ORF350 (Spring 2017) Analysis of Big Data **Assignment: 3**

Conceptual Challenges, Analytics Edge and High Dimensional Regression

Lecturer: Han Liu *Email: hanliu@princeton.edu*

Note that the due date for this assignment is Monday March 27, 5:00pm. Please submit your hard copy at the ORF 350 dropbox in the Sherrerd Hall student lounge. Electronic submissions will not be accepted unless there is an extreme and compelling case.

1 Conceptual Challenges (15 points)

Select the answers as instructed (Note: each question **may have one or more** correct answers). Each question counts **3 points**. For each question, you **get zero point if any wrong answer is chosen**. If you miss one or more correct answers but all the chosen ones are correct, then you get **1 points**. If all the correct answers are chosen, you get full points.

(1) Let $\{X_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p_\theta(x)$ be n random samples (Let X be their population variable). We denote their realizations (or outcomes) to be $\{x_i\}_{i=1}^n$. Select all the WRONG statements.

A. The realizations $\{x_i\}_{i=1}^n$ are deterministic quantities.

B. Though a random sample X_i can fluctuate, its variance must be deterministic and finite.

C. Without the values of realizations, we cannot tell whether a statistic is consistent or not.

D. Without the values of realizations, we cannot give an estimate for the parameter of interest (e.g., θ).

E. The law of large numbers can be applied to both random samples $\{X_i\}_{i=1}^n$ and their realizations $\{x_i\}_{i=1}^n$.

F. The population variable X and the first random sample X_1 are identically distributed.

(2) Unbiasedness and Consistency. Select all the wrong statement:

A. Unbiasedness implies consistency.
B. Consistency implies unbiasedness.
C. Biased estimators can never be consistent.
D. Inconsistent estimators can be unbiased.
E. Let θ be the parameter of interest. If an estimator $\widehat{\theta}_n$ satisfies

$$\sqrt{n}\Big(\frac{\widehat{\theta}_n - \theta}{\sigma}\Big) \xrightarrow{D} N(0,1).$$

Then $\widehat{\theta}_n$ must be consistent.

(3) Law of Large Numbers (LLN) and Central Limit Theorem (CLT). Select all the WRONG statements:

A. Suppose $\{X_i\}_{i=1}^n$ are i.i.d. random samples and $\mathbb{E}X = \mu$. If $\sqrt{n}(\overline{X}_n - \mu) \xrightarrow{D} N(0, \sigma^2)$, then $\overline{X}_n \xrightarrow{P} \mu$.
B. If $\sqrt{n}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0,1)$, then $\widehat{\theta}_n - \theta \xrightarrow{D} N(0, n^{-1})$.
C. If $\sqrt{n}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0,1)$, then $\mathbb{P}\left(\theta \in [\widehat{\theta}_n - \frac{z_{\alpha/2}}{\sqrt{n}}, \widehat{\theta}_n + \frac{z_{\alpha/2}}{\sqrt{n}}]\right) \geq 1 - \alpha$ for sufficiently large n, where z_α is the α upper quantile of the standard normal distribution.

(4) Linear Regression and Ordinary Least Squares (OLS). Select all the WRONG statements:

A. In the regression model $Y = \boldsymbol{X}^T\boldsymbol{\beta} + \epsilon$, Y and $\boldsymbol{X}$ are deterministic quantities and ϵ_i is a random noise.
B. If the random samples $\{(Y_i, \boldsymbol{X}_i)\}$ independent follow the linear regression model $Y = \boldsymbol{X}^T\boldsymbol{\beta} + \epsilon$ where ϵ is independent of $\boldsymbol{X}$ and $\epsilon \sim N(0, \sigma^2)$, the OLS estimator of the coefficient vector $\widehat{\boldsymbol{\beta}}$ is unbiased.
C. The OLS estimator of the coefficient vector $\widehat{\boldsymbol{\beta}}$ is always unique.

