FSRM 588: Financial Data Mining Homework 1: Chapters 1-3

Fall 2015

Due 6:40pm on September 28

NOTES: Submit all your code with your assignment. Homework must be neatly written-up or typed for submission. Please read and follow the instructions carefully. I reserve the right to refuse homework that is deemed (by me) to be excessively messy.

Instructions:

- HW1 is intended to complement the materials in Chapters 1-3 and help you familiarize with all of the algorithms introduced in class.
- To this end, HW1 contains two parts: Part I is theoretical and Part II is computer exercises. Your answers to Part II MUST be typed up and printed out but your answers to Part I can be either typed up or written up. Submit the answers to both Part I and Part II in class.
- Submit the R code you use to the email address fsrm588@gmail.com as an attachment before class. The subject of the email should be "hw1 + your last name + your first name". For instance, "hw1 Yang Dan". Only .r files are allowed and the title of the attached document should be "hw1 + your last name + your first name.r". For instance, "hw1YangDan.r"
- The content of the email can be empty. There is no need to say thank me and I won't reply to the emails sent to fsrm588@gmail.com. fsrm588@gmail.com is only used for the purpose of submission of homework, project, and exam. Please reach me via dyang@stat.rutgers.edu if you have questions or need help.

PART I: Theory

1. *Statistical decision theory.*

   (a) Show that the solution to (2.12) is (2.13) on Page 18 of the textbook.

   (b) Show that plugging (2.15) into (2.9) and minimizing with respect to β will lead to (2.16) on Page 19 of the textbook.

2. *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 $\tilde{\sigma}^2$.

3. *Backward stepwise regression.* Suppose we have the multiple regression fit of $\mathbf{y}$ on $\mathbf{X}$. 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.

4. *Ridge regression with centering.* Consider the ridge regression problem

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

Show that this problem is equivalent to the problem

$$\hat{\beta}^{c} = \arg\min_{\beta^c} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0^c - \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)\beta_j^c \right)^2 + \lambda \sum_{j=1}^{p} (\beta_j^c)^2 \right\},$$

where $\bar{x}_j = N^{-1}\sum_{i=1}^{N} x_{ij}$. More specifically,

(a) give the correspondence between $\hat{\beta}^c$ and the $\hat{\beta}^{\text{ridge}}$;

(b) show that the two predicted output vectors are the same.

Show that a similar result holds for the Lasso.

5. *Kernel ridge regression (linear kernel).* Recall from class that the ridge regression estimator may be expressed as

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

and the corresponding prediction rule is

$$\hat{Y} = f(X) = X^T\hat{\beta}^{\text{ridge}} = X^T(\mathbf{X}^T\mathbf{X} + \lambda I_p)^{-1}\mathbf{X}^T\mathbf{y}.$$

Prove that

$$\hat{\beta}^{\text{ridge}} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda I_n)^{-1}\mathbf{y}.$$

6. *Coordinate descent.* Suppose we have a cost function $J(\theta)$ that depends on the parameter $\theta = (\theta_1, \ldots, \theta_p)^T$, 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

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

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 $(k \neq j)$ 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,

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

Here by an abuse of notation, the function J has only one argument β_j because we view other $\tilde{\beta}_k(\lambda)$ $(k \neq j)$ 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" and "input," as well as the penalty parameter.

(b) Assume that $\sum_{i=1}^{N} x_{ij} = 0$ and $\sum_{i=1}^{N} x_{ij}^2 = 1$. Find the updated value of β_j so that the function $J(\beta_j)$ is minimized.

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.

(c) Now consider the *elastic net* estimator, which minimizes

$$J_0(\beta) = ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda \left\{\alpha||\beta||_2^2 + (1-\alpha)||\beta||_1\right\}.$$

The elastic net has two tuning parameters, $\lambda > 0$ and $0 < \alpha < 1$. Repeat part (b) of this problem with the elastic net cost function $J_0(\beta_j)$ in place of $J(\beta_j)$.

7. *Ridge regression property.* Consider the ridge regression problem

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} x_{ij}\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\},$$

where the response and predictors are centered (note that we do not assume the predictors are normalized for now). Assume the following model

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_j,$$

where $\epsilon_j \overset{iid}{\sim} N(0, \sigma^2)$. Show the detailed derivations to the following questions, not just answers.

(a) Derive the solution $\hat{\beta}^{ridge}$ as an explicit function of $\mathbf{y}, \mathbf{X}, \lambda$.

(b) Compute the expectation $\mathbb{E}\hat{\beta}^{ridge}$.

(c) Compute the variance $\text{var}(\hat{\beta}^{ridge})$.

(d) What is the distribution of $\hat{\beta}^{ridge}$?

(e) Suppose the predictors are orthonormal (normalized and orthogonal to each other), what would $\hat{\beta}^{ridge}$ be as an explicit function of $\mathbf{y}, \mathbf{X}, \lambda$? Note that (e) is different from (f) where the solution involves $\hat{\beta}_j^{OLS}$.

(f) Suppose the predictors are orthonormal, show

$$\hat{\beta}_j^{ridge} = \hat{\beta}_j^{OLS}/(1+\lambda),$$

where $\hat{\beta}_j^{OLS}$ is the OLS solution.

