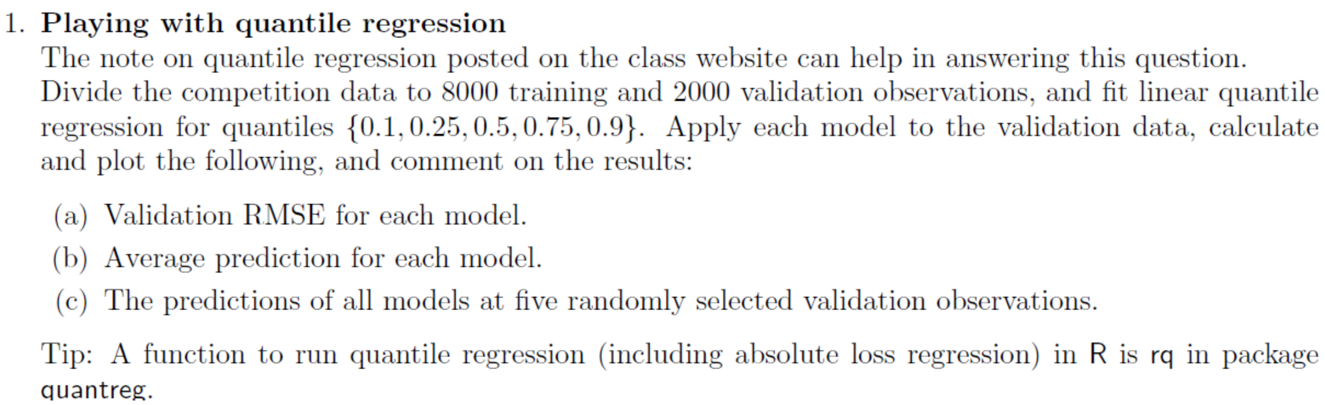
SL_EX2

roi hezkiyahu

15 11 2022

```
library(dplyr)
library(ggplot2)
library(purrr)
library(quantreg)
library(tidymodels)
```

Q1



1. **Playing with quantile regression**
The note on quantile regression posted on the class website can help in answering this question. Divide the competition data to 8000 training and 2000 validation observations, and fit linear quantile regression for quantiles $\{0.1, 0.25, 0.5, 0.75, 0.9\}$. Apply each model to the validation data, calculate and plot the following, and comment on the results:

   (a) Validation RMSE for each model.

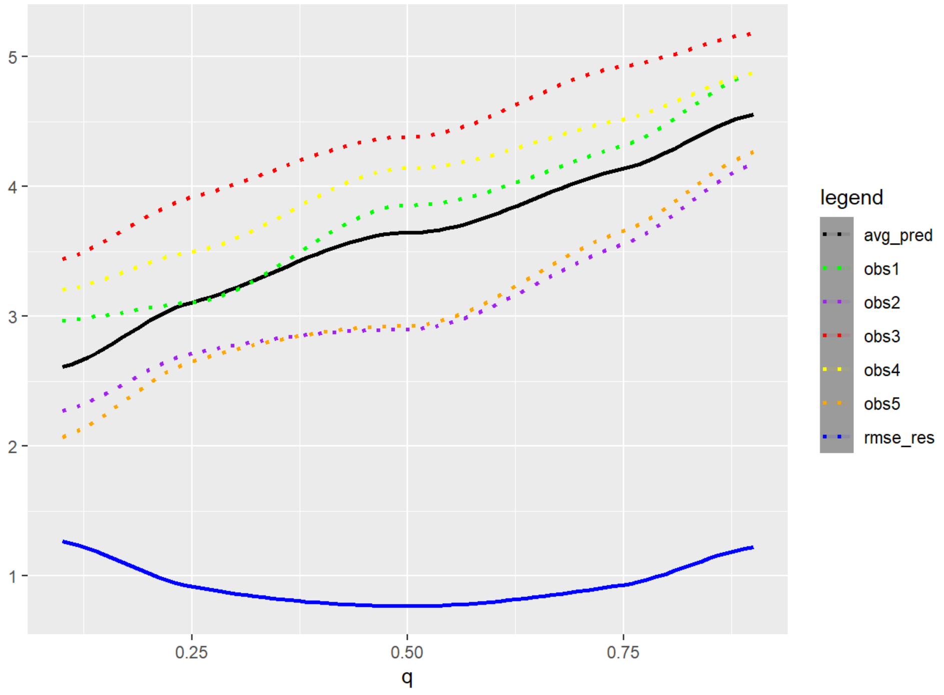
   (b) Average prediction for each model.