(5) Assuming the true distribution of the data follows a linear model $Y = \beta_0 + \beta_1 X + \epsilon$. We fit an ordinary least squares regression using this true model on the data, as the number of data points goes to infinity, your estimator will have

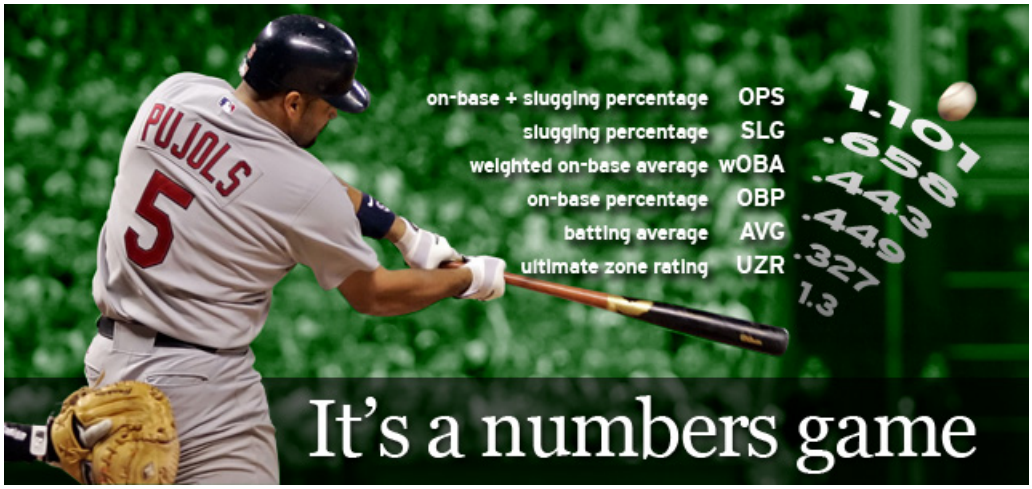
A. variance approaching zero.
B. lower variance but not approaching zero.
C. same variance.
D. lower bias approaching zero.
E. lower bias but not approaching zero.
F. same bias.

2 Short Question (10 points)

Prove or Give Counter Examples.

- If $X_n \xrightarrow{P} X$ and $Y_n \xrightarrow{P} Y$, then $X_n + Y_n \xrightarrow{P} X + Y$.
- If $X_n \xrightarrow{D} X$ and $Y_n \xrightarrow{D} Y$, then $X_n + Y_n \xrightarrow{D} X + Y$.
- If $X_n \xrightarrow{D} C$ for some constant C, then $X_n \xrightarrow{P} C$.

3 Moneyball: The Analytics Edge in Sports (15 points)



Baseball is one of the most symbolic sports in the US. Back into the 20th century, baseball teams relied on respected and experienced scouts to recruit players. However, as statistical analytics were introduced to this game, the rule changed. Michael Lewis's bestselling book *Moneyball: The Art of Winning an Unfair Game* tells the story of how Billy Beane, the general manager of the Oakland Athletics in 1998, effectively used statistical analytics to turn this losing team into a winning team with very stringent payroll budget. His great success was due to the discovery of significant features that create runs through statistical analysis. In particular, Billy found that on-base percentage (`OBP`) and slugging percentage (`SLG`) have much better performance in measuring offensive capability than on batting average (`BA`), which has always been baseball's most famous and well-published statistic. In this problem, we would use the real data to verify Billy's claim, and see how regression analysis can reshape decision making procedures in baseball team management and lead to dramatic enhancement in team performance.

First of all, let's familiarize ourselves with some key terminologies.

1. Plate appearances: It is counted every time a player comes to bat regardless of the outcome of that time at the plate
2. At bat: It is counted only the times a player gets a hit or make an out.
3. Batting average (`BA`): It represents the percentage of at bats that result in hits for a particular baseball player.
4. On-base percentage (`OBP`). It is a measure of the number of times a player gets on base by hit, walk, or hit by pitch, expressed as a percentage of his total number of plate appearances.
5. Slugging percentage (`SLG`): It is calculated as total bases divided by at bats.

