Lasso & Post-Lasso OLS

A comparison in different datasets

Jieteng Chen

December 15, 2019

1 Introduction

Lasso is a good tool for the estimation in sparse linear models and it minimizes the residual sum of square error subject to the sum of the absolute value of the coefficients being less than a constant. However, the estimates produced by Lasso are biased, which is the price for the sparse solution. While, *post-lasso OLS* applies *ordinary least squares*(OLS) to the model selected by the first-step lasso and and has the advantage of a smaller bias.(Belloni, A. and V. Chernozhukov. 2013). When the model is really sparse and lasso selects the true model perfectly, the *OLS post-lasso* estimator becomes the oracle estimator, which also means it can give us a more accurate prediction.

In this report, I simulate different dataset and compare the performance of Lasso and Post-Lasso OLS. The results tell me that if the model is sparse and lasso selection correctly includes all components of the true model as a subset, the performance of these two method are good, especially *Post-Lasso OLS*, which can get better predictions and more accurate estimation on the parameter. This finding is consistent with Belloni, A. and V. Chernozhukov.

2 Example One

Consider a high dimensional approximately sparse linear regression model:

$$y_i = x_i'\beta + \epsilon,$$

$$E[\epsilon_i x_i] = 0,$$

$$\beta \in R^P, i = 1, \ldots, n$$

where y_i and x_i is the ith observation, and $x_i = (x+_{i1}, x_{i2}, \ldots, x_{ip})$ is a p-dimensional vector. And β is also a p-dimensional vector which the first **s** elements are nonzero and the others are all zero. In this example, the the first 5 elements in β are 5 and the others are all zero.

```
library(hdm)
library(ggplot2)
set.seed(1234567)
n_train=80 # train sample size,samll
p=100 # number of variables
```

```r
s=5 # number of variables with non-zero true coefficients
X=matrix(rnorm(n_train*p),ncol = p) # train X
colnames(X)=paste("X",1:p,sep="")
beta=c(rep(5,s),rep(0,p-s)) # The parameter vector, very sparse
Y<-X%*%beta+rnorm(n_train,0,1)  # train Y
n_testing=10000# testing set size,large
X_testing=matrix(rnorm(n_testing*p),ncol = p) #testing X
Y_testing=X_testing%*%beta+rnorm(n_testing) # testing Y
```

After generating the train and testing data, we first use Lasso(without post) to estimate the paramaters and predict the $\hat{y}$ of the testing data.

```r
lasso.reg<-rlasso(Y~X,post = FALSE,intercept = FALSE)# pure lasso, not post-lasso
summary(lasso.reg,all = FALSE)
```

```
##
## Call:
## rlasso.formula(formula = Y ~ X, post = FALSE, intercept = FALSE)
##
## Post-Lasso Estimation:  FALSE
##
## Total number of variables: 100
## Number of selected variables: 13
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -2.542377 -0.568568  0.005816  0.580416  1.908033
##
##     Estimate
## X1     4.762
## X2     4.654
## X3     4.859
## X4     4.604
## X5     4.823
## X13    0.028
## X32   -0.001
## X36    0.016
## X47    0.056
## X49    0.029
## X54    0.008
## X56    0.037
## X81    0.009
##
## Residual standard error: 0.9825
## Multiple R-squared:  0.9921
## Adjusted R-squared:  0.9905
```

```
## Joint significance test:
##  the sup score statistic for joint significance test is 61.04 with a p-value of     0

yhat.lasso.testing=predict(lasso.reg,newdata = X_testing) # prediction in test set
```

We can see that Lasso gives us 13 nonzero estimates of the coefficients, although the true number of nonzero coefficients is only 5. Next, we focus on the Post-lasso OLS.

```
post.lasso.reg=rlasso(Y~X,post = TRUE,intercept = FALSE) # now use the post-lasso
summary(post.lasso.reg,all = FALSE)

##
## Call:
## rlasso.formula(formula = Y ~ X, post = TRUE, intercept = FALSE)
##
## Post-Lasso Estimation:  TRUE
##
## Total number of variables: 100
## Number of selected variables: 5
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -2.025340 -0.545029 -0.006518  0.497335  1.869807
##
##    Estimate
## X1    4.985
## X2    4.886
## X3    5.077
## X4    4.824
## X5    5.083
##
## Residual standard error: 0.8877
## Multiple R-squared:  0.9935
## Adjusted R-squared:  0.9931
## Joint significance test:
##  the sup score statistic for joint significance test is 61.04 with a p-value of 0.002

yhat.postlasso.testing=predict(post.lasso.reg,newdata = X_testing) # prediction in test set
```

Post-Lasso seems to give a more sparse result compared with rigorous lasso.It finds out the 5 nonzero coefficients exactly. Now, calculate the MSE and compare them.

```
MSE<-apply(cbind((Y_testing-yhat.lasso.testing)^2,(Y_testing-yhat.postlasso.testing)^2),2,mean)
names(MSE)<-c("lasso MSE","   post-lasso MSE")
print(MSE,digits = 4) # Mean Squared Error Value

##         lasso MSE    post-lasso MSE
##             1.404             1.073
```

We can see that Post-Lasso OLS can have a better performance and a more accurate prediction than Lasso in the term of MSE.

