Problem set 2

- **Problem 1 (6 pts)**
  In this problem we compare linear predictors in terms of their training MSEs.

  Suppose we have n training data (y_i, x_i) for $1 \leq i \leq n$. Here each x_i contains values of three features, i.e. $x_i = (x_{i1}, x_{i2}, x_{i3})^\top$. (We think these x_i's as realizations of a random vector $X = (X_1, X_2, X_3)^\top$.)

  Consider the following linear predictor that uses the feature X_1 and X_2,

  $$\hat{f}_1(x) = \hat{\alpha}_1 x_1 + \hat{\alpha}_2 x_2, \qquad \text{for any } x = (x_1, x_2, x_3)^\top.$$

  Here the coefficients $\hat{\alpha}_1$ and $\hat{\alpha}_2$ are computed by OLS, that is,

  $$(\hat{\alpha}_1, \hat{\alpha}_2) = \underset{\alpha_1, \alpha_2 \in \mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \alpha_1 x_{i1} - \alpha_2 x_{i2})^2. \tag{0.1}$$

  Similarly, consider another linear predictor that uses (X_1, X_2, X_3), i.e.

  $$\hat{f}_2(x) = \hat{\gamma}_1 x_1 + \hat{\gamma}_2 x_2 + \hat{\gamma}_3 x_3, \qquad \text{for any } x = (x_1, x_2, x_3)^\top.$$

  where the coefficients $(\hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3)$ are computed by OLS as well.

  1. (**2 pts**) Suppose the true model of (y_i, x_i) is

     $$Y = \beta_1 X_1 + \beta_2 X_2 + \epsilon.$$

     Recall that for each predictor $\hat{f}_j$ with $j \in \{1, 2\}$, its training MSE is defined as

     $$\text{MSE}(\hat{f}_j) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}_j(x_i))^2. \tag{0.2}$$

     Is there enough information to compare MSE($\hat{f}_1$) and MSE($\hat{f}_2$)? If so, give your conclusion and prove it. Otherwise, state your reasoning.
     (Hint: try to use the optimality of the OLS estimates from (0.1).)

  2. (**2 pts**) Suppose the true model of (y_i, x_i) is

     $$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon.$$

     Would you expect the same conclusion as part 1? Justify your answer.

  3. (**2 pts**) Suppose instead of $\hat{f}_2$, we compare $\hat{f}_1$ with the following linear predictor that uses X_1 and X_3:

     $$\hat{f}_3(x) = \hat{\beta}_1 x_1 + \hat{\beta}_2 x_3, \qquad \text{for any } x = (x_1, x_2, x_3)^\top.$$

     Once again, $(\hat{\beta}_1, \hat{\beta}_2)$ are computed by OLS. Suppose the true model of (y_i, x_i) is

     $$Y = \beta_1 X_1 + \beta_2 X_2 + \epsilon.$$

     Is there enough information to compare MSE($\hat{f}_1$) and MSE($\hat{f}_3$)? If so, give your conclusion and prove it. Otherwise, state your reasoning.
     (Hint: try to use the optimality of the OLS estimates from (0.1).)

Problem 2 (9 pts)

Now let's design some simulation study to verify our answers to **Problem 1**. First we set the seed to 0 for reproducibility: `set.seed(0)`.

Let us generate (y_i, x_i) in the following way.

(i) Generate $x_i = (x_{i1}, x_{i2}, x_{i3})^\top$ with $1 \le i \le n$ i.i.d. from $N(\mu, \Sigma)$ with

$$\mu = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \rho \\ 0 & \rho & 1 \end{bmatrix}.$$

Here $\rho \in [0, 1]$ represents the correlation between X_2 and X_3. This requires to simulate random vectors from a multivariate normal distribution. You might find the function `mvrnorm` in the R-package `MASS` useful.

(ii) Generate ϵ_i with $1 \le i \le n$ i.i.d. from $N(0, 1)$.

(iii) Generate y_i as

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i, \qquad 1 \le i \le n,$$

with $\beta_1 = \beta_2 = 0.5$.

We examine the training MSEs of each predictor in **Problem 1**.