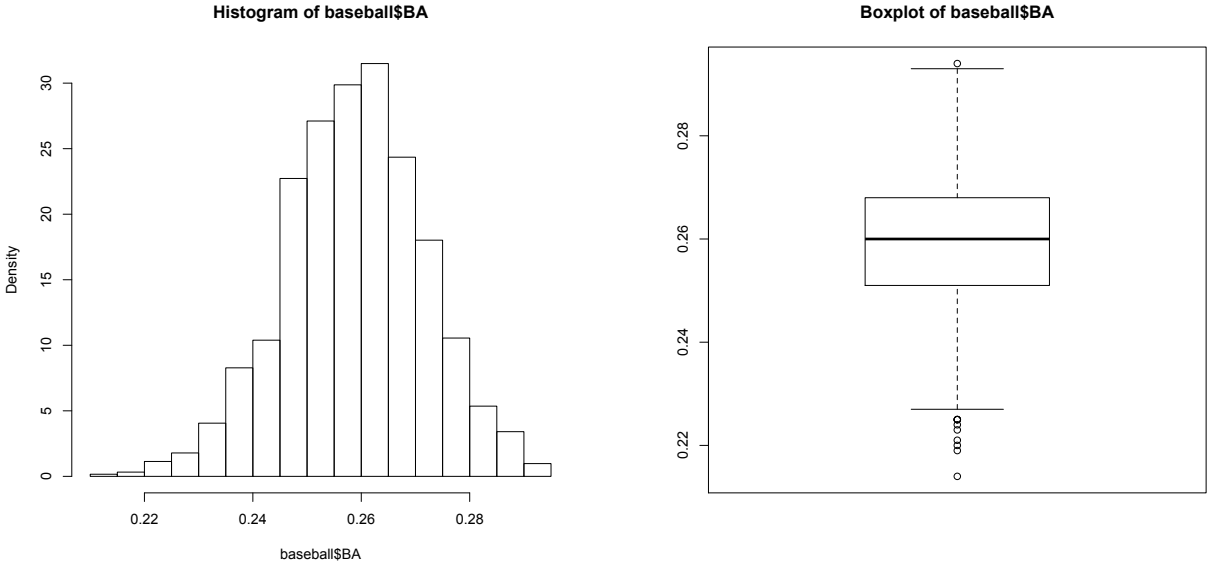
In this problem, we would use the dataset *moneyball.rda* that has statistics of performance of all Major League Baseball (MLB) teams from 1962 to 2012. We list below meanings of important features in the dataset. You are encouraged to google the detailed information if you are not pretty sure about the terminologies here.

1. `RS`: runs scored
2. `RA`: runs allowed

3. `W`: the number of winning games
4. `OBP`: on-base percentage
5. `SLG`: slugging percentage
6. `BA`: batting average
7. `Playoffs`: whether making a playoff, 1 means yes and 0 means no
8. `OOBP`: opponent on-base percentage
9. `OSLG`: opponent slugging percentage

Read in the moneyball.rda file using the load() function.

3.1 Preliminary Analysis. Plot histograms and boxplots for `OBP`, `SLG`, `BA`. Also report the mean and median for these three statistics. The purpose of doing this is to get an initial idea of how the data are distributed and detect potential outliers. Below we present the histogram and boxplot for `BA` for your reference.



The mean and median of `BA` are 0.259 and 0.26, meaning that the distribution is not skewed at all. You can also verify this from the boxplot and histogram. Analyze the other two quantities similarly.

3.2 The goal of this question is to compare regression models. Please fit the following two models (including an intercept term). Use only the samples from 1997 or earlier.

$$\text{Model 1: Predict } R \text{ using } BA \tag{3.1}$$
$$\text{Model 2: Predict } R \text{ using } OBP, SLG \tag{3.2}$$

Report the R-squared value for each regression fit. Which fit would you prefer based on the R-squared value?

3.3 The goal of this question is to combine different models and do prediction. After reviewing the case study videos for this dataset, let us complete the model to predict the number of wins. We will fit the following 3 regression models.

$$\mathrm{R} = \beta_1 \mathrm{OBP} + \beta_2 \mathrm{SLG} + \beta_3 + \varepsilon \tag{3.3}$$
$$\mathrm{RA} = \beta_4 \mathrm{OOBP} + \beta_5 \mathrm{OSLG} + \beta_6 + \varepsilon \tag{3.4}$$
$$\mathrm{W} = \beta_7 \mathrm{R} + \beta_8 \mathrm{RA} + \beta_9 + \varepsilon \tag{3.5}$$

We will train this model on the samples only from 1999 to 2011 (inclusive). After fitting all 3 models to estimate $(\widehat{\beta}_1, \ldots, \widehat{\beta}_9)$, we will use the following prediction strategy:

$$\widehat{W} = \widehat{\beta}_7(\widehat{\beta}_1 \mathrm{OBP} + \widehat{\beta}_2 \mathrm{SLG} + \widehat{\beta}_3) + \widehat{\beta}_8(\widehat{\beta}_4 \mathrm{OOBP} + \widehat{\beta}_5 \mathrm{OSLG} + \widehat{\beta}_6) + \widehat{\beta}_9, \tag{3.6}$$

where we used a plug-in estimator for R and RA. Predict the number of wins for all the samples in 2012 using the above equation. What is the (Pearson) correlation between the number of predicted wins and the number of observed wins for all the teams in 2012?