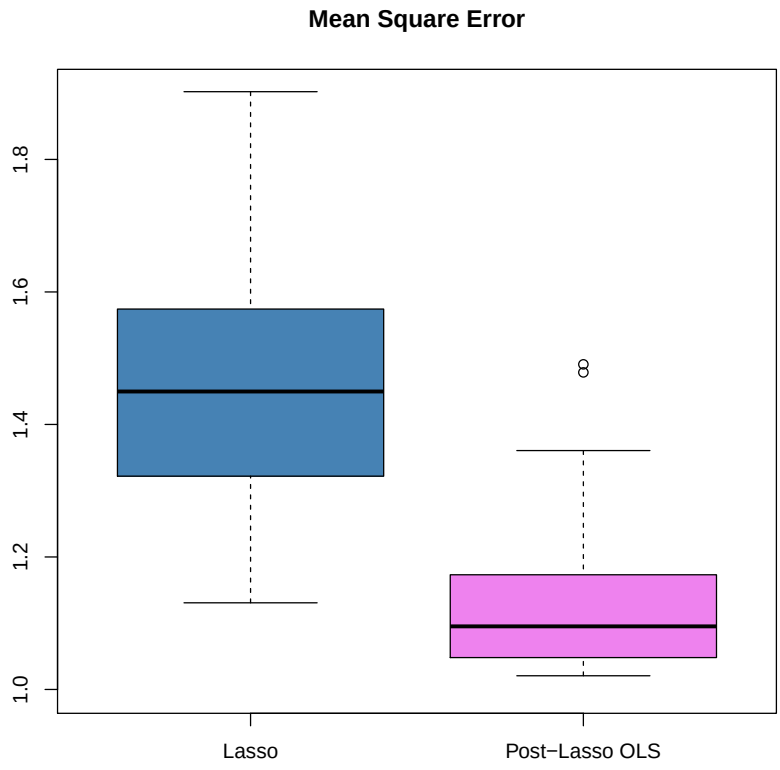
3 Example Two

Here I iterate the approach above 100 times and plot the MSE of Lasso and Post-Lasso OLS. To see whether Post-Lasso OLS is really better than Lasso.Because the coefficients do not have exact meaning and I just show the distribution of β_1 and β_3.

```
n_ite=100# Number of iteration
MSE.lasso=rep(0,n_ite) # MSE of Lasso
MSE.Postlasso=rep(0,n_ite) # MSE of Post-Lasso OLS
beta.lasso.1=rep(0,n_ite) # beta 1 estimated by Lasso
beta.postlasso.1=rep(0,n_ite) # beta 1 estimated by Post-Lasso OLS
beta.lasso.3=rep(0,n_ite) # beta 1 estimated by Lasso
beta.postlasso.3=rep(0,n_ite) # beta 1 estimated by Post-Lasso OLS
# I use the same testing set as in example One, but generate train set every round.
for (i in 1:n_ite){
  X.train=matrix(rnorm(n_train*p),ncol = p)
  colnames(X.train)=paste("X",1:p,sep="")
  Y.train<-X.train%*%beta+rnorm(n_train,0,1)
  lasso.reg<-rlasso(Y.train~X.train,post = FALSE,intercept = FALSE)
  yhat.lasso.testing=predict(lasso.reg,newdata = X_testing)
  beta.lasso.3[i]=lasso.reg$coefficients["X3"]
  beta.postlasso.3[i]=post.lasso.reg$coefficients["X3"]
  beta.lasso.1[i]=lasso.reg$coefficients["X1"]
  beta.postlasso.1[i]=post.lasso.reg$coefficients["X1"]
  post.lasso.reg=rlasso(Y.train~X.train,post = TRUE,intercept = FALSE)
  yhat.postlasso.testing=predict(post.lasso.reg,newdata = X_testing)
  MSE.lasso[i]=mean((Y_testing-yhat.lasso.testing)^2)
  MSE.Postlasso[i]=mean((Y_testing-yhat.postlasso.testing)^2)
}
```

Then we can plot MSE.

