ML-Problem-Set-2

Gyongyver Kamenar (2103380)

3/7/2022

```
library(glmnet)
library(ggplot2)
library(purrr)
library(tidyverse)
library(kableExtra)
# My theme
devtools::source_url('https://raw.githubusercontent.com/gyongyver-droid/ceu-data-analysis/master/Assignm
theme_set(theme_gyongyver())
```

Problem 1

A)

Show that the solution to this problem is given by $\hat{\beta}_0^{ridge} = \sum_{i=1}^{n} Y_i/(n+\lambda)$. Compare this to the OLS estimator.

To minimize the expression we have to take the derivative and set it equal to 0.

$$\sum_{i=1}^{n} 2*(Y_i - b)*(-1) + 2\lambda b = 0$$

Transform to

$$-2\sum_{i=1}^{n}(Y_i - b) + 2\lambda b = 0$$

Divide by 2

$$-\sum_{i=1}^{n}(Y_i - b) + \lambda b = 0$$

Divide the summa into 2 parts. Only the Y part contains i and the b is taken n times.

$$-[\sum_{i=1}^{n}(Y_i) - nb] + \lambda b = 0$$

Reorganize the sides:

$$nb + \lambda b = \sum_{i=1}^{n}(Y_i)$$

$$(n+\lambda)b = \sum_{i=1}^{n}(Y_i)$$

Divide by $n + \lambda$

$$(n+\lambda)b = \sum_{i=1}^{n}(Y_i)$$

$$b = \sum_{i=1}^{n}(Y_i)/(n+\lambda)$$

Which is the solution of the problem:

$$\hat{\beta}_0^{ridge} = \sum_{i=1}^{n}(Y_i)/(n+\lambda)$$

Compating this to the OLS:

$$\hat{\beta}_0^{OLS} = \overline{Y} = \sum_{i=1}^{n}(Y_i)/n$$

So based on the two above formulas, we can see that $\hat{\beta}_0^{ridge}$ has $+\lambda$ in the denominator. We know that $\lambda = 0$ in the Ridge regression so the $\hat{\beta}_0^{ridge}$ coefficient will be smaller than the OLS coefficient. The higher the λ (penalty term) the higher the denominator so the ridge coefficient will be smaller. So we can see that λ is really a penalty / shrinkage parameter.

b)

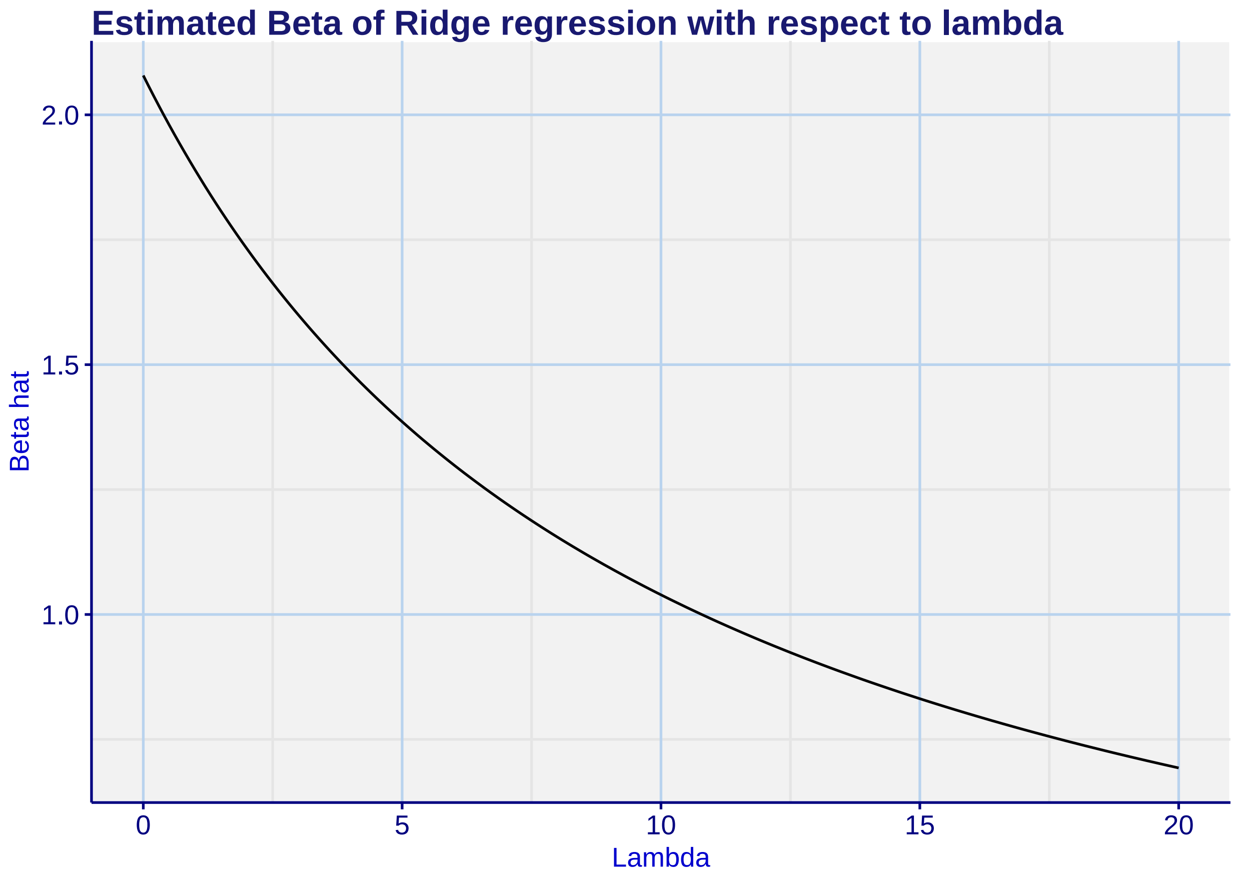
```
simulate_ridge<-function(n=10,sd=2){
  n=n
  e <- rnorm(n=n,mean=0,sd=sd)
  beta<-matrix(1,nrow = n,ncol = 1)
  y <- beta + e

  lambda <-seq(0,20,0.1)
  beta_hat <-sum(y)/(n+lambda)
  data.frame(lambda,beta_hat, beta=1,y_hat=beta_hat+e)
}

simulate_ridge() %>% kable()
```

