
tmp=rbind(tmp,lm(y~x1+x2+x3)$coef)}


>install.packages(lars)   #installing the package
>library(lars) #loading the package
>lmod=lars(cbind(x1,x2,x3),y)) #we need to design the matrix
>lmod
```

Output:

```
    Call:
lars(x = cbind(x1, x2, x3), y = y)
R-squared: 1
Sequence of LASSO moves:
     x3 x2 x1
Var   3  2  1
Step  1  2  3
```

We haven't got much information from this.So we use the **coef()** command

Code:

```
>coef(lmod)
```

Output:

```
                x1        x2         x3
[1,]  0.000000 0.0000000 0.00000000
[2,]  0.000000 0.0000000 0.03140557
[3,]  0.000000 0.6593039 0.05037904
[4,] -6.011976 2.6634341 0.05037455
```

Now lets understand the output we got.

This command is performing a constrained optimization.So we have a square over which we are trying to minimize the squared error loss.Depending on the size of λ,the size of the square will change.

If λ=0 then obviously estimator will be [0 0 0].The square has shrunk to origin,which is our first row.Now we increase λ and when we increase it sufficiently large, the square is now so large that our least square estimator is already inside the square.So constrained optimization is same as unconstrained optimization,as a result we are getting the least square estimator.The last row in our output is our least square estimator and the first row is always [0 0 0]

Code:

```
>lmod$lambda
```

From this command we will get the values of λ's at which the changes are occuring.

Output:

```
[1] 2.069175e+03 1.124120e+03 1.119665e-03
```

The row vector [0 0 0] changes to [0 0 0.0314557] when λ=1.11965e-03

The row vector [0 0 0.0314557] changes to [0.000000 0.6593039 0.05037904] when λ=1.124120e+03

The row vector [0.000000 0.6593039 0.05037904] changes to [-6.011976 2.6634341 0.05037455] when λ=2.069175e+03

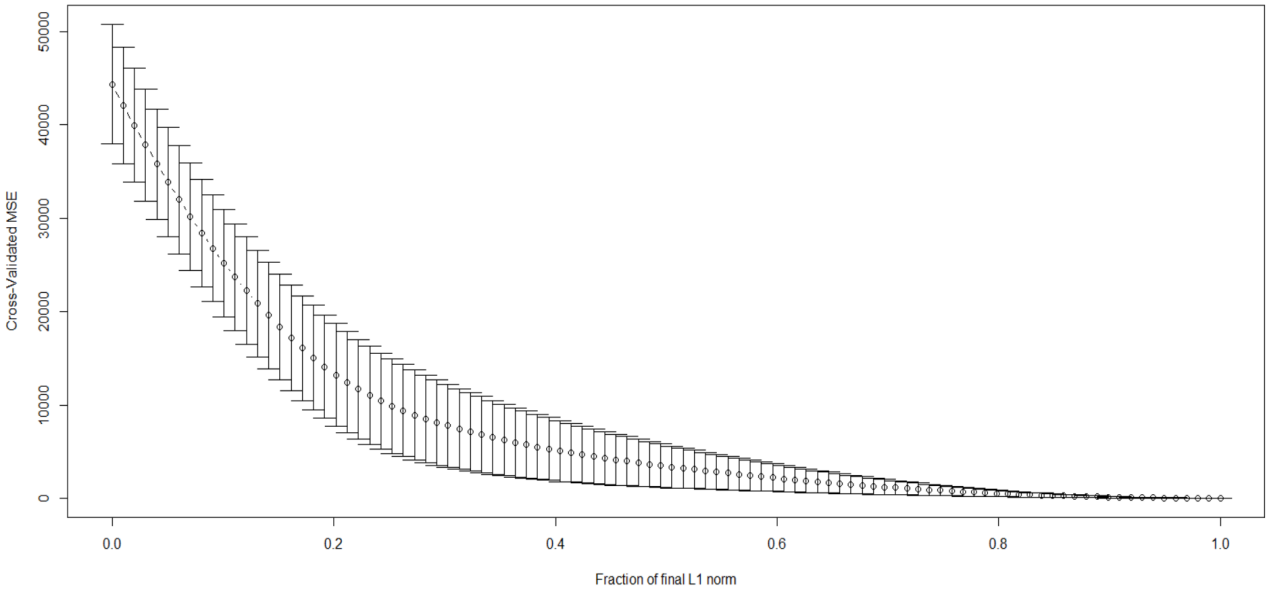
So how to choose those values of lambda?

For that we can perform the cv using the command **cv.lars()**

Code:

```
>cvlmod=cv.lars(cbind(x1,x2,x3),y)
```

Output:



We can see that when λ =0 then MSE is pretty high and as λ increases MSE is drastically coming down and then going pretty close to 0 which indicates that we must be having multicollinearity. The vertical parts around each point gives the standard error.

The point from which the curve starts stabilizing ,take it as the value of λ.Also the choice of λ is not that crucial because as we can see that if we take any value after that elbow point then MSE doesn't change much.