   (c) The predictions of all models at five randomly selected validation observations.

   Tip: A function to run quantile regression (including absolute loss regression) in R is rq in package quantreg.

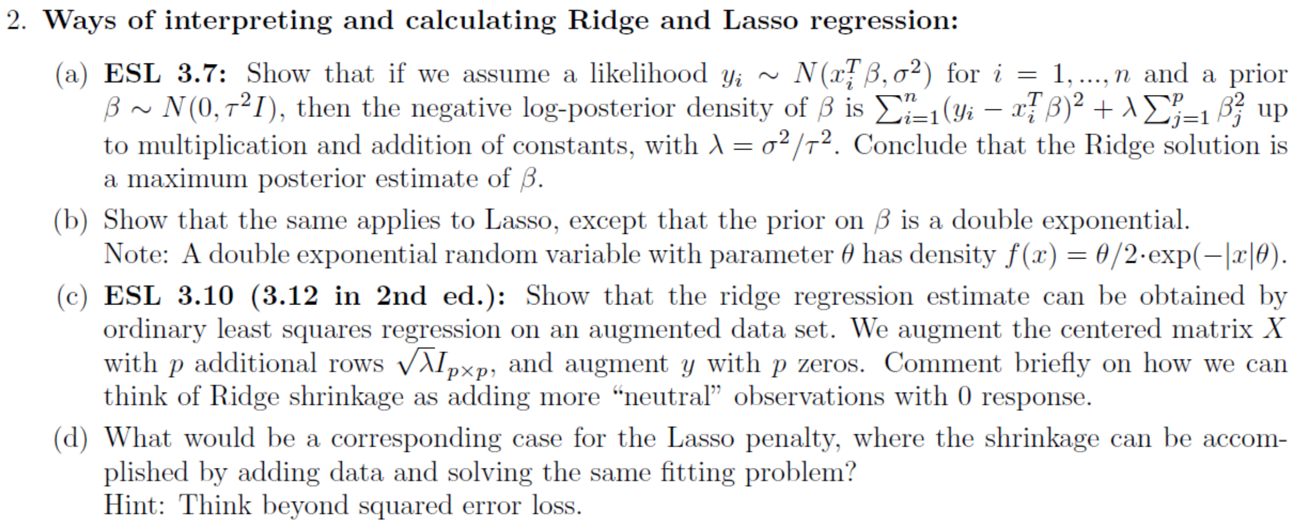
```
con <- url("http://www.tau.ac.il/~saharon/StatsLearn2022/train_ratings_all.dat")
X <- tibble(read.table(con))
con <- url("http://www.tau.ac.il/~saharon/StatsLearn2022/train_y_rating.dat")
y <- read.table(con)

X_full <- X %>% mutate(y = y$V1)
splits <- X_full %>%
  initial_split(prop = 0.8)

X_tr <- training(splits)
X_val <- testing(splits)
set.seed(123)
random_obs <- sample(1:2000,5)
fit_qr <- function(q,X_tr,X_val,random_obs){
  rq_lm <- rq(y~.,tau = q, data = X_tr)
  y_pred <- predict(rq_lm,X_val)
  y_true <- X_val %>% pull(y)
  rmse_res <- rmse_vec(y_pred,y_true)
  avg_pred <- mean(y_pred)
  random_point_pred <- y_pred[random_obs]
  return( list(rmse_res,avg_pred,random_point_pred))
}
qs <- c(0.1,0.25,0.5,0.75,0.9)
rmses <- c()
avg_preds <- c()
random_point_preds <- c()
for (i in 1:5){
  cur_res <- fit_qr(qs[i],X_tr,X_val,random_obs)
  rmses <- c(rmses,cur_res[[1]])
  avg_preds <- c(avg_preds,cur_res[[2]])
  random_point_preds <- c(random_point_preds,cur_res[[3]])
}

red_df <- tibble(q = qs,rmse_res = rmses, avg_pred = avg_preds,
       obs1 = random_point_preds[seq(1,25,5)],
       obs2 = random_point_preds[seq(2,25,5)],
       obs3 = random_point_preds[seq(3,25,5)],
       obs4 = random_point_preds[seq(4,25,5)],
       obs5 = random_point_preds[seq(5,25,5)])

colors = c("rmse_res"="blue", "avg_pred" ="black","obs1"="green","obs2"="purple","obs3"= "red","obs4"= "yellow","obs5"= "ora
nge")
ggplot(data = red_df) +
  geom_smooth(aes(x = q, y = rmse_res,color = "rmse_res")) +
  geom_smooth(aes(x = q, y = avg_pred,color = "avg_pred")) +
  geom_smooth(aes(x = q, y = obs1,color = "obs1"),linetype='dotted') +
  geom_smooth(aes(x = q, y = obs2,color = "obs2"),linetype='dotted') +
  geom_smooth(aes(x = q, y = obs3,color = "obs3"),linetype='dotted') +
  geom_smooth(aes(x = q, y = obs4,color = "obs4"),linetype='dotted') +
  geom_smooth(aes(x = q, y = obs5,color = "obs5"),linetype='dotted') +
  labs(x="q", y = "", color = "legend")+
  scale_color_manual(values = colors)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



