Homework 1

Ngoc Duong

9/30/2020

Question 1

Let X denote an $n \times p$ matrix with each row an input vector and y denote an n-dimensional vector of the output in the training set. For fixed q > 1, define

a) Show that $Bridge_\lambda(\beta)$ for $\lambda > 0$ is a convex function in β, which is strictly convex for $q > 1$

We have the Bridge regression function with penalty term as:

$$Bridge_\lambda(\beta) = (y - X\beta)^T(y - X\beta) + \lambda\Sigma_{j=1}^{p}|\beta_j|^q$$

We can look at the two terms separately:

The first term is the sum of square residuals, or the cost function for the OLS linear regression optimization problem. In the case of least squares, the cost function is always convex in β regardless of the data (X, y), although it may not be strictly convex unless X is full-rank.

For $q > 1$, we want to show the second term $\lambda\Sigma_{j=1}^{p}|\beta_j|^q$ is strictly convex by showing its individual component $|\beta_j|^q$ is strictly convex.

By the definition of convexity, we want to show

$$|t\beta_1 + (1-t)\beta_2|^q < t|\beta_1|^q + (1-t)|\beta_2|^q$$

where $t \in (0, 1)$

This is equivalent to $|t\beta_1 + (1-t)\beta_2| < (t|\beta_1|^q + (1-t)|\beta_2|^q)^{1/q}$

Define p st $\frac{1}{q} + \frac{1}{p} = 1$. Then by Holder's Inequality and Triangle Inequality, we have:

$$|t\beta_1 + (1-t)\beta_2| \leq t|\beta_1| + (1-t)|\beta_2| = t^{1/p}t^{1/q}|\beta_1| + (1-t)^{1/p}(1-t)^{1/q}|\beta_2|$$

$$< (t|\beta_1|^q + (1-t)|\beta_2|^q)^{1/q}(t + 1 - t)^{1/p} = (t|\beta_1|^q + (1-t)|\beta_2|^q)^{1/q}$$

which holds when $q > 1$

Therefore, $Bridge_\lambda(\beta)$ when $\lambda > 0$ is convex in β and strictly convex for $q > 1$.

b) We want to show that for $q > 1$ there is a unique minimizer, $\hat{\beta}(\lambda)$, with $\Sigma_{j=1}^{p}|\hat{\beta}_j(\lambda)|^q \leq t_0$

We already showed that $Bridge_\lambda(\beta)$ is a (strictly) convex function in part a, then suppose there exists a unique minimizer, we will have

$$\frac{\partial Bridge_\lambda(\beta)}{\partial\beta} = \frac{\partial(y - X\beta)^T(y - X\beta)}{\partial\beta} + \frac{\lambda\partial\Sigma_{j=1}^{p}|\beta_j|^q}{\partial\beta} = 0$$

$$\Leftrightarrow \frac{\partial (y - X\beta)^T (y - X\beta)}{\partial \beta_j} = -\frac{\lambda \partial \Sigma_{j=1}^{p} |\beta_j|^q}{\partial \beta_j}$$

Denote the LHS as $S_j(\beta, X, y)$ and RHS as $-d(\beta_j, \lambda, q)$. And rewrite β as $(\beta_j, \beta_{(-j)})$ where, $\beta_{(-j)}$ is a (p-1) vector whose elements are $\beta_{i \neq j}$. We want to show there exists only one point β_j that satisfies the above equation, therefore is the unique minimizer.

We have, $S_j(\beta, X, y) = 2x_j^T x_j \beta_j + 2\Sigma_{i \neq j} x_j^T x_i \beta_j - 2x_j^T \beta_j$ is linear in β_j and has positive slope $2x_j^T x_j$, given fixed $\beta_{(-j)}$. Meanwhile, the function $-d(\beta_j, \lambda, q) = -\lambda q |\beta_j|^{q-1} sign(\beta_j)$ is non-linear in β_j and monotone decreasing when $q > 1$. This function is also continuous and differentiable when $q > 1$, except when $\beta_j = 0$ for $q \in (1, 2)$. We can then see that the two functions will meet at one point/has a unique solution when $q > 1$.

c) Then show that for $q = 1$, there exists a minimizer and for all minimizers, $\hat{\beta}(\lambda)$, the penalty function takes the same value $s(\lambda) = \Sigma_{i=1}^{p} |\hat{\beta}(\lambda)|^q \leq t$

When $q = 1$, the problem becomes LASSO, which can have either unique or infinitely many solutions. We can prove the latter by contradiction.

Consider an example with two solutions $\hat{\beta}_1 \neq \hat{\beta}_2$ where $\mathbf{X}\hat{\beta}_1 \neq \mathbf{X}\hat{\beta}_2$, and let m be some metric of the Lasso fit. Since we assumed these are two different fits, they give two different values of m_1 and m_2. Let $m^* = min(m_1, m_2)$. For any $0 < t < 1$, we have

$$(y - X(t\hat{\beta}_1 + (1-t)\hat{\beta}_2))^2 + \lambda |t\hat{\beta}_1 + (1-t)\hat{\beta}_2| < tm^* + (1-t)m^* = m^*,$$

by the definition of convexity, and because both terms on the LHS are either strictly convex or convex.

