NOTES. There are three sections of the homework. Section 1 and Section 2 are required for all students. While Section 3 is only required for Ph.D. students in Statistics Department, you are all encouraged to try and earn bonus points. All theoretical problems are to be done individually. Problems in Section 2 can be done in a group of no more than 5 students. A group must have at most one Ph.D. student. I strongly suggest those who are not familiar with `R` to find someone who is a good `R` programmer to form a group. Keep the code you develop as you and your group may be asked to present your work later.

1. Theoretical Problems

1. *Forward stepwise regression.* Suppose we have the QR decomposition for the $N \times q$ matrix $\boldsymbol{X}_1$, and we have an additional $p-q$ predictors in the matrix $\boldsymbol{X}_2$. We wish to establish which one of these additional features will reduce the residual sum of squares most when included with those in $\boldsymbol{X}_1$. Describe an efficient procedure for doing this.

Solution. The columns of the matrix $\boldsymbol{X}_1$ need not be the first q columns of the original input matrix $\boldsymbol{X}$, but we can always relabel the columns, so without loss of generality, we assume $\boldsymbol{X}_1 = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q)$. Suppose we have the QR decomposition $\boldsymbol{X}_1 = \boldsymbol{Q}_1\boldsymbol{R}_1$, where $\boldsymbol{Q}_1 = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q)$ is an orthogonal matrix (columns are orthonormal), and $\boldsymbol{R}_1$ is an upper triangular matrix. For any matrix $\boldsymbol{A}$, let span$(\boldsymbol{A})$ be the linear space spanned by its columns. The key observation is that span$(\boldsymbol{X}_1) =$ span$(\boldsymbol{Q}_1)$.

Now let us evaluate the contribution of some other predictor $\boldsymbol{x}_k$, where $q < k \le p$. The projection of $\boldsymbol{x}_k$ on span$(\boldsymbol{X}_1)$, denoted by $\mathcal{P}_{\boldsymbol{X}_1}(\boldsymbol{x}_k)$, is given by

$$\mathcal{P}_{\boldsymbol{X}_1}(\boldsymbol{x}_k) = \langle \boldsymbol{x}_k, \boldsymbol{w}_1\rangle \boldsymbol{w}_1 + \cdots + \langle \boldsymbol{x}_k, \boldsymbol{w}_q\rangle \boldsymbol{w}_q. \tag{1.1}$$

Let

$$\boldsymbol{r}_k = \boldsymbol{x}_j - \mathcal{P}_{\boldsymbol{X}_1}(\boldsymbol{x}_k) \quad \text{and} \quad \boldsymbol{w}_k = \boldsymbol{r}_k / \|\boldsymbol{r}_k\|. \tag{1.2}$$

The current fitted values is the projection of $\boldsymbol{y}$ onto span$(\boldsymbol{X}_1)$, denoted by $\hat{\boldsymbol{y}}_1 = \mathcal{P}_{\boldsymbol{X}_1}(\boldsymbol{y})$. If we add $\boldsymbol{x}_k$ into the current set of predictors, the fitted values will become

$$\hat{\boldsymbol{y}}_1 + \langle \boldsymbol{y}, \boldsymbol{w}_k\rangle \boldsymbol{w}_k,$$

and hence the current residual sum of squares is reduced by

$$\langle \boldsymbol{y}, \boldsymbol{w}_k\rangle^2. \tag{1.3}$$

Therefore, in order to find an additional predictor which improve the current fit the most, we need to find the one which maximizes the quantity in (1.3).

Now the problem boils down to computing (1.3) for every $q < k \le p$, and then finding out the maxima. Since we have already known $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q$, we can go through the steps (1.1), (1.2) and (1.3) using an algorithm.

Here is the final observation. W.L.O.G., let us assume it is the predictor $\boldsymbol{x}_{q+1}$ which maximizes the quantity (1.3). By adding $\boldsymbol{x}_{q+1}$, the current set of predictors is given by the matrix $(\boldsymbol{X}_1, \boldsymbol{x}_{q+1})$. Since we have already computed (1.1) and (1.2), we also know the QR decomposition of $(\boldsymbol{X}_1, \boldsymbol{x}_{q+1})$ is given by

$$(\boldsymbol{X}_1, \boldsymbol{x}_{q+1}) = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q, \boldsymbol{w}_{q+1})\tilde{\boldsymbol{R}},$$

where $\tilde{\boldsymbol{R}}$ is a $(q+1) \times (q+1)$ upper triangular matrix. You can work out the entries of $\tilde{\boldsymbol{R}}$ from $\boldsymbol{R}_1$ and (1.1) and (1.2). But for the purpose of forward stepwise selection, the matrix $\tilde{\boldsymbol{R}}$ is not important. As long as we have the orthonormal columns

$$\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q, \boldsymbol{w}_{q+1}$$

we can cycle through (1.1)–(1.3) to add one more predictor and keep going.