we can clearly see that as q increases the observation prediction increases and the average prediction increases meaning our results lean towards higher values
we can also see that the farther away q is from 0.5 the rmse increases as well

Q2



2. **Ways of interpreting and calculating Ridge and Lasso regression:**

(a) **ESL 3.7:** Show that if we assume a likelihood $y_i \sim N(x_i^T\beta, \sigma^2)$ for $i = 1, ..., n$ and a prior $\beta \sim N(0, \tau^2 I)$, then the negative log-posterior density of β is $\sum_{i=1}^n (y_i - x_i^T\beta)^2 + \lambda \sum_{j=1}^p \beta_j^2$ up to multiplication and addition of constants, with $\lambda = \sigma^2/\tau^2$. Conclude that the Ridge solution is a maximum posterior estimate of β.

(b) Show that the same applies to Lasso, except that the prior on β is a double exponential.
Note: A double exponential random variable with parameter θ has density $f(x) = \theta/2 \cdot \exp(-|x|\theta)$.

(c) **ESL 3.10 (3.12 in 2nd ed.):** Show that the ridge regression estimate can be obtained by ordinary least squares regression on an augmented data set. We augment the centered matrix X with p additional rows $\sqrt{\lambda} I_{p\times p}$, and augment y with p zeros. Comment briefly on how we can think of Ridge shrinkage as adding more "neutral" observations with 0 response.

(d) What would be a corresponding case for the Lasso penalty, where the shrinkage can be accomplished by adding data and solving the same fitting problem?
Hint: Think beyond squared error loss.

a

we are lookin for$p(\beta|y)$
from Bayesien theory we know that: $p(\beta|y) = p(y|\beta)p(\beta)$

$$p(y|\beta) = C_\sigma e^{-\frac{\sum_{i=1}^n (y_i - x_i^t\beta)^2}{2\sigma^2}}$$

$$p(\beta) = C_\tau e^{-\frac{\sum_{j=1}^p \beta_j^2}{2\tau^2}}$$

$$p(y|\beta)p(\beta) \propto e^{-\frac{\sum_{i=1}^n (y_i - x_i^t\beta)^2}{2\sigma^2} - \frac{\sum_{j=1}^p \beta_j^2}{2\tau^2}} = e^{-\frac{\sum_{i=1}^n (y_i - x_i^t\beta)^2 + \frac{\sigma^2}{\tau^2}\sum_{j=1}^p \beta_j^2}{2\sigma^2}} = e^{-\frac{\sum_{i=1}^n (y_i - x_i^t\beta)^2 + \lambda\sum_{j=1}^p \beta_j^2}{2\sigma^2}}$$

$$-ln(p(y|\beta)p(\beta)) \propto \sum_{i=1}^n (y_i - x_i^t\beta)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

b

$$\text{we can define } \theta = \frac{1}{\tau^2} \text{ and write the probabbility as a funtion of } \tau:$$

$$p(\beta) = C_\tau e^{-\frac{\sum_{j=1}^{p} |\beta_j|}{\tau^2}}$$