Since the fit of $t\hat{\beta}_1 + (1-t)\hat{\beta}_2$ achieves a lower metric value than m^*, there is a contradiction. Therefore, $\mathbf{X}\hat{\beta}_1 = \mathbf{X}\hat{\beta}_2$ where $\hat{\beta}_1 \neq \hat{\beta}_2$.

We can see that the LASSO estimate may not be unique, in which case all minimizers can satisfy the same constraint or lead to the same penalty value. Therefore, since the penalty function has a unique value, $s(\lambda)$ well-defined for both $q = 1$ and $q > 1$ when $\lambda > 0$.

d) We want to show both problems are equivalent

$min_\beta[(y - X\beta)^2 + \lambda \Sigma^p |\beta|^q](1)$

We start from $min(y - X\beta)^2$ s.t. $\Sigma^p |\beta_j|^q \leq s(\lambda)$. By KKT conditions, this can be rewritten as:

$$min(y - X\beta)^2 + \mu(\Sigma^p |\beta_j|^q - s(\lambda)) = g(\mu, \beta)(2)$$

where $\frac{\partial g(\mu, \beta)}{\partial \beta} = 0$, subject to $\mu \geq 0$, $\Sigma^p |\beta_j|^q \leq s(\lambda)$, and $\mu(\Sigma^p |\beta_j|^q - s(\lambda)) = 0$

Let $\hat{\beta}_\lambda = argmin_\beta Bridge_\lambda(\beta), \lambda > 0$.

First-order conditions of (1) and (2) being equal means that $\mu = \lambda$, but $\lambda > 0$ by assumption, so μ must be positive. This along with the last KKT condition $\mu(\Sigma^p |\beta_j|^q - s(\lambda)) = 0$ implies that $\Sigma^p |\beta_j|^q - s(\lambda) = 0$. Since $\hat{\beta}_\lambda$ satisfies this condition, we can say there exists a solution that meets non-affine inequality constraints in this convex problem, which implies strong duality. Since KKT conditions are always sufficient and necessary under strong duality, the two forms are equivalent.

Question 2

We perform best subset, forward stepwise, and backward stepwise selection on a single dataset. For each approach, we obtain $p + 1$ models, containing 0,1,2,…,p predictors.

a) The model produced by best subset selection with k predictors has the smallest training RSS, since best-subset selection will cover every possible model, but forward and backward selection will only go down some particular path so they might miss the optimal choice.

b) The best test RSS model could be any of the three. However, depending on the data, the models that overfit or underfit will have lower test RSS.

c) True or False

(i) True, because once forward selection has selected some variables in the model, it will keep the variables, regardless of what variables it adds next.

(ii) True. Backward selection starts off with full model and removes the worst predictor based on some selection criteria (AIC, BIC, p-value, etc.) at each step.

(iii) False. Backward and forward stepwise regressions might go down different paths when including/eliminating variables so even at the step k and (k+1) of these two methods, it's possible that one model is not nested in the other.

(iv) False, for the same reason as in (iii).

(v) False. Best subset might replace a predictor going from step k to (k+1) because it looks at different combinations that have (k+1) predictors and the chosen combination at step (k+1) may not include all the predictors chosen at step k.

Question 3

a) Best-subset selection

Under an orthogonal design matrix, as a property of this type of matrix, we have $\mathbf{X}^T\mathbf{X} = \mathbf{I}$. In addition, we also have $y = X\hat{\beta}_{OLS}$, so $\hat{\beta}_{OLS} = X^T y$. Then we have:

$$(y - X\beta)^T(y - X\beta) = y^T y - \beta^T\beta - 2y^T X\beta = (\beta - \hat{\beta}^{OLS})^T(\beta - \hat{\beta}^{OLS}) + y^T(I - XX^T)y(1)$$

, but the second term does not contain β so in the minimization problem we don't need to consider it.

The best subset solution can be rewritten as the following optimization problem with constraint:

$$argmin_\beta (y - X\beta)^T(y - X\beta), \text{ s.t. } \Sigma_{j=1}^p \ I_{\beta_j \neq 0} \leq k$$

which, from the derivation (1) above, is equivalent to

$$argmin_\beta (\beta - \hat{\beta}^{OLS})^T(\beta - \hat{\beta}^{OLS}) = argmin_\beta \Sigma_{i=1}(\beta_i - \hat{\beta}_i^{OLS})^2, \text{ s.t. } \Sigma_{j=1}^p \ I_{\beta_j \neq 0} \leq k$$

where $\hat{\beta}^{OLS} \in \mathbf{R}^p$ and the elements of $\hat{\beta}^{OLS}$ are ordered from largest to smallest in abolute value, i.e., $|\hat{\beta}_{(1)}^{OLS}| \geq |\hat{\beta}_{(2)}^{OLS}| \geq ... \geq |\hat{\beta}_{(p)}^{OLS}|$