2. *Backward stepwise regression.* Suppose we have the multiple regression fit of $\boldsymbol{y}$ on $\boldsymbol{X}$. We want to find a variable, when dropped, will increase the residual sum of squares the least. Show that the variable with the smallest absolute value of Z-score is the right one to drop.

Solution. According to the discussion of Section 3.2.3 of ESL, if we have already included $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{p-1}$, then by adding $\boldsymbol{x}_p$, the residual sum of squares is reduced by

$$\frac{\langle \boldsymbol{y}, \boldsymbol{z}_p \rangle^2}{\|\boldsymbol{z}_p\|^2}. \tag{1.4}$$

On the other hand, by (3.28) and (3.29) of ESL, the Z-score for the coefficient $\hat{\beta}_p$ is given by

$$Z_p = \frac{\langle \boldsymbol{y}, \boldsymbol{z}_p \rangle}{\hat{\sigma}\|\boldsymbol{z}_p\|}. \tag{1.5}$$

It follows that the absolute value $|Z_p|$ is a "proxy" of the quantity (1.4). This argument is true for every predictor $\boldsymbol{x}_j$ if it is the last one we add in. Therefore, the variable with the smallest absolute value of Z-score is the right one to drop.

3. *Maximum likelihood estimate.* Suppose we have a set of training data $(x_1, y_1), \ldots, (x_N, y_N)$ coming from the linear model:

$$y_i = x_i^T \beta + \epsilon_i, \quad 1 \le i \le N.$$

Assume x_i's are fixed, and ϵ_i's are independent and identically distributed (i.i.d.) as $N(0, \sigma^2)$. The *likelihood function* is defined as

$$L(\beta, \sigma^2) = \prod_{i=1}^N \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - x_i^T \beta)^2}{2\sigma^2} \right] \right\}.$$

The estimate $(\tilde{\beta}, \tilde{\sigma}^2)$ which maximizes the likelihood function $L(\beta, \sigma^2)$ is called *maximum likelihood estimate.* Show that $\tilde{\beta}$ is the same as the least square estimate $\hat{\beta}$. Also find an expression for $\tilde{\sigma}^2$.

Solution. It is often easier to work with log-likelihood

$$l(\beta, \sigma^2) = \log\left[L(\beta, \sigma^2)\right] = -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - x_i^T \beta)^2 - \frac{N}{2} \log(2\pi\sigma^2).$$

The first observation is no matter what value σ^2 takes, $\tilde{\beta}$ has to minimize the function

$$\sum_{i=1}^N (y_i - x_i^T \beta)^2,$$

which is the residual sum of squares, so $\tilde{\beta}$ must be the same as the least square estimate $\hat{\beta}$. To find $\tilde{\sigma}^2$, we maximize

$$l(\hat{\beta}, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - x_i^T \hat{\beta})^2 - \frac{N}{2} \log(2\pi\sigma^2)$$

as a function of σ^2. By setting the derivative to zero

$$\frac{\partial l(\hat{\beta}, \sigma^2)}{\partial \sigma^2} = \frac{1}{2(\sigma^2)^2} \sum_{i=1}^N (y_i - x_i^T \hat{\beta})^2 - \frac{N}{2\sigma^2} = 0,$$

we obtain

$$\tilde{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N (y_i - x_i^T \hat{\beta})^2.$$

Note that the MLE of σ^2 is different from the unbiased estimate $\hat{\sigma}^2$ given in the middle of page 47 of ESL.

4 (Gauss Markov Theorem)**.** Prove the matrix version of the Gauss-Markov Theorem as presented in the lecture. [Hint: We say a symmetric $p \times p$ matrix A is *positive semi-definite* if $\alpha^T A \alpha \ge 0$ for all $\alpha \in \mathbb{R}^p$.]