$$p(y|\beta)p(\beta) \propto e^{-\frac{\sum_{i=1}^{n}(y_i - x_i^t\beta)^2}{2\sigma^2} - \frac{\sum_{j=1}^{p} |\beta_j|}{2\tau^2}} = e^{-\frac{\sum_{i=1}^{n}(y_i - x_i^t\beta)^2 + \frac{\sigma^2}{\tau^2}\sum_{j=1}^{p} |\beta_j|}{2\sigma^2}} = e^{-\frac{\sum_{i=1}^{n}(y_i - x_i^t\beta)^2 + \lambda\sum_{j=1}^{p} |\beta_j|}{2\sigma^2}}$$

$$-ln(p(y|\beta)p(\beta)) \propto \sum_{i=1}^{n}(y_i - x_i^t\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

c

$$\text{for writing convenience reasons define } \delta = \sqrt{\lambda}$$

$$X' := \begin{bmatrix} X \\ \delta I_p \end{bmatrix} \quad Y' := \begin{bmatrix} Y \\ 0_p \end{bmatrix}$$

$$\text{for out new set up the LSE is } \hat{\beta}' = (X'^t X')^{-1} X'^t Y' = (X^t X + \lambda I_p) \begin{bmatrix} X^t & \delta I_p \end{bmatrix} \begin{bmatrix} Y \\ 0_p \end{bmatrix} =$$

$$= (X^t X + \lambda I_p)(X^t Y + \delta I_p 0_p) = (X^t X + \lambda I_p) X^t Y = \hat{\beta}_{ridge}$$

we augmented the data set with p observation each with size δ in the direction of only one of the xes with y=0

thus for each x_i our RSS will be pentelized proportional to the size of β_i

due to the fact that as β_i increaces our new observation becomces farther from the target y=0

d

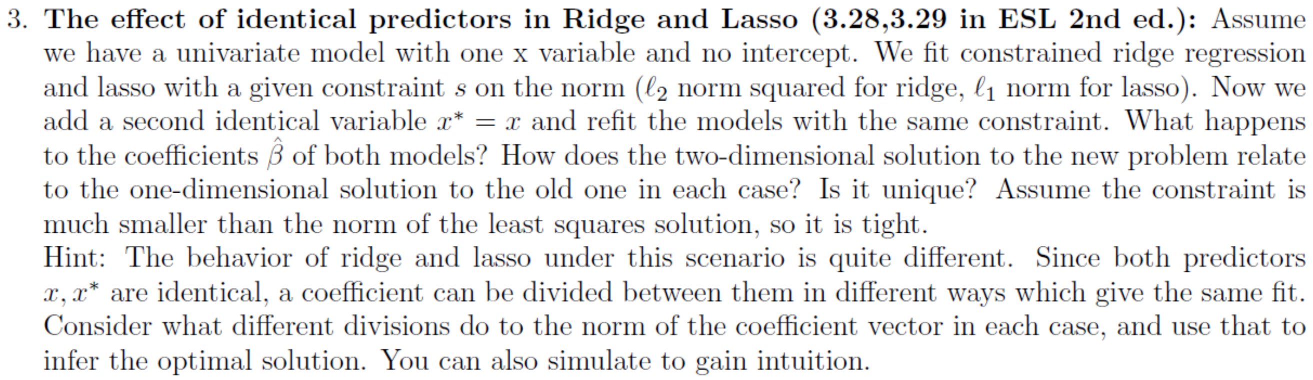
let our augmented dataset be:

$$X' := \begin{bmatrix} X \\ \lambda I_p \end{bmatrix} \quad Y' := \begin{bmatrix} Y \\ 0_p \end{bmatrix}$$

and lets take a look at the absolute loss:

$$L(Y', \beta) = \sum_{i=1}^{n+p} |y_i' - x_i'^t\beta| = \sum_{i=1}^{n} |y_i - x_i^t\beta| + \sum_{i=n}^{n+p} |y_i' - x_i'^t\beta| = L(Y, \beta) + \lambda\sum_{i=n}^{n+p} |\beta_i|$$

Q3



3. **The effect of identical predictors in Ridge and Lasso (3.28,3.29 in ESL 2nd ed.):** Assume we have a univariate model with one x variable and no intercept. We fit constrained ridge regression and lasso with a given constraint s on the norm (ℓ_2 norm squared for ridge, ℓ_1 norm for lasso). Now we add a second identical variable $x^* = x$ and refit the models with the same constraint. What happens to the coefficients $\hat{\beta}$ of both models? How does the two-dimensional solution to the new problem relate to the one-dimensional solution to the old one in each case? Is it unique? Assume the constraint is much smaller than the norm of the least squares solution, so it is tight.