Let $S = \{i | \beta_i \neq 0\}$, then we can rewrite the objective function as:

$$\Sigma_{i \in S}(\beta_i - \hat{\beta}_i^{OLS})^2 + \Sigma_{i \notin S}(\beta_i - \hat{\beta}_i^{OLS})^2 = \Sigma_{i \in S}(\beta_i - \hat{\beta}_i^{OLS})^2 + \Sigma_{i \notin S}(\hat{\beta}_i^{OLS})^2$$

(because when $i \notin S$, $\beta_i = 0$)

If we set $\beta_i = \hat{\beta}_i^{OLS}$ for $i \in S$, i.e., when $\beta_i \neq 0$, the objective function becomes $\Sigma_{i \notin S}(\hat{\beta}_i^{OLS})^2$, and we can see that this function is minimized with the (p-k) smallest OLS coefficient estimates, which also means the set S should contain the indices for the largest k values of $\hat{\beta}_i^{OLS}$, and $\beta_i = \hat{\beta}_i^{OLS}$

Therefore, we have the solution $\hat{\beta}_j^{OLS}.I(|\hat{\beta}_j^{OLS}| \geq |\hat{\beta}_{(M)}^{OLS}|)$.

b) Ridge regression

We have the solution for ridge as:

$$\hat{\beta}^{ridge} = argmin_{\beta}\{(y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta\}$$

Again, in the above function, we assume X the design matrix has dimension $n \times p$, and y the outcome vector has dimension $n \times 1$. Then, we can define the new

$$X_{\lambda} = \begin{bmatrix} X \\ \sqrt{\lambda}I_p \end{bmatrix}$$

, where I_p is a $p \times p$ identity matrix, and if we define

$$y_{\lambda} = \begin{bmatrix} y \\ O_p \end{bmatrix}$$

where O_p is a vector of zeroes with dimension $p \times 1$.

This can be treated as an OLS problem, so we have:

$$\hat{\beta}^{ridge} = argmin_{\beta}(X_{\lambda}\beta - y_{\lambda})^2$$

The closed-form solution is then:

$$\hat{\beta}^{ridge} = (X_{\lambda}^T X_{\lambda})^{-1} X_{\lambda}^T y_{\lambda} = (X^T X + \lambda I_p)^{-1} X^T y = (I_p + \lambda I_p)^{-1} X^T y$$

, since $X^T X = I_p$ (with $X_{n \times p}$ that is orthogonal)

Then we have:

$$\hat{\beta}^{ridge} = \frac{X^T y}{1 + \lambda} = \frac{\hat{\beta}^{OLS}}{1 + \lambda}$$

c) LASSO

We have the objective function:

$$min_{\beta}\{\frac{1}{2}(y - X\hat{\beta})^T(y - X\hat{\beta}) + \lambda\Sigma_i|\hat{\beta}_i|\}$$

Assume $\hat{\beta} \neq 0$, then take the first derivative of the function with penalty term w.r.t $\hat{\beta}$ and solve, we obtain:

$$-X^T(y - X\hat{\beta}) + sign(\hat{\beta})\lambda = 0$$

$$\Leftrightarrow -X^T y + X^T X\hat{\beta} + sign(\hat{\beta})\lambda = 0$$

$$\Leftrightarrow -X^T y + \hat{\beta} + sign(\hat{\beta})\lambda = 0$$

($X^T X = I$ since X is orthogonal)

$$\Leftrightarrow \hat{\beta}^{LASSO} = X^T y - sign(\hat{\beta}^{OLS})\lambda = \hat{\beta}^{OLS} - sign(\hat{\beta}^{OLS})\lambda$$

- Case 1: For $\lambda > 0$, if $\hat{\beta}^{LASSO} > 0$, then $RHS = \hat{\beta}^{OLS} - \lambda > 0$, which also implies $\hat{\beta}^{OLS} > \lambda > 0$. Then given this positive $\hat{\beta}^{OLS}$, the solution can be expressed as $\hat{\beta}^{LASSO} = \hat{\beta}^{OLS} - \lambda = sign(\hat{\beta}^{OLS})(|\hat{\beta}^{OLS}| - \lambda)$

- Case 2: For $\lambda > 0$, if $\hat{\beta}^{LASSO} < 0$, then $RHS = \hat{\beta}^{OLS} - (-\lambda) < 0 \Leftrightarrow \hat{\beta}^{OLS} < -\lambda < 0$. Then given this negative $\hat{\beta}^{OLS}$, the solution can be expressed as $\hat{\beta}^{LASSO} = \hat{\beta}^{OLS} + \lambda = -|\hat{\beta}^{OLS}| + \lambda = sign(\hat{\beta}^{OLS})(|\hat{\beta}^{OLS}| - \lambda)$

The remaining case is $\hat{\beta}^{LASSO} = 0$, which happens when $|\hat{\beta}^{OLS}| < \lambda$, because the assumption of OLS estimate being different from 0 is not met.