PART II: Computer

8 *Simulations with ridge regression.* Let $\mathbf{X} = (x_{ij})_{1\le i\le N;\ 1\le j\le p}$, $\beta = (\beta_1, ..., \beta_p)^T$, $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_N)^T$, and

$$\mathbf{y} = (y_1, ..., y_N)^T = \mathbf{X}\beta + \boldsymbol{\epsilon}.$$

Simulate independent 100 datasets, each with $N = 500$, $p = 10000$, $\beta = (1, 1, ..., 1)^T \in \mathbb{R}^p$, and independent $x_{ij}, \epsilon_i \sim N(0, 1)$ $(1 \le i \le N;\ 1 \le j \le p)$. For each dataset, compute ridge regression estimators $\hat{\beta}^{\text{ridge}} = \hat{\beta}_\lambda^{\text{ridge}}$, where $\lambda = 0.1, 0.2, 0.3, ..., 10$ (so that you compute 100 ridge estimators for each dataset) and compute the mean-squared error (MSE)

$$\text{MSE}(\hat{\beta}_\lambda^{\text{ridge}}) = \frac{1}{p}||\hat{\beta}_\lambda^{\text{ridge}} - \beta||_2^2$$

for each value of λ in each dataset. Compute the average MSE for each value of λ over the 100 datasets, and plot the average MSE as a function of λ. Which value of λ corresponds to the smallest average MSE? Report the total run-time for your simulations (the run-time can be recorded using the `proc.time()` function in R).

9 *Simulations with subset selection.* Reproduce Figure 3.6 on Page 59 of the textbook according to the caption; only for best subset selection, forward stepwise and backward stepwise.

10 *Zipcode data with all methods in Chapters 1-3.* Consider the zipcode data, only 2's and 3's. The zipcode data are available from Sakai. Use the following code to load the data: load("hw1.RData"). The names of the variables are self-explanatory. Do the following exercises.

(a) Ex. 2.8 on Page 40. Compare the classification performance of linear regression and k - nearest neighbor classification on the zipcode data. In particular, consider only the 2's and 3's, and k = 1, 3, 5, 7 and 15. Show both the training and test error for each choice, plot the error curves

(b) Can you obtain the best subset selection result for all k if you do not manipulate the arguments of the function "regsubsets"? What is causing the problem?

(c) Run best subset selection algorithm with additional argument "really.big=TRUE". Can you obtain the result?

(d) Run best subset selection algorithm with additional argument "nvmax=3, really.big=TRUE". Figure out what nvmax does and which variables are retained by the best model with only 3 variables.

(e) Can you run forward selection without additional argument? Can you find out which variables are retained when we consider models with 9 predictors?

(f) Run forward selection with "nvmax=256". Now, can you find out which variables are retained when we consider models with 9 predictors? which variables are retained when we consider models with 3 predictors?

(g) Run backward selection with "nvmax=256". Which variables are retained when we consider models with 3 predictors? Compare your results from parts (d), (f), and (g)?

(h) For the forward selection, show the pictures of RSS, adjusted RSq, Cp, and BIC against the number of variables. Color the points that should be chosen according to 3 criteria, adjusted RSq, Cp, and BIC. What are the numbers of variables that should be chosen according to these criteria respectively?

(i) Repeat part (h) for backward selection.

(j) If we use the best model according to Cp from the forward selection, what are the training and testing errors? How about the best model according to BIC from the forward selection?

(k) Repeat part (j) for backward selection. Among the four models in parts (j)(k), which has the smallest training error? which has the smallest test error? which one is the best?

(l) Run ridge regressions with tuning parameter grid

```
lambda.grid <- 10^seq(4,-3,length=100)
```

Do not standardize the variables. Why not? Plot the coefficient path for ridge regression. Do not use the built-in command `plot` on the output from `glmnet` directly. Extract the coefficient matrix first and plot the rows of the matrix, except for the one for the intercept. You will notice that the coefficient for one variable tends to be large which diminishes the signal from others. Replot the path with that variable removed.

(m) What are the testing and training errors if OLS is used? Obtain the result by setting $\lambda = 0$ in ridge regression instead of using the function `lm`.

(n) Obtain the coefficient estimates from OLS by using both ridge with $\lambda = 0$ and the function `lm`. Plot the coefficient estimates from ridge with $\lambda = 0$ against the coefficient estimates from OLS. Check if they are the same.

(o) Compute the training and test errors for ridge regression for all choices of λ on the grid. Plot these errors against the logarithm of λ. What is the lambda value that gives the smallest test error? What is the lambda value that gives the smallest train error? What lambda should we use for classification on a future observation? If that lambda is used, what are the test and training errors for the current dataset and what is the chance for misclassification on a future observation?

(p) Run lasso with

```
lambda.grid <- 10^seq(0,-5,length=n.lambda)
```

Repeat (l) with ridge replaced by lasso

(q) Repeat (m) with ridge replaced by lasso

(r) Repeat (n) with ridge replaced by lasso

(s) Repeat (o) with ridge replaced by lasso

(t) Summarize the training and test errors for the best model chosen by forward selection, backward selection, ridge, and lasso, as well as OLS and NN. Which method works the best?