Hint: The behavior of ridge and lasso under this scenario is quite different. Since both predictors x, x^* are identical, a coefficient can be divided between them in different ways which give the same fit. Consider what different divisions do to the norm of the coefficient vector in each case, and use that to infer the optimal solution. You can also simulate to gain intuition.

Ridge

our new optimization problem is:

$$minimize||y - x\beta_1 - x^*\beta_2|| + \lambda\beta_1^2 + \lambda\beta_2^2 := minimizeL^*$$

$$\frac{\partial L^*}{\partial\beta_1} = 2\sum_{i=1}^{n}(y_i - x_i(\beta_1 + \beta_2))x_i + 2\lambda\beta_1$$

$$\frac{\partial L^*}{\partial\beta_2} = 2\sum_{i=1}^{n}(y_i - x_i(\beta_1 + \beta_2))x_i + 2\lambda\beta_2$$

setting both to zero and subtracting between the two equations gets us:

$2\lambda\beta_1 = 2\lambda\beta_2 \Rightarrow$ in the optimal solution: $\beta_1 = \beta_2$

(in the case where $\lambda = 0$ any combination that satisfiys: $\beta_1 + \beta_2 = \beta$ is an optimal solution)

now plugging the constraint back to L^* yields the following problem:

$$minimize||y - 2x\beta_1|| + 2\lambda\beta_1^2$$

using our knowlage for its minimizer we can conclude: $\hat{\beta}_1 = (4x^tx + 2\lambda)^{-1}2x^ty = (2x^tx + \lambda)^{-1}x^ty = \hat{\beta}_2$

we get a smaller coefficient then $\hat{\beta}$ and the solution is unique

Lasso

our new optimization problem is:

$$minimize||y - x\beta_1 - x^*\beta_2|| + \lambda|\beta_1| + \lambda|\beta_2| := minimizeL^*$$

$$\frac{\partial L^*}{\partial \beta_1} = 2\sum_{i=1}^{n}(y_i - x_i(\beta_1 + \beta_2))x_i + sign(\beta_1)\lambda$$

$$\frac{\partial L^*}{\partial \beta_2} = 2\sum_{i=1}^{n}(y_i - x_i(\beta_1 + \beta_2))x_i + sign(\beta_2)\lambda$$

setting both to zero and subtracting between the two equations gets us:

$sign(\beta_1)\lambda = sign(\beta_2)\lambda \Rightarrow$ in the optimal solution: $sign(\beta_1) = sign(\beta_2)$

(in the case where $\lambda = 0$ any combination that satisfiys: $\beta_1 + \beta_2 = \beta$ is an optimal solution)

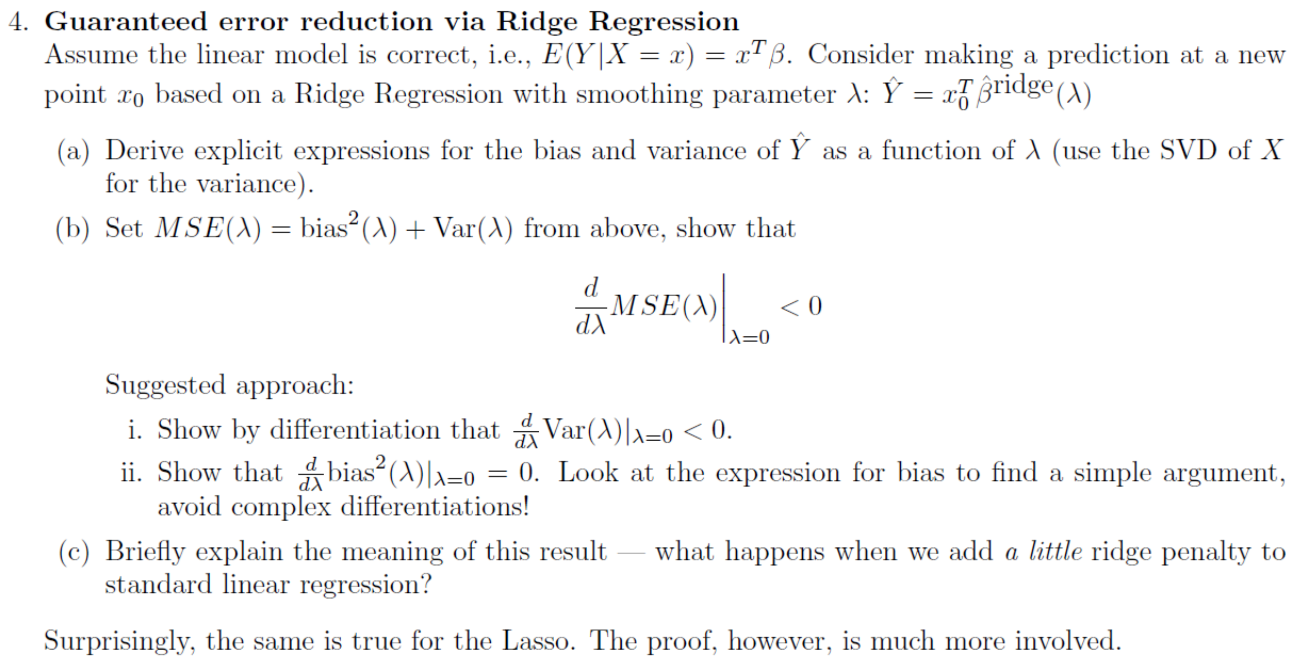
now plugging the constraint back to L^* yields the following problem:

$$minimize||y - x(\beta_1 + \beta_2)|| + sign(\beta_1)\lambda(\beta_1 + \beta_2) = ||y - x(\beta_1 + \beta_2)|| + sign(\beta_1 + \beta_2)\lambda(\beta_1 + \beta_2) = ||y - x(\beta')|| + sign(\beta')\lambda\beta'$$

we know that the minimezer for this function is: $\hat{\beta} \Rightarrow$

$\Rightarrow \{(\hat{\beta}_1, \hat{\beta}_2);\quad \hat{\beta}_1 + \hat{\beta}_2 = \hat{\beta} \wedge sign(\beta_1) = sign(\beta_2)\}$ is the set of optimal solutions

Q4



4. **Guaranteed error reduction via Ridge Regression**

Assume the linear model is correct, i.e., $E(Y|X = x) = x^T\beta$. Consider making a prediction at a new point x_0 based on a Ridge Regression with smoothing parameter λ: $\hat{Y} = x_0^T\hat{\beta}^{\text{ridge}}(\lambda)$

(a) Derive explicit expressions for the bias and variance of $\hat{Y}$ as a function of λ (use the SVD of X for the variance).

(b) Set $MSE(\lambda) = \text{bias}^2(\lambda) + \text{Var}(\lambda)$ from above, show that

$$\frac{d}{d\lambda}MSE(\lambda)\Big|_{\lambda=0} < 0$$

Suggested approach:

i. Show by differentiation that $\frac{d}{d\lambda}\text{Var}(\lambda)|_{\lambda=0} < 0$.

ii. Show that $\frac{d}{d\lambda}\text{bias}^2(\lambda)|_{\lambda=0} = 0$. Look at the expression for bias to find a simple argument, avoid complex differentiations!

(c) Briefly explain the meaning of this result — what happens when we add *a little* ridge penalty to standard linear regression?

Surprisingly, the same is true for the Lasso. The proof, however, is much more involved.

a

recall:

$$X = UDV^t$$

$$X^tX = VD^2V^t$$

$$(X^tX)^{-1} = VD^{-2}V^t$$

$$(X^tX + \lambda I_p)^{-1} = V(D^2 + \lambda I_p)^{-1}V^t$$

$$X(X^tX + \lambda I_p)^{-1}X^t = UD(D^2 + \lambda I_p)^{-1}DU^t = D^2(D^2 + \lambda I_p)^{-1}$$

$$\hat{Y} = X\hat{\beta} = X(X^tX + \lambda I_p)^{-1}X^tY$$

express the variance

$$V(\hat{Y}) = V(X\hat{\beta}) = V(X(X^tX + \lambda I_p)^{-1}X^tY) = UD(D^2 + \lambda I_p)^{-1}DU^t\sigma^2IUD(D^2 + \lambda I_p)^{-1}DU^t =$$

$$= \sigma^2UD(D^2 + \lambda I_p)^{-1}D^2(D^2 + \lambda I_p)^{-1}DU^t = \sigma^2UD^4(D^2 + \lambda I_p)^{-2}U^t = \sigma^2D^4(D^2 + \lambda I_p)^{-2}$$