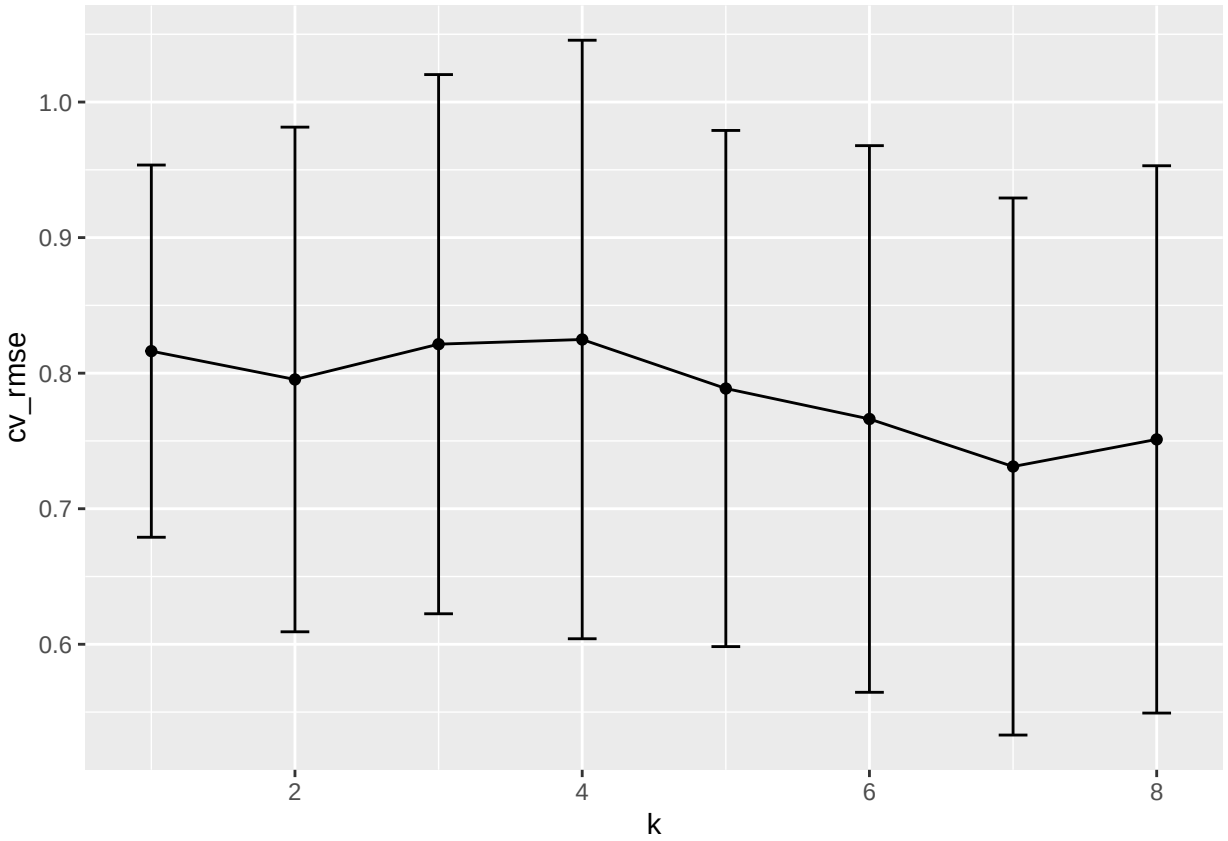
To summarize all the cases, we take the "positive part" of the solution in each case, and obtain the general formula for the LASSO solution:

$$\hat{\beta}_j^{LASSO} = sign(\hat{\beta}_j^{OLS})(|\hat{\beta}_j^{OLS}| - \lambda, 0)_+$$

. If $|\hat{\beta}_j^{OLS}|$ in some scenario is under the threshold λ, the Lasso will shrink the coefficients all the way to 0.