4 Big Data. Big Profit. (35 points)

Behind every successful website, there are the advertisements that paid for its growth. Online advertising represent a major source of traffic as well as income for modern businesses. The product may be exceptional, but it is the advertisements that bring the users and (especially for free services) pay the bills.

At the center of this industry, the advertisement exchanges that auction out advertisement spaces as a website loads and demand side platforms (DSPs) that compete to these spaces up.

Enter you. A young, ambitious data scientist at one of these DSPs: iPinYou. You are ready to show them what you got. Your first assignment? Design a Real-Time Bidding (RTB) algorithm that, for any given ad space, predicts how many clicks it may get and decides how much to bid for it.

You will be using the `glmnet` package with the iPinYou dataset[1], which provides you with information about previous auctions including properties of the space (height, width, etc.), information about the website viewer, and the bids by iPinYou and competing exchanges.

Install the `glmnet` package. Load the bidding data `iPinYou.RData` into R using `load()`. You can use `ls()` to see the objects that are loaded. Let's get started! You have a career to build!

[1]This dataset is a slightly simplified version of the one used in the Real-Time Bidding (RTB) algorithm competition held in 2013.

4.1 Click Me! Click Me! (20 points)

Our first task is to predict whether a user will click on a certain add. We consider only the `Region`, `City`, `AdX`, `Domain`, `Key_Page`, `Ad_Vis`, `Ad_Form`, `Ad_Width`, `Ad_Height` and `Floor_Price` columns as our features, and we're interested in predicting `Click`. We treat the first 7 relevant columns (`Region` to `Ad_Form`) as categorical features, and we will use multiple indicator columns to express each of these features. Please see the following hint for the specific instructions.

(**Hint:** Indicator columns are columns whose entries can only be ones or zeros. We usually use multiple indicator columns to represent a single categorical feature. If the feature has k different categories, you need $(k-1)$ indicator columns to express it. To illustrate that, consider the feature `Region` that has three categories: Region 1, Region 3, and Region 6. You would use two indicators to represent these three regions: Region 3 would correspond to "10" and Region 6 to "01". The entries "00" would then implicitly correspond to Region 1. The reason why we use indicator columns for categorical data is that unlike numeric features, numbers in categorical features do not represent magnitude but only difference. For example, if we stick to using the original column for the categorical feature `Region`, it does not make any sense that Region 6 is "bigger" than Region 1. In your code, you may choose the implicit category for each categorical variable, but please indicate your choice in the write-up.)

The last three columns (`Ad_Width`, `Ad_Height` and `Floor_Price`) are numeric data (as opposed to categorical data). Standardize all three columns in the following way: for each column, subtract the mean value of that column and then divide by the standard deviation of that column. This will set the transformed column to have a mean of 0 and standard deviation of 1.

You should have 21 training columns when you're done (2 columns for `Region`, 5 columns for `City`, etc. This does not include the `Click` column.) Lastly, convert the `Click` column to have value 1 if there is at least one click, 0 if there are no clicks at all. After this preprocessing step, do the following steps.

- **Part a:** Fit the model on the training data using the `glmnet` function to predict if there is a click or not. You should predict 0 if you think there is not going to be a click, and 1 if you think there will be. Plot the regularization paths of Lasso and Ridge under the Binomial family. An example of the Lasso's regularization path is shown in Figure 1. See Section A for Under-the-Hood details on the objective function that the package `glmnet` aims to optimize under the Binomial family.
  (**Note:** In `glmnet` and, later, in `cv.glmnet`, please use `standardize=FALSE` since we already standardized manually.)

- **Part b:** Based on the 2 graphs you just plotted, which features seem to be more important than others?
  (**Note:** You may choose your own criteria for how to identify important features. However, you must explain these choices clearly in your write-up.)

- **Part c:** For both Lasso and Ridge regression, use a 5-fold cross validation (via the `cv.glmnet` function) to determine the tuning parameter λ. Mark the corresponding L1-norm on your plot made in Part a[2]. Also plot the cross validation. Your graphs will roughly look like the form in Figure 1 above. Using a few sentences, interpret these four graph (from Part a and Part c) and give some intuition on why models with larger degrees of freedom do not necessarily do better in the cross validation.

- **Part d:** Evaluate both models by using the testing data. Record each classification test error by reporting the prediction accuracy for data points where (in truth) $y_i = 1$ and for data points where $y_i = 0$. (For each of the two models, you will be reporting two numbers.)