express the $bias^2$

$$E(\hat{Y}) = E(X\hat{\beta}) = E(X(X^tX + \lambda I_p)^{-1}X^tY) = X(X^tX + \lambda I_p)^{-1}X^tE(Y)$$

$$bias^2 = (E(Y) - E(\hat{Y}))^T(E(Y) - E(\hat{Y})) = E(Y)^t(I - X(X^tX + \lambda I_p)^{-1}X^t)^t(I - X(X^tX + \lambda I_p)^{-1}X^t)E(Y) =$$

$$= E(Y)^t(I - D^2(D^2 + \lambda I_p)^{-1})^t(I - D^2(D^2 + \lambda I_p)^{-1})E(Y) = E(Y)^t(I - 2D^2(D^2 + \lambda I_p)^{-1} + D^4(D^2 + \lambda I_p)^{-2})E(Y)$$

b

differentiate

$$\frac{\partial 2D^2(D^2+\lambda I_p)^{-1}}{\partial\lambda} = -2D^2(D^2+\lambda I_p)^{-2}$$

$$\frac{\partial D^4(D^2+\lambda I_p)^{-2}}{\partial\lambda} = -2D^4(D^2+\lambda I_p)^{-3}$$

$$\frac{\partial bias^2}{\lambda} = E(Y)^t(2D^2(D^2+\lambda I_p)^{-2} - 2D^4(D^2+\lambda I_p)^{-3})E(Y) = E(Y)^t((2(D^2+\lambda I_p)^{-1})(D^2(D^2+\lambda I_p)^{-1} - (D^2(D^2+\lambda I_p)^{-1})^2)E($$

plug in $\lambda = 0 \Rightarrow E(Y)^t((2(D^2)^{-1})(D^2(D^2)^{-1} - (D^2(D^2)^{-1})^2)E(Y) = 0$

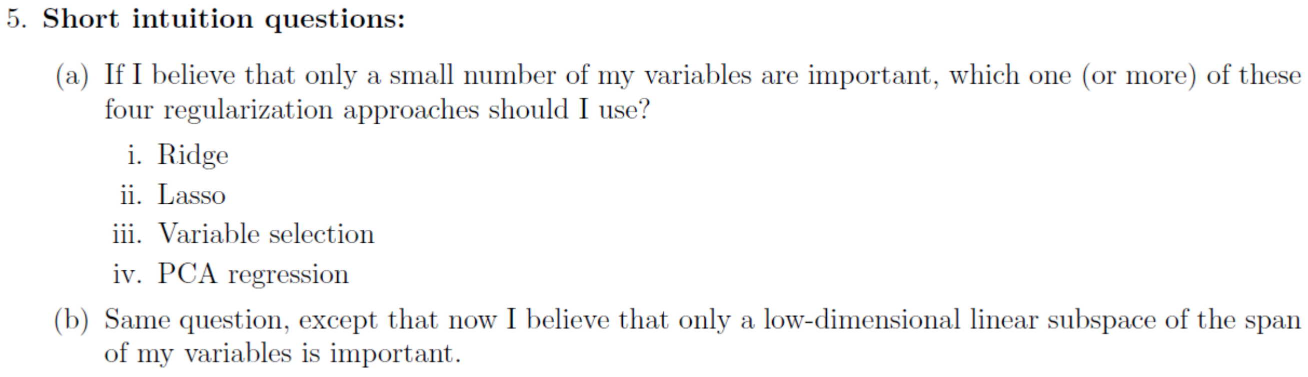
$$\frac{\partial\sigma^2 D^4(D^2+\lambda I_p)^{-2}}{\partial\lambda} = -\sigma^2 D^4(D^2+\lambda I_p)^{-3}$$

notice that all the values for D are positive, λ, σ^2 are also positive thus: $\frac{\partial V(\hat{Y})}{\partial\lambda} < 0$

c

adding a little ridge penalty will result in lowering the variance
as we proved around $\lambda = 0$ the variance of our predictions will decreace if we increace λ
also our bias should stay close to zero thus our general MSE will decrease

Q5



5. **Short intuition questions:**

(a) If I believe that only a small number of my variables are important, which one (or more) of these four regularization approaches should I use?

i. Ridge
ii. Lasso
iii. Variable selection
iv. PCA regression

(b) Same question, except that now I believe that only a low-dimensional linear subspace of the span of my variables is important.

a

ii) lasso tends to shrink some fo the betas to zero thus preforming variable selection, also if p (and the training time of the model) is not very large we can preform Variable selection

b

iv) PCA projects our X to a new low dimensional linear space with a recombination of our original predictors