Question 4

a) Best-subset linear regression with k chosen by 5-fold cross-validation



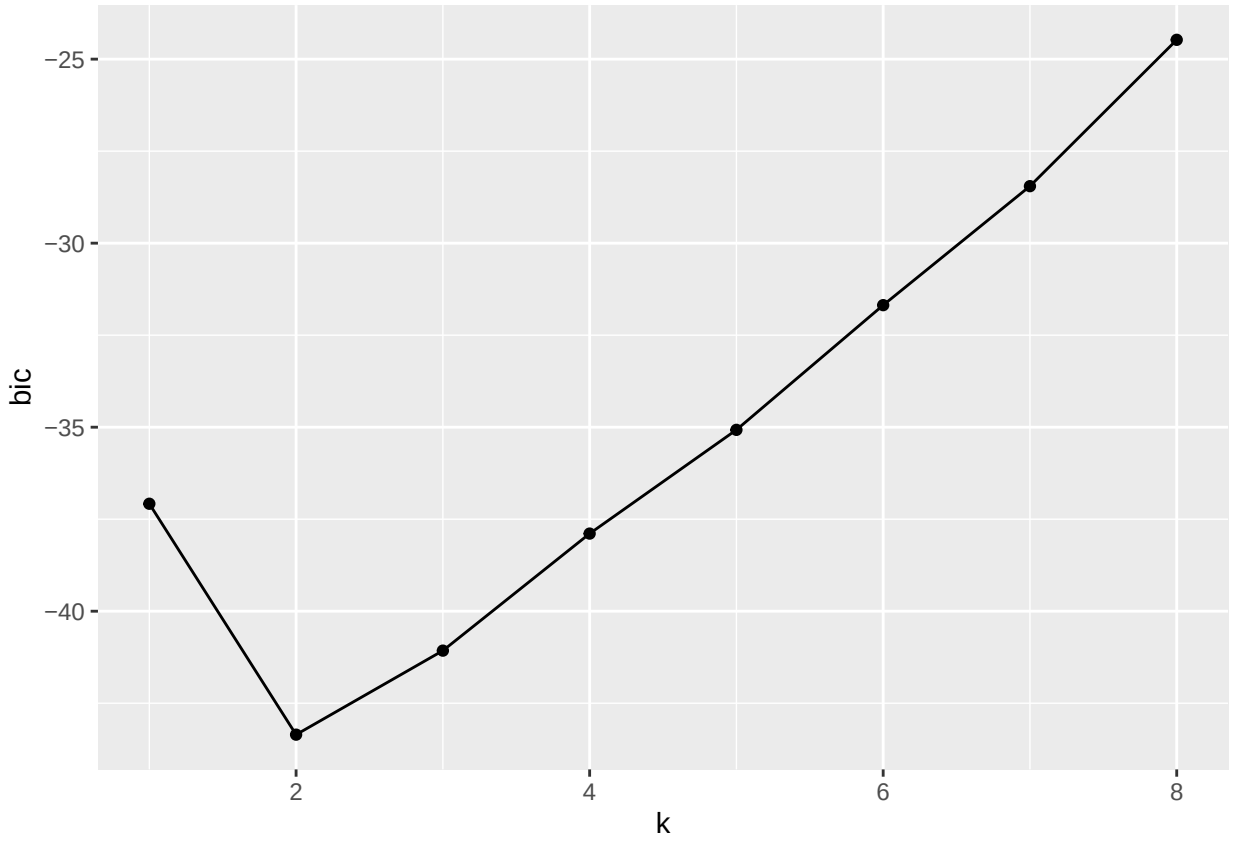

```
## [1] 0.1750045
```

The subset size k that gives the lowest CV RMSE is 7. The associated mean CV MSE is 0.73. On test data, MSE is 0.516, and the standard error for this MSE on the test set is 0.175.

The model's variable names and coefficients can be found below:

```
##  (Intercept)       lcavol      lweight          age         lbph         svi1
##  0.259061747  0.573930391  0.619208833 -0.019479879  0.144426474  0.741781258
##          lcp        pgg45
## -0.205416986  0.008944996
```

b) Best-subset linear regression with k chosen by BIC




```
## [1] 0.4924823
```

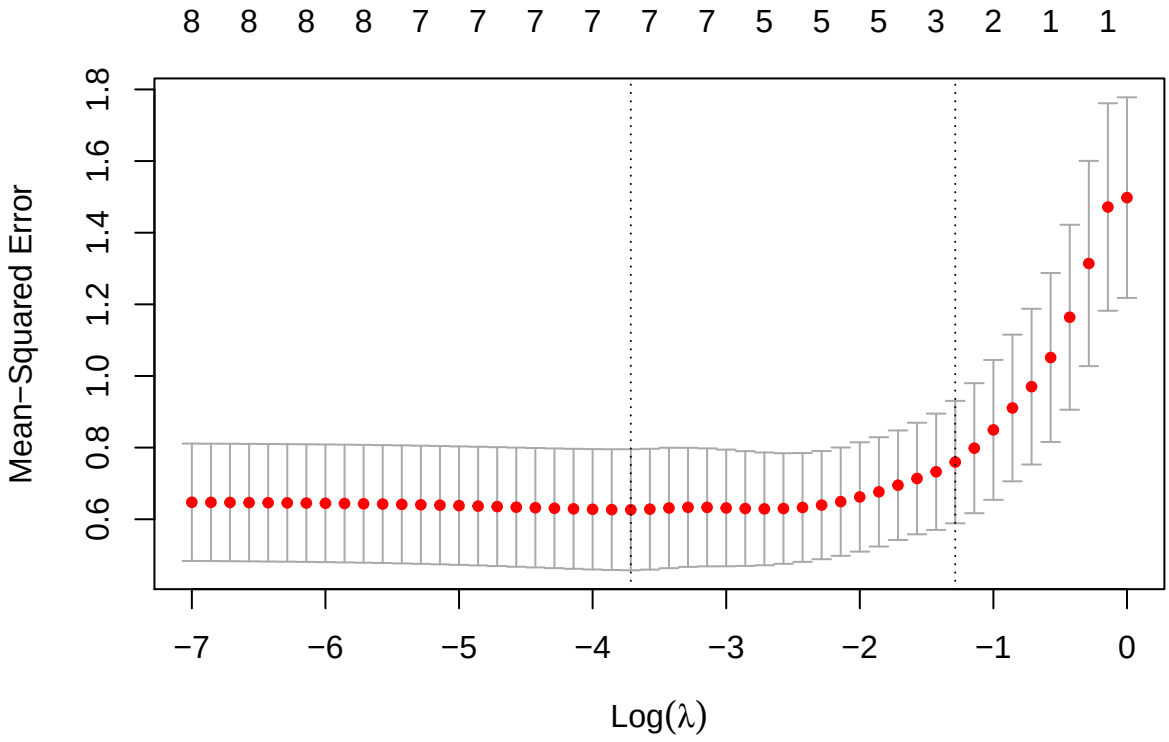
```
## [1] 0.1431235
```

The subset k that gives the lowest BIC is 2. The associated MSE on test data is 0.492, and the standard error for this RMSE on the test set is 0.143.