1. (**3 pts**) Follow steps (i) – (iii) to generate the training data with $\rho = 0.1$ and $n = 100$. Fit your predictors $\hat{f}_j$ with $j \in \{1, 2, 3\}$. Then compute and compare the training MSEs of $\hat{f}_j$ with $j \in \{1, 2, 3\}$ as in (0.2). Did you get the same conclusions as in **Problem 1**?

2. (**3 pts**) Repeat part 1 above $N = 100$ times, meaning that you generate N training data sets and computing N training MSEs for each predictor. (Note that you should NOT set seed for each repetition.)

   Did the comparison of training MSEs vary across repetitions? Comment on your findings.

3. (**3 pts**) Now repeat part 2 above for $\rho = 0.95$. Comment on the differences you find comparing to $\rho = 0.1$.

- **Problem 3 (17 pts)**

  We have seen that as the number of features used in a model increases, the training error will necessarily decrease, but the test error may not. We will now explore this in a simulated data set.

  1. (**2 pts**) Generate a data set with $p = 20$ features, $n = 1000$ observations, and an associated quantitative response vector $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ generated according to the model

     $$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

     where $\boldsymbol{\beta} \in \mathbb{R}^p$ satisfies $\beta_1 = \beta_2 = \beta_3 = 2, \beta_4 = \beta_5 = 0.5$ and $\beta_6 = \cdots = \beta_{20} = 0$. The design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ has entries generated as i.i.d. realizations of $N(0, 1)$. The error $\boldsymbol{\epsilon} \in \mathbb{R}^n$ also contains entries generated as i.i.d. realizations of $N(0, 1)$.

     (**Note:** for reproducibility, you need to specify `set.seed(0)` at the beginning before generating the data.)

  2. (**1 pts**) Randomly split your dataset into a training set containing 100 observations and a test set containing 900 observations.

  3. (**2 pts**) Perform best subset selection on the training set, and plot the training set MSE associated with the best model of each size.

  4. (**2 pts**) Plot the test set MSE associated with the best model of each size.

  5. (**2 pts**) For which model size do the training set MSE and test set MSE take on their minimum value? Comment on your results.

  6. (**2 pts**) How does the model at which the test set MSE is minimized compare to the true model used to generate the data? Comment on the coefficient values.

  7. (**2 pts**) Create a plot displaying

     $$\sqrt{\sum_{j=1}^{p} (\beta_j - \hat{\beta}_j^{(k)})^2}$$

     for a range of values of k, where $\hat{\beta}_j^{(k)}$ is the jth coefficient estimate for the best model containing k coefficients. Comment on what you observe. How does this compare to the test MSE plot from part 4?

  8. (**2 pts**) Repeat parts 3 - 6 for forward stepwise selection.

  9. (**2 pts**) Repeat parts 3 - 6 for backward stepwise selection.

- **Problem 4 (12 pts)**

  In this problem you will compare the performance of lasso and ridge regression in different linear models. Consider $p = 50$ and $n = 1100$.

  (a) Set the random seed by using `set.seed(0)` and generate the design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and the error $\boldsymbol{\epsilon} \in \mathbb{R}^n$ as part 1 of Problem 4.

  (b) Generate the response $\mathbf{y} \in \mathbb{R}^n$ based on the model

  $$y_i = \sum_{k=1}^{p} \beta_k X_{ik} + \epsilon_i, \qquad \forall i = 1, \ldots, n,$$

  with $\beta_1 = \cdots = \beta_5 = 2$, and $\beta_j = 0$ for $j \geq 6$.

  (c) Randomly split the data into a training set with 100 observations and a test set containing 1000 observations.

  (d) Set the grid of λ by using the R command

  ```
  grid = 10^seq(10,-2,length = 100)
  ```

  1. (**3 pts**) Fit both the ridge regression and the lasso with λ selected by cross validation on the `grid` generated as above. Which method leads to a smaller test set MSE?

  2. (**3 pts**) Repeat steps (a)–(d) for generating the data by using different seeds

     `set.seed(2),...,set.seed(50)`

     and also repeat part 1 for each seed. Save the test error for both, lasso and ridge for all seeds. Together with the results from part 1, this should give you 50 test MSEs for ridge and lasso. Make boxplots of the test errors for these two procedures and comment on the results.

  3. (**6 pts**) Redo parts 1 and 2 by using $\beta_j = 0.5$ for all $j = 1, \ldots, 50$, in step (b).