Solution. Suppose $\tilde{\beta}$ is a LUE of β, then it must take the form $\tilde{\beta} = \boldsymbol{C}^T\boldsymbol{y}$, where $\boldsymbol{C}$ is a $N \times p$ matrix; and satisfies $\mathbb{E}\tilde{\beta} = \beta$ for all $\beta \in \mathbb{R}^p$. We need to show the following matrix

$$\text{Var}(\tilde{\beta}) - \text{Var}(\hat{\beta})$$

is positive semi-definite. By definition, we need to show

$$\alpha^T \left[\text{Var}(\tilde{\beta}) - \text{Var}(\hat{\beta})\right] \alpha \geq 0 \tag{1.6}$$

for all $a \in \mathbb{R}^p$. Now let $\theta = \alpha^T\beta$. We see that both $\alpha^T\hat{\beta}$ and $\alpha^T\tilde{\beta}$ are LUE for θ. According to the univariate Gauss-Markov theorem, we must have

$$\text{Var}(\alpha^T\tilde{\beta}) \geq \text{Var}(\alpha^T\hat{\beta}).$$

The proof of (1.6) is completed by noting that

$$\text{Var}(\alpha^T\tilde{\beta}) = \alpha^T\text{Var}(\tilde{\beta})\alpha \quad \text{and} \quad \text{Var}(\alpha^T\hat{\beta}) = \alpha^T\text{Var}(\hat{\beta})\alpha.$$

5. *Principal components regression.* Let $\boldsymbol{X}^c$ be the centered input matrix, that is, every column sum is zero. Let $\boldsymbol{X}^c = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ be the singular value decomposition of $\boldsymbol{X}^c$. Let $v_1, \ldots, v_p$ be columns of the matrix $\boldsymbol{V}$. Define the principle components $\boldsymbol{z}_j = \boldsymbol{X}^c v_j$ for $1 \leq j \leq p$. *Principal component regression* uses the first M principal components $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M$ as the predictors and uses the least square method to produce the fitted values

$$\hat{\boldsymbol{y}}_{(M)}^{\text{pcr}} = \bar{y}\mathbf{1} + \sum_{m=1}^{M} \hat{\theta}_m \boldsymbol{z}_m,$$

where $\bar{y} = \frac{1}{N} y_i$ is the sample mean of outputs, and $\mathbf{1}$ is a N-dimenional vector with all entries equal to one. The fitted value can also be expressed as a linear combination of the original precidtors

$$\hat{\boldsymbol{y}}_{(M)}^{\text{pcr}} = \bar{y}\mathbf{1} + \boldsymbol{X}^c \hat{\beta}_{(M)}^{\text{pcr}}.$$

(a) Find out the estimated coefficients $\hat{\theta}_m$ in terms of $\boldsymbol{y}$ and the principal components.

(b) Show that

$$\hat{\beta}_{(M)}^{\text{pcr}} = \sum_{m=1}^{M} \hat{\theta}_m v_m.$$

(c) Find the relationship between the least square estimate $\hat{\beta} = ((\boldsymbol{X}^c)^T\boldsymbol{X}^c)^{-1}(\boldsymbol{X}^c)^T\boldsymbol{y}$ and $\hat{\beta}_{(p)}^{\text{pcr}}$, which is the obtained when all the principal components are used.

(d) Compare pricipal component regression and ridige regression in terms of different ways they shrink the least square estimates.

Solution.

(a) Because $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_p$ are orthogonal, we have

$$\hat{\theta}_m = \frac{\langle \boldsymbol{y}, \boldsymbol{z}_m \rangle}{\langle \boldsymbol{z}_m, \boldsymbol{z}_m \rangle}.$$

(b) Since $\boldsymbol{z}_m = \boldsymbol{X}^c v_m$, we have

$$\sum_{m=1}^{M} \hat{\theta}_m \boldsymbol{z}_m = \sum_{m=1}^{M} \hat{\theta}_m \boldsymbol{X}^c v_m = \boldsymbol{X}^c \sum_{m=1}^{M} \hat{\theta}_m v_m.$$

(c) In principle they should be the same, but here is a direct argument. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p$ be the columns of the matrix $\boldsymbol{U}$, and $d_1 \geq d_2 \geq \cdots \geq d_p > 0$ be the diagonal entries of $\boldsymbol{D}$. Observe that using the SVD, the least square estimate can be written as

$$\hat{\beta} = \boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}^T\boldsymbol{y} = \sum_{m=1}^{p} \frac{\langle \boldsymbol{y}, \boldsymbol{u}_m \rangle}{d_m} v_m$$