The model's variable names and coefficients can be found below:

```
## (Intercept)      lcavol     lweight
##  -1.0494396   0.6276074   0.7383751
```

c) LASSO regression with λ chosen by 5-fold cross-validation




```
## [1] 0.02437284
```

```
## [1] 0.5215968
```

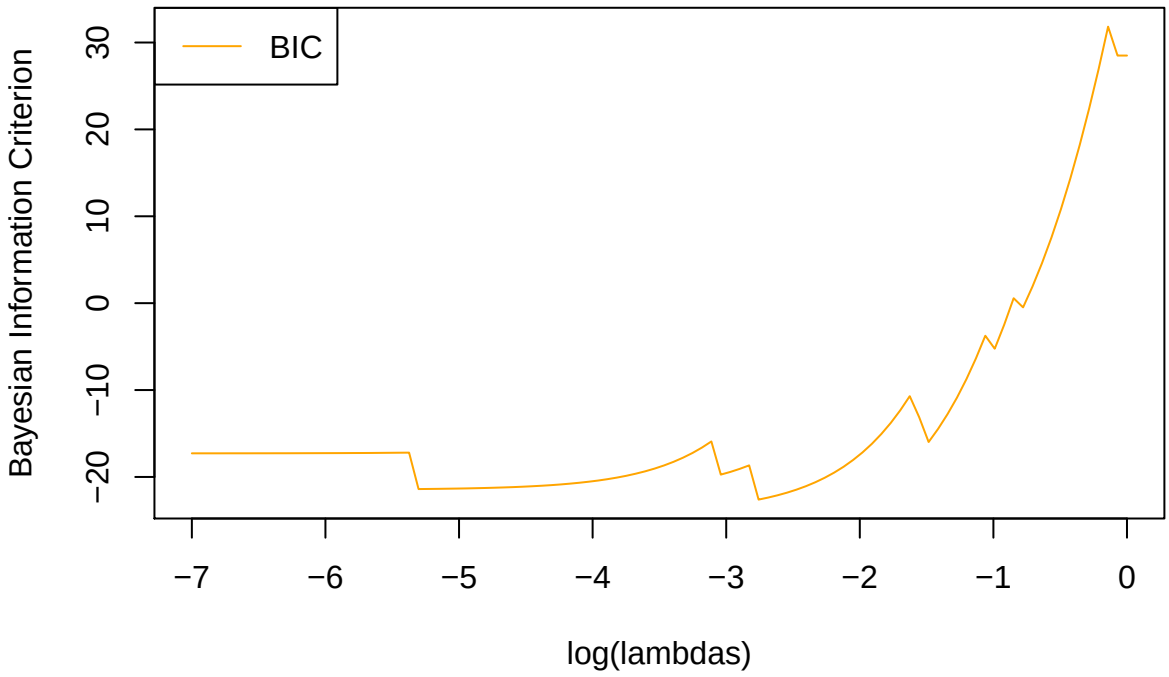
```
## [1] 0.1431235
```

The best lambda chosen for lasso that minimizes CV MSE is 0.0243. The associated CV MSE is 0.626. On test data, the MSE is 0.52, and the standard error for this RMSE on the test set is 0.143.

The model's variable names and coefficients can be found below:

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept) -0.523377591
## lcavol       0.519172509
## lweight      0.576435708
## age         -0.011305237
## lbph         0.126973431
## svi          0.609657579
## lcp         -0.094972940
## gleason      .
## pgg45        0.006090354
```

d) LASSO regression with λ chosen by BIC