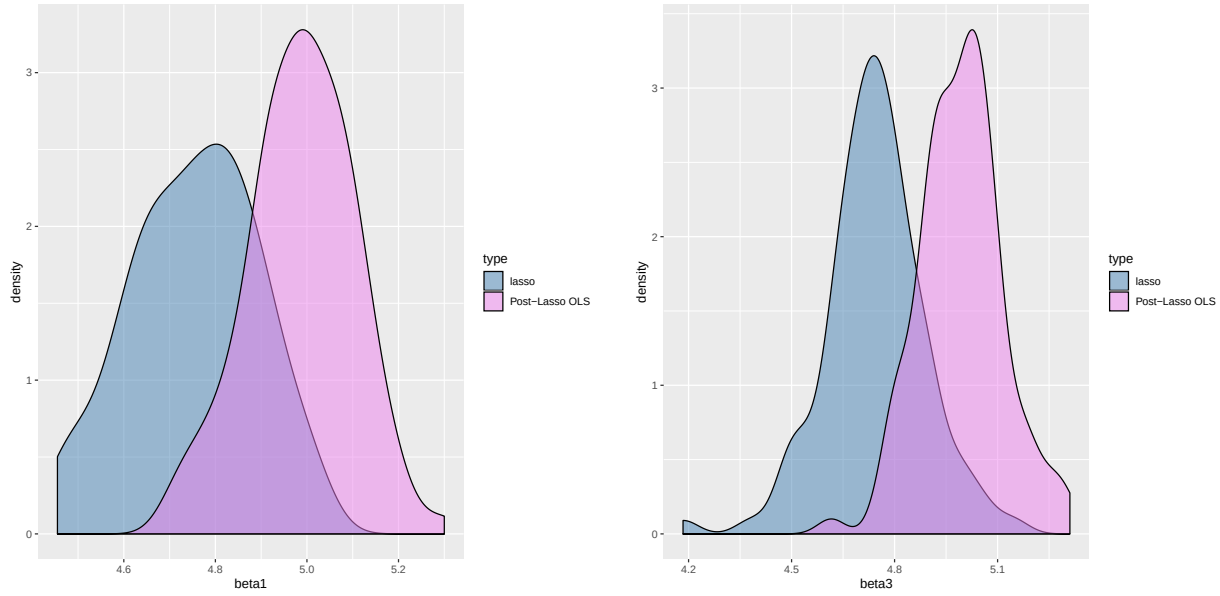
```
# Plot the distribution of the Mean Square Errors of these two regression methods
MSE=data.frame(cbind(MSE.lasso,MSE.Postlasso))
colnames(MSE)=c("Lasso","Post-Lasso OLS")
boxplot(MSE,col = c("steelblue","violet"),main="Mean Square Error",ylim=c(1,1.9))
```



It is obvious that the overall MSE of Post-Lasso OLS is smaller than Lasso, which means the Post One can have a better prediction power. Because the nonzero coefficients are homogeneous and don't have any exact meaning, so we can just focus on the first and third estimates, the β_1 and β_3 and explore their distribution.

```
gg1<- rbind(data.frame(beta1=beta.lasso.1),data.frame(beta1=beta.postlasso.1))
gg1$type=c(rep("lasso",100),rep("Post-Lasso OLS",100))
ggplot(gg1, aes(x=beta1, fill=type)) +
  geom_density(alpha=.5)+
  scale_fill_manual(values=c("steelblue","violet"))
```

```
gg3<- rbind(data.frame(beta3=beta.lasso.3),data.frame(beta3=beta.postlasso.3))
gg3$type=c(rep("lasso",100),rep("Post-Lasso OLS",100))
ggplot(gg3, aes(x=beta3, fill=type)) +
  geom_density(alpha=.5)+
  scale_fill_manual(values=c("steelblue","violet"))
```



The true values of β_1 and β_3 are 5, according to density lines above, we can find that Post-Lasso OLS also gets a more accurate estimation of β_1 and β_3 and estimates produced by Post-Lasso OLS have smaller bias. But Lasso tries to shrinkage the coefficients toward zero, so it always underestimates β's, and the estimated β's also are biased.