lambda	beta_hat	beta	y_hat
0.0	0.5789277	1	2.4531597
0.1	0.5731958	1	-0.8239528
0.2	0.5675762	1	-0.0964724
0.3	0.5620658	1	0.5233247
0.4	0.5566613	1	1.3997671
0.5	0.5513598	1	-2.1410437
0.6	0.5461583	1	1.1819280
0.7	0.5410540	1	1.5001661
0.8	0.5360442	1	-3.2717269
0.9	0.5311264	1	0.6082971
1.0	0.5262980	1	2.4005299
1.1	0.5215565	1	-0.8755921
1.2	0.5168998	1	-0.1471488
1.3	0.5123254	1	0.4735844
1.4	0.5078314	1	1.3509372
1.5	0.5034154	1	-2.1889880
1.6	0.4990756	1	1.1348454
1.7	0.4948100	1	1.4539221
1.8	0.4906167	1	-3.3171543
1.9	0.4864939	1	0.5636646
2.0	0.4824398	1	2.3566717
2.1	0.4784527	1	-0.9186959
2.2	0.4745309	1	-0.1895176
2.3	0.4706730	1	0.4319319
2.4	0.4668772	1	1.3099831
2.5	0.4631422	1	-2.2292612
2.6	0.4594665	1	1.0952362
2.7	0.4558486	1	1.4149607
2.8	0.4522873	1	-3.3554838
2.9	0.4487812	1	0.5259519
3.0	0.4453290	1	2.3195610
3.1	0.4419296	1	-0.9552190
3.2	0.4385816	1	-0.2254670
3.3	0.4352840	1	0.3965430
3.4	0.4320356	1	1.2751415
3.5	0.4288354	1	-2.2635681
3.6	0.4256822	1	1.0614519
3.7	0.4225750	1	1.3816871
3.8	0.4195129	1	-3.3882582
3.9	0.4164948	1	0.4936655
4.0	0.4135198	1	2.2877517
4.1	0.4105871	1	-0.9865616
4.2	0.4076956	1	-0.2563530
4.3	0.4048446	1	0.3661035
4.4	0.4020332	1	1.2451390
4.5	0.3992605	1	-2.2931429
4.6	0.3965259	1	1.0322956
4.7	0.3938284	1	1.3529405
4.8	0.3911674	1	-3.4166037
4.9	0.3885421	1	0.4657128
5.0	0.3859518	1	2.2601837
5.1	0.3833959	1	-1.0137528
5.2	0.3808735	1	-0.2831751
5.3	0.3783842	1	0.3396431
5.4	0.3759271	1	1.2190330
5.5	0.3735018	1	-2.3189017
5.6	0.3711075	1	1.0068773
5.7	[illegible]	1	[illegible]

```
ggplot(simulate_ridge())+
  geom_line(aes(x=lambda,y=beta_hat))+
  labs(title = "Estimated Beta of Ridge regression with respect to lambda",y="Beta hat", x="Lambda")
```



C)

Repeat part b) 1000 times, for each value of lambda compute bias, variance and MSE of $\hat{\beta}_0^{ridge}$.