```
## [1] 0.06344539
```

```
## [1] 0.4584993
```

```
## [1] 0.1480111
```

The lambda chosen to minimize BIC under cross-validation in 0.063. On test data, the MSE is 0.46, and the standard error for this RMSE on the test set is 0.148.

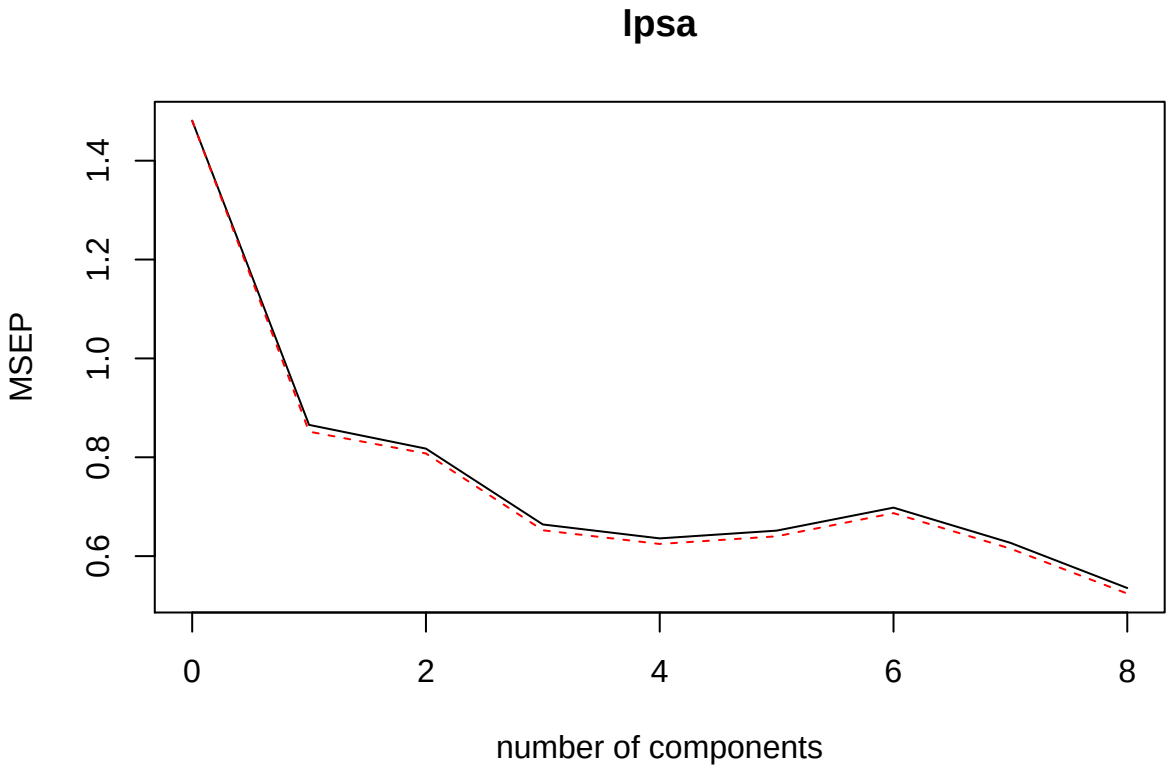
The selected model's variable names and estimated coefficients can be found below:

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) -0.682235658
## lcavol       0.466106795
## lweight      0.512708026
## age          .
## lbph         0.096019992
## svi          0.471698256
## lcp          .
## gleason      .
## pgg45        0.003009552
```

e) Principal component regression with q chosen by 5-fold cross-validation

```
## Data:    X dimension: 67 8
##  Y dimension: 67 1
## Fit method: svdpc
## Number of components considered: 8
##
## VALIDATION: RMSEP
## Cross-validated using 5 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           1.217   0.9303   0.9042   0.8149   0.7974   0.8072   0.8355
## adjCV        1.217   0.9230   0.8987   0.8076   0.7903   0.8001   0.8288
##        7 comps  8 comps
```

```
## CV       0.7918   0.7318
## adjCV    0.7844   0.7240
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       42.83    63.24    76.20    83.92    89.61    94.32    97.82   100.00
## lpsa    45.18    50.84    59.58    61.00    61.17    62.08    66.36    69.44
```




```
## [1] 0.521274
```

```
## [1] 0.178724
```

We see that the lowest CV error occurs when number of PCs is 8, which is the whole data. The MSE on the test set is 0.521, and the SE of the error on the test set is 0.178.

The model's all 8 PCs and coefficients can be found below:

```
## , , 8 comps
##
##                lpsa
## lcavol   0.71640701
## lweight  0.29264240
## age     -0.14254963
## lbph     0.21200760
## svi1     0.30961953
## lcp     -0.28900562
## gleason -0.02091352
## pgg45    0.27734595
```

Comments:

The first cross-validated best-subset model returns the optimal number of k as 7 variables. However, it does not perform as well as other models on the test set (test MSE 0.7). Since coefficient estimates of best-subset

is just the same as OLS, there might still be high variance and that reflects through poorer performance on test set. LASSO or PCR may be better when to goal is to minimize MSE under cross-validation, depending on the correlations between the predictor variables.

However, best subset can obtain smaller BIC than the LASSO with best lambda. This might be because the best subset model only has 2 variables (probably most meaningful variables in the dataset) and BIC penalizes for lack of parsimoniousness.

Some other observations include: PCR model does not have intercept because the design matrix has been transformed into PCs. If we center the predictor variables before fitting, LASSO model may also not have intercept term. The sparsity of the LASSO models varies depending on the selected lambda which is affected by the criterion of interest (MSE or BIC). Specifically, the LASSO model with BIC as the criterion is more sparse than the LASSO with tuned lambda and smallest CV error. Overall, the test MSE across the models are not too far apart (except for overfitting in best subset with k =7), same with the standard error of this test MSE statistic. In summary, some models might do better than others depending on the nature of the data (presence of multicollinearity,) and the criterion of interest (BIC, MSE, variable selection/sparse model, dimension reduction, etc.)