Since $\boldsymbol{z}_m = \boldsymbol{X}^c v_m = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T v_m = d_m \boldsymbol{u}_m$, by part (b) and part (a), we know

$$\hat{\beta}^{\text{pcr}}_{(p)} = \sum_{m=1}^{p} \hat{\theta}_m v_m = \sum_{m=1}^{p} \frac{\langle \boldsymbol{y}, \boldsymbol{z}_m \rangle}{\langle \boldsymbol{z}_m, \boldsymbol{z}_m \rangle} v_m = \sum_{m=1}^{p} \frac{\langle \boldsymbol{y}, d_m \boldsymbol{u}_m \rangle}{\langle d_m \boldsymbol{u}_m, d_m \boldsymbol{u}_m \rangle} v_m = \sum_{m=1}^{p} \frac{\langle \boldsymbol{y}, \boldsymbol{u}_m \rangle}{d_m} v_m.$$

(d) Comparing equation (3.47) of ESL, we see that principal components regression is very similar to ridge regression: both operate via the principal components of the input matrix. Ridge regression shrinks the coefficients of the principal components, shrinking more depending on the size of the corresponding singular value; principal components regression discards the $p - M$ principal components corresponding to the $p - M$ smallest singular values. Principal component regression can be viewed as more "extreme" version of ridge regression.

6. *Coordinate descent.* Suppose we have a *cost function* $J(\theta)$ which depends on the parameter $\theta = (\theta_1, \dots, \theta_p)^T$, and we want to choose a θ to minimize the cost function. The *gradient descent* algorithm starts with some "initial guess" value for θ, and repeatedly performs the update

$$\theta_j := \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} J(\theta) \quad 1 \le j \le p.$$

Here $\alpha > 0$ is called the *learning rate*, which needs to be chosen suitably. This is a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of J. Note that the update is simultaneously performed for all values of j.

The *coordinate descent* algorithm uses the same idea as gradient descent, but each time it only updates one θ_j, while all other θ_k, $k \ne j$ are held as fixed. Let us use Lasso as an example. The cost function is

$$J(\beta) = \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where we have suppressed the intercept for convenience. Denote by $\tilde{\beta}_k(\lambda)$ the current estimate for β_k at penalty level λ. Suppose we now want to perform an update on β_j, we can rewrite the cost function to isolate β_j,

$$J(\beta_j) = \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \sum_{k \ne j}^{p} x_{ik} \tilde{\beta}_k(\lambda) - x_{ij}\beta_j \right)^2 + \lambda \sum_{k \ne j} |\tilde{\beta}_k(\lambda)| + \lambda |\beta_j|.$$

Here by an abuse of notation, the function J has only one argument β_j because we view other $\tilde{\beta}_k(\lambda)$, $k \ne j$ as fixed. We want to perform an update on β_j such that the function $J(\beta_j)$ is minimized.

(a) Show that this can be viewed as a univariate Lasso problem by identifying the new "output", "input", as well as the penalty parameter.

(b) Assume that the output is centered, and the predictors are normalized. Find the updated value of β_j so that the function $J(\beta_j)$ is minimized. [Hint. If you cannot find it, then derive equation (3.84) of ESL.]

To conclude the algorithm, we cycle through each variable β_j in turn until convergence to the Lasso estimate $\hat{\beta}^{\text{lasso}}$, which probably takes several rounds.

Solution.

(a) We can view $y_i - \sum_{k \ne j}^{p} x_{ik} \tilde{\beta}_k(\lambda)$ as the new output, x_{ij} as the univariate input on the object i, for each $1 \le i \le N$. The penalty parameter is still λ.