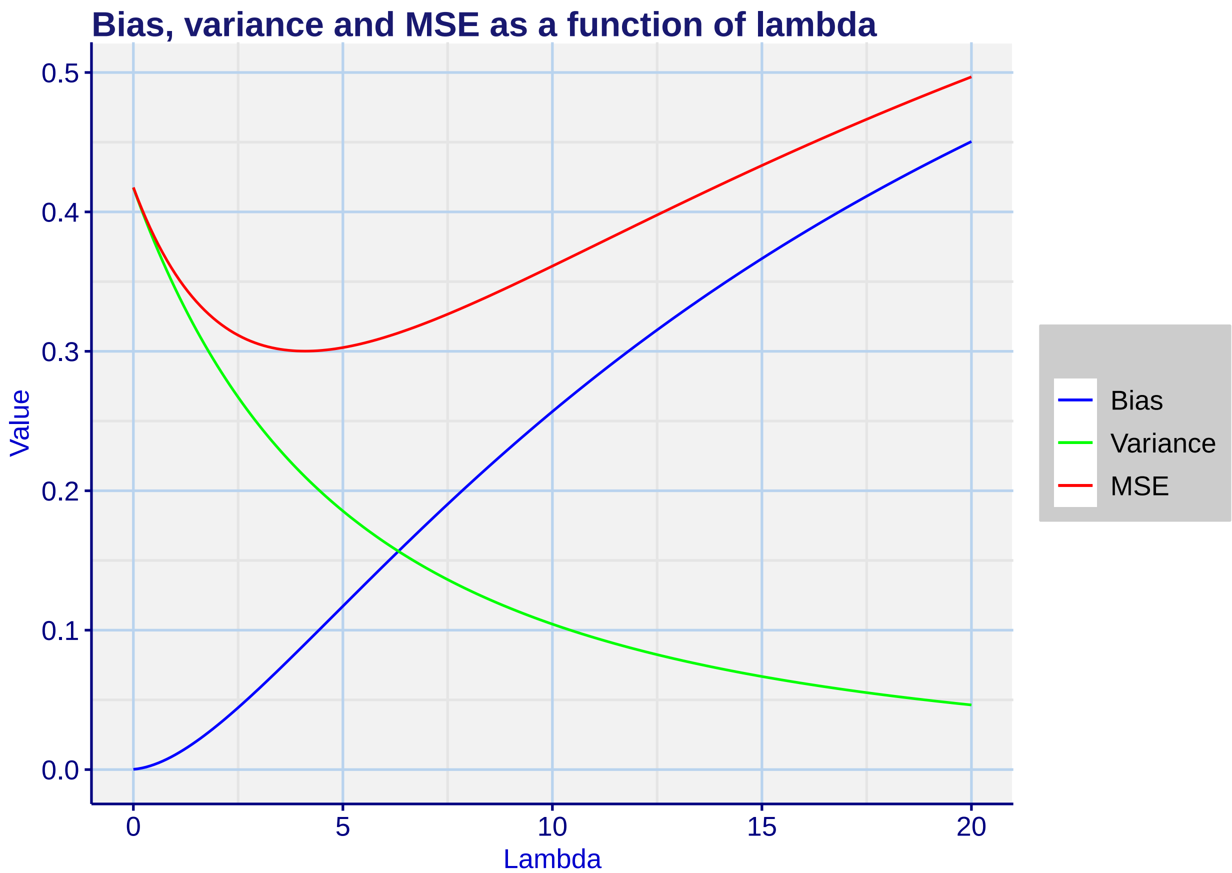
```
library(purrr)
df_1 <- map_df(seq(1,1000,1), ~{
  results = simulate_ridge(n=10)
  tibble(
    lambda = results$lambda,
    beta_hat = results$beta_hat,
    error = 1 - results$beta_hat
  )


}) %>% group_by(lambda) %>% summarise(bias=mean(error)^2, var=var(beta_hat), mse=bias+var)
```

D)

Plot bias, variance and MSE as a function of lambda and interpret the result.

```r
ggplot(df_1, aes(x=lambda))+
  geom_line(aes(y=bias,color="Bias"))+
  geom_line(aes(y=var, color="Variance"))+
  geom_line(aes(y=mse, color="MSE"))+
  scale_colour_manual("",
                      breaks = c("Bias", "Variance", "MSE"),
                      values = c("blue", "green", "red"))+
  labs(title = "Bias, variance and MSE as a function of lambda",y="Value",x="Lambda")
```



We can see, that as lambda is increasing, the bias is also increasing and the variance decreasing as we had expected based on on the theory of bias-variance tradeoff. The MSE takes U-shape as expected, so we can calculate that the lowest MSE is around lamdba = 5.

Problem 2

A)

$$max_{u_1,u_2}\ Var(u_1X + u_2Y) \quad s.t.\ \ u_1^2 + u_2^2 = 1$$

and suppose that

$$Var(X) > Var(Y) \quad and \quad Cov(X,Y) = E(XY) = 0$$

We can expand the variance formula:

$$Var(u_1X + u_2Y) = u_1^2Var(X) + u_2^2Var(Y) + 2u_1u_2Cov(X,Y)$$

and we know that the covariance is 0, so the problem is the following:

$$max_{u_1,u_2}\ (u_1^2Var(X)+u_2^2Var(Y)) \qquad s.t.\ \ u_1^2+u_2^2=1\ \ and\ \ Var(X)>Var(Y)\ \ \ and\ \ \ Cov(X,Y)=E(XY)=0$$

From this, it is trivial to see that $u_1^2Var(X)+u_2^2Var(Y)$ will be maximized if $u_1^2=1$ and $u_2^2=0$ because of the $Var(X)>Var(Y)$ condition. Therefore, there is no need to actually derive optimization problem.

The first principle component vector is $(u_1,u_2)=(1,0)$.

Problem 3