4 Example Three

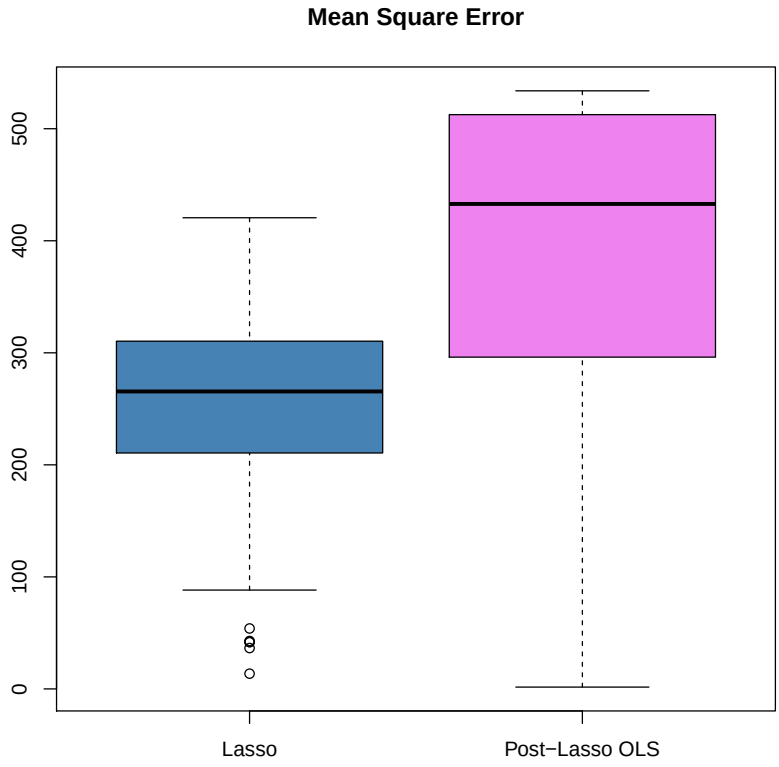
Here, I want to see the result if the model is not very sparse and explore whether Post-Lasso OLS can still be better than Lasso. Just change the true model and the other details are the same as Example Two. The number of variables with non-zero coefficients is 20 and the total number of variables is 100.

```
set.seed(777)
s=20 # number of variables with non-zero true coefficients
beta=c(rep(5,s),rep(0,p-s))
X_testing=matrix(rnorm(n_testing*p),ncol = p) #new X
Y_testing=X_testing%*%beta+rnorm(n_testing) # new Y
```

```
n_ite=100# Number of iteration
MSE.lasso=rep(0,n_ite)
MSE.Postlasso=rep(0,n_ite)
MSE.OLS=rep(0,n_ite)
for (i in 1:n_ite){
  X.train=matrix(rnorm(n_train*p),ncol = p)
  Y.train<-X.train%*%beta+rnorm(n_train,0,1)
  lasso.reg<-rlasso(Y.train~X.train,post = FALSE)
  yhat.lasso.testing=predict(rlasso(Y.train~X.train,post = FALSE),newdata = X_testing)
  post.lasso.reg=rlasso(Y.train~X.train,post = TRUE)
  yhat.postlasso.testing=predict(post.lasso.reg,newdata = X_testing)
```

```
  MSE.lasso[i]=mean((Y_testing-yhat.lasso.testing)^2)
  MSE.Postlasso[i]=mean((Y_testing-yhat.postlasso.testing)^2)
}
```

```
# Plot the distribution of the Mean Square Errors of these two regression methods
MSE=data.frame(cbind(MSE.lasso,MSE.Postlasso))
colnames(MSE)=c("Lasso","Post-Lasso OLS")
boxplot(MSE,col = c("steelblue","violet"),main="Mean Square Error")
```



Look at the result above, when the true model is not vary sparse, both of those two methods can not have a good prediction, and Post-Lasso OLS even worse than Lasso. This example also illustrates that Post-Lasso OLS can't overwhelm Lasso totally. We need to choose the most suitable model in specific problem.

5 Conclusion

OLS post-lasso estimator has the advantage of a smaller bias.That is the reason why it has a greater prediction power in certain situation. Regularization by the $l_1 - norm$ naturally helps the Lasso estimator to avoid overtting and get a sparse solution, but it also shrinks the fitted coefficients towards zero, causing a potentially signicant bias. Fortunately, Post-Lasso estimator can estimate the paramters with less bias. In the extreme case, when the model is sparse and lasso perfectly selects the true model, the estimation produced by post-lasso ols becomes perfect almostly.

In reality, the number of parameters in models to be estimated is always large relative to the sample size ($p \gg n$). The estimation and inference of *Post-Lasso OLS* should have a wide range of applications in econometrics and other disciplines.

References

[1] Belloni, A. and V. Chernozhukov. 2013. "Least squares after model selection in high dimensional sparse models," Bernoulli, 19(2).

[2] Robert Tibshirani. 1996. "Regression shrinkage and selection via the Lasso," Journal of the Royal Statistical Society. Series B, 58(1).

[3] Victor Chenozhukov, Christian Hansen, Martin Spindler. "HIGH-DIMENSIONAL METRICS IN R". Tutorial of **R** pachage *hdm*.