(b) Let us consider the minimization problem with the cost function

$$Q(t) = at^2 - 2bt + 2c|t|,$$

where $a > 0$, $b \in \mathbb{R}$ and $c \ge 0$ are constants. We first the minimizer on the positive half line $t \ge 0$. When $t \ge 0$, the function becomes

$$Q^+(t) = at^2 - 2(b - c)t.$$

Depending on whether $b - c \geq 0$ or not, the minimizer of $Q^+(t)$ subject to $t \geq 0$ is given by $b - c$ or 0 respectively. Similarly, when $t \leq 0$, the cost function becomes

$$Q^-(t) = at^2 - 2(b + c)t.$$

Depending on whether $b + c \leq 0$ or not, the minimizer of $Q^-(t)$ subject to $t \leq 0$ is given by $b + c$ or 0 respectively. Combining these two cases, we see that the minimizer of $Q(t)$ is given by

$$\tilde{t} = \begin{cases} b + c & \text{if } b + c < 0; \\ b - c & \text{if } b - c > 0; \\ 0 & \text{if } -c \leq b \leq c. \end{cases}.$$

The solution $\tilde{t}$ can be written in the compact form

$$\tilde{t} = S(b, c);$$

where $S(b, c)$ is defined as $S(b, c) = \text{sign}(b)(|b| - c)_+$. Here for any real number t, $(t)_+$ denotes its positive part, that is, $(t)_+ = t$ when $t \geq 0$ and $(t)_+ = 0$ when $t < 0$.

Now let us go back to our Lasso problem. We only need to note that we can write $2J(\beta_j) = a\beta_j^2 - 2b\beta_j + 2c|\beta_j| + R$, where R is a remainder term which does not involve β_j, and

$$a = \sum_{i=1}^{N} x_{ij}^2 = 1, \quad b = \sum_{i=1}^{N} \left\{ x_{ij} \left(y_i - \sum_{k \neq j}^{p} x_{ik} \tilde{\beta}_k(\lambda) \right) \right\}, \quad \text{and } c = \lambda.$$

2. Data Analysis

7. This problem uses two data sets `hw1-netflix-training.csv` and `hw1-netflix-test.csv`. There are small subsets of the data from the 'Netflix competition'. Neflix is a movie rental company that invites their customers to rate movies they watched. The goal of the competition was to use the data collected by Netflix to predict ratings for yet unavailable customer–movie pairs.

The training data comprises four colums: a number identifying a movie, a number identifying a customer, the rating the customer gave to the movie (an integer between 1 and 5), and the date that the rating was given. The test data is similar but without the date. A third file gives the years the movies came out and their names (not necessarily of direct use for prediction tasks).

In this problem you are asked to propose, implement and evaluate nearest neighbor methods for predicting the movie ratings for the test data. All your decisions of model choice and tuning of estimation techniques have to be made using only the training data.

8. This problem uses the `BostonHousing2` data available at `http://lib.stat.cmu.edu/datasets/boston_corrected.txt`. It contains 506 census tracts of Boston from the 1970 census. Each row represents a town. There are 21 columns, among which we will use 14 variables listed below. The variable `CMEDV` is the output, and the rest are predictors.

CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
B	$1000(B - 0.63)^2$ where B is the proportion of blacks by town
LSTAT	% lower status of the population
CMEDV	Median value of owner-occupied homes in $1000's

(a) Use the procedure you developed for Problem 1 to perform the forward stepwise selection. Present your result as a sequence of predictors according to the order they enter into the model. Remember to start with the intercept. [Note. You should write your own code instead of using any built in R function.]

(b) Use the procedure you developed for Problem 2 to perform the backward stepwise selection. Present your result as a sequence of predictors according to the reverse order they get dropped. Remember to start with the full model with the intercept.

(c) Implement Lasso to produce a sequence of predictors according to the order they enter into the model. This time you can use the built in function `lars` in the R package `lars`. Remember to include the intercept.

(d) Implement LARS to produce a sequence of predictors according to the order they enter into the model. Remember to include the intercept.

(f) Comment on your findings.

9. This problem uses a dataset `diabetes` that is available in R package `lars`, which you can load using the `data()` function. There are 10 baseline predictors: age, sex, body mass index, average blood pressure and six blood serum measurements for $N = 442$ diabetes patients. The output/response of interest is a quantitative measurement of disease progression one year after baseline. The variable `x` is a 442×10 matrix, with each row representing a patient, and each column representing a predictor. The variable `x2` is a 442×64 matrix, corresponding to the "Quadratic Model" with $p = 64$ predictors, including interactions and squares of the 10 original predictors:

Quadratic Model: 10 main effects, 45 interactions, 9 squares.

We only have 9 squares because the predictor "sex" is dichotomous, so we do not include its square. For this problem, we consider this "Quadratic Model" with $p = 64$ predictors.