Appendix (codes used in the assignment)

```
#read in dataset
data = read.table("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/prostate.data") %>%
  mutate(svi = as.factor(svi))

#a) train = data %>% filter(train == "TRUE") %>% select(-train)
test = data %>% filter(train == "FALSE") %>% select(-train)
num_vars  = 8
set.seed(77)
folds = caret::createFolds(train$lpsa, k = 5)
fold_error = matrix(0, nrow = 5, ncol = num_vars,
                    dimnames = list(paste(1:5), paste(1:8)))

#function to predict outcome given the coefficients obtained by best subset
predict.regsubsets = function(object, newdata, id,...) {
  form  = as.formula(object$call[[2]])
  mat   = model.matrix(form, newdata)
  coefs = coef(object, id = id)
  xvars = names(coefs)
  mat[, xvars] %*% coefs}

#function to calculate RMSE
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))}

#for loop to get CV rmse under 5-fold CV
for(j in 1:5) {
  train_fold = train[-folds[[j]],]
  validate_fold = train[folds[[j]],]
  best_fit = regsubsets(lpsa ~ ., data = train_fold, nvmax = 8)
  for (i in 1:num_vars) {
    pred = predict(best_fit, validate_fold, id = i)
    fold_error[j, i] = rmse(actual = validate_fold$lpsa, predicted = pred)}}

#put error data together for visualization
```

```r
cv_error = tibble(cv_rmse = apply(fold_error, 2, mean),
                  rmse_sd = apply(fold_error, 2, sd), k = c(1:8))

#plot CV RMSE across k
ggplot(cv_error, aes(x = k, y = cv_rmse)) + geom_line() + geom_point() +
  geom_errorbar(aes(ymin = cv_rmse - rmse_sd, ymax = cv_rmse + rmse_sd),
                width = 0.2, position = position_dodge(0.05))

#calculate test error
#predict outcome y on test set
models <- regsubsets(lpsa~., data = train, nvmax = 8)
pred_subset = predict.regsubsets(models, test, id = 7)
test_error_subset = mean((test$lpsa - pred_subset)^2)
test_se_subset = sd((test$lpsa - pred_subset)^2)/sqrt(30)
test_se_subset

coef(models, 7)

#b) #all possible subsets
best_fit = regsubsets(lpsa ~ ., data = train_fold, nvmax = 8)
#summarize BICs associated with each subset
bic = tibble(bic = summary(best_fit)$bic, k = c(1:8))
#visualize
ggplot(bic, aes(x = k, y = bic)) + geom_line() + geom_point()

#use k=2, find rmse on the test set and standard error
pred_subset2 = predict.regsubsets(models, test, id = 2)
test_error_subset2 = mean((test$lpsa - pred_subset2)^2)
test_error_subset2
test_se_subset2 = sd((test$lpsa - pred_subset2)^2)/sqrt(30)
test_se_subset2

coef(models, 2)

#c) x_train = train %>% select(-lpsa)
y_train = train$lpsa

set.seed(13)
cv.lasso <- cv.glmnet(data.matrix(x_train), y = y_train,
                      nfolds = 5, family = "gaussian",
                      type.measure = "mse", alpha = 1,
                      standardize = T, #standardize using built-in function
                      lambda = exp(seq(-7, -0, length=50)))

plot(cv.lasso)

cv.lasso$lambda.min #best lambda
mse.min <- min(cv.lasso$cvm) #cv mse
mse.min
pred_lasso = predict(cv.lasso, data.matrix(test[,-9]), lambda = cv.lasso$lambda.min)
rmse_lasso = mean((test$lpsa - pred_lasso)^2)
rmse_lasso
test_se_lassocv = sd((test$lpsa - pred_subset2)^2)/(sqrt(30))
```

```r
test_se_lassocv

coef(cv.lasso, cv.lasso$lambda.min)

#d) lambdas = exp(seq(-7, -0, length=100))
bic <- c()
for (i in seq(lambdas)) {
  lasso.bic = ic.glmnet(data.matrix(x_train), data.matrix(y_train),
                        alpha = 1, lambda = lambdas[i], crit = "bic")
  bic[i] = lasso.bic$ic[1]}

# Plot information criteria against tried values of lambdas
plot(log(lambdas), bic, col = "orange", type = "l", ylab = "Bayesian Information Criterion")
legend("topleft", lwd = 1, col = c("orange"), legend = c("BIC"))

lambda_bic <- lambdas[which.min(bic)]
lambda_bic

#refit LASSO with chosen Lambda
set.seed(13)
lasso.bic.mod = glmnet(data.matrix(x_train), y_train,
                       alpha = 1, lambda = lambda_bic, standardize = TRUE)

#predict on test data
lasso_bic_pred = predict(lasso.bic.mod , data.matrix(test[,-9]))
mean((test$lpsa - lasso_bic_pred)^2) #MSE
sd((test$lpsa - lasso_bic_pred)^2)/(sqrt(30)) #MSE SE
coef(lasso.bic.mod, lambda_bic)

#e) PCR
pcr.mod = pcr(lpsa ~ ., data = train,
              scale = TRUE, validation = "CV",
              segments = 5, segment.type = "random")

summary(pcr.mod)
validationplot(pcr.mod, val.type = "MSEP")

pcr_pred = predict(pcr.mod, test, ncomp=8)
mean((test$lpsa - pcr_pred)^2) #MSE
sd((test$lpsa - pcr_pred)^2)/(sqrt(30))

coef(pcr.mod)
```