(a) Run LARS for 10 steps to obtain an estimate β^0. Set $\boldsymbol{\mu_0} = \boldsymbol{X}\beta^0$, and the residual $\boldsymbol{\epsilon}_0 = \boldsymbol{y} - \boldsymbol{\mu}_0$. From now on we view $\boldsymbol{\mu}_0$ as the true mean vector. Report your estimate β^0.

(b) Generate a simulated output vector $\boldsymbol{y}^*$ from the model

$$\boldsymbol{y}^* = \boldsymbol{\mu}_0 + \boldsymbol{\epsilon}^*,$$

where $\boldsymbol{\epsilon}^* = (\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_N^*)^T$ is a random sample, with replacement, from the components of $\boldsymbol{\epsilon}_0$.

(c) Run LARS with $K = 40$ steps. For each $1 \leq k \leq K$, record your estimated mean vector $\boldsymbol{\mu}^{k*}$, and compute the *proportion explained* by $\boldsymbol{\mu}^{k*}$

$$\mathrm{pe}(\boldsymbol{\mu}^{k*}) = 1 - \frac{\|\boldsymbol{\mu}^{k*} - \boldsymbol{\mu}_0\|^2}{\|\boldsymbol{\mu}_0\|^2}.$$

(d) Run Lasso with $K = 40$ steps. For each $1 \leq k \leq K$, record your estimated mean vector $\boldsymbol{\mu}^{k*}$ as well as the estimated parameter β^{k*}, and compute the *proportion explained* $\mathrm{pe}(\boldsymbol{\mu}^{k*})$.

(e) Run forward stepwise selection with $K = 40$ steps. For each $1 \leq k \leq K$, record your estimated mean vector $\boldsymbol{\mu}^{k*}$, and compute the *proportion explained* $\mathrm{pe}(\boldsymbol{\mu}^{k*})$.

(f) Repeat the Steps (b)–(e) for $B = 100$ times.

(g) For each $1 \leq k \leq K$, compute the average of $\mathrm{pe}(\boldsymbol{\mu}^{k*})$ that you obtained from LARS over the 100 simulations. Plot this average number versus step number k.

(h) For each $1 \leq k \leq K$, compute the average of $\mathrm{pe}(\boldsymbol{\mu}^{k*})$ that you obtained from Lasso over the 100 simulations; also compute the average number of nonzero coefficient β^{k*}. In the same graph, plot the average of pe versus the average number of nonzero coefficients.

(i) For each $1 \leq k \leq K$, compute the average of $\mathrm{pe}(\boldsymbol{\mu}^{k*})$ that you obtained from forward stepwise selection over the 100 simulations. In the same graph, plot the average of pe versus step number k.

(j) Comment on your findings.

3. Theoretical Problems for Ph.D. Students

10. Consider a linear regression model with p parameters, fit by least squares to a set of training data $(x_1, y_1), \ldots, (x_N, y_N)$ drawn at random from a population. Statistically, this means that the random vectors

(x_i, y_i) are independent and identically distributed. Suppose we also have some test data $(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_M, \tilde{y}_M)$ drawn at random from the same population as the training data. Assume

$$\mathbb{E}(y_1|x_1) = x_1^T\beta \quad \text{and} \quad \mathrm{Var}(y_1|x_1) = \sigma^2.$$

Define

$$R_{\mathrm{tr}}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta^T x_i)^2 \quad \text{and} \quad R_{\mathrm{te}}(\beta) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \beta^T \tilde{x}_i)^2.$$

Let $\hat{\beta}$ be the least square estimate obtained using the training set, prove that

$$\mathbb{E}[R_{\mathrm{tr}}(\hat{\beta})] \leq \mathbb{E}[R_{\mathrm{te}}(\hat{\beta})].$$

11. Prove Proposition 4.1 of *Supplementary reading 1.*

12 (Primal Witness Dual)**.** Prove Proposition 4.2 of *Supplementary reading 1.*

13. Derive (5.1) and (5.2) of *Supplementary reading 1.* Specifically

(a) Find the equiangular direction, and compare your result with (5.1).

(b) Verify that the distance that LARS proceeds in the current step is given by (5.2).

14 (Elastic-Net)**.** Consider the *elastic-net* optimization problem:

$$\min_{\beta}\left\{\|\boldsymbol{y} - \boldsymbol{X}\beta\|^2 + \lambda[\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1]\right\}.$$

Show how one can turn this into a Lasso problem. [Hint: use an augmented version of $\boldsymbol{X}$ and $\boldsymbol{y}$.]