[2]This is a bit statistically meaningless for Ridge regression since Ridge regression constrains the L2-norm, not the L1-norm. Unfortunately, there's no easy way to set the x-axis to be the L2-norm.

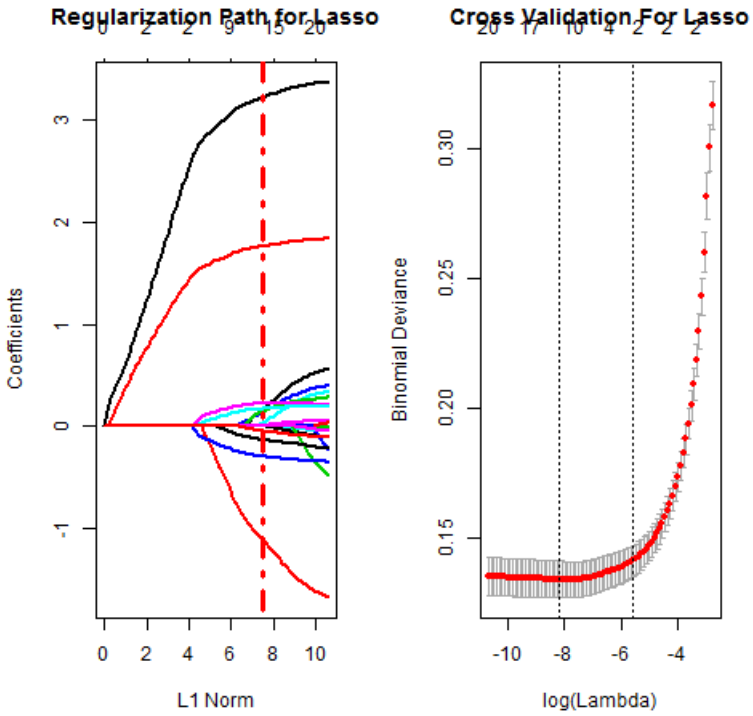


Figure 1: An (truncated) example of what your coefficients (left) and cross-validation (right) might look like. The red dotted line in the coefficient graph marks the L1-norm chosen by cross validation, and left-most lambda in the cross-validation graph should correspond to this L1-norm. Your solution to this problem will have 4 plots similar to the left plot, 2 plots similar to the right plot.

You will have a total of 4 plots for this question (2 for normal `glmnet`, 2 for using cross-validation). [This question helps you understand how `glmnet` works.]

4.2 Know The Enemy (15 points)

Unfortunately, there is another DSP bidding for the same spaces as you. To give yourself an edge, you decide to predict how much the other exchange will bid. At the same time, you get to review exactly how the Lasso and Ridge regression "regularizes" our statistical model. Fun!

We consider only the `AdX` (the exchange of original for the ad space) and `iPinYou_Bid` columns (what iPinYou had bid on the space) as our features, and we're interested in predicting `Comp_Bid`[3] (the bid by the competitor). Standardize all three columns in the following way: for each column, subtract the mean value of that column and then divide by the standard deviation of that column. This will set the transformed column to have a mean of 0 and standard deviation of 1. After the transformation, our linear relationship is

$$\text{Comp\_Bid} = \beta_1 \text{AdX} + \beta_2 \text{iPinYou\_Bid} + \epsilon$$

for a Gaussian noise term ϵ. Here, β_1, β_2 are two coefficients. Notice that since we standardize our variables, we no

[3]This linear regression does not make complete since `AdX` is a categorical variable, but as we will see, it carries some predictive power if we "pretend" it's a numerical variable.

longer have an intercept. Follow these steps:

- Determine the MLE coefficients of the linear regression. That is, use the `lm` function to compute the linear regression coefficients.
- Determine the Lasso coefficients of the linear regression. That is, use the `glmnet` function to compute the linear regression coefficients under a Gaussian model. Pick the Lasso coefficients that have half the L1-norm of the MLE coefficients. See Section A for Under-the-Hood details.
- For both a discretized grid of values for $\beta_1, \beta_2 \in [-.5, 1]$, compute the mean square error (MSE) according to each pair of β_1 and β_2. Generate the discretized grid by

  ```
  beta1 = seq(-.5,1,length.out=100)
  beta2 = seq(-.5,1,length.out=100)
  ```

  and recall that for response y_i and data $\mathbf{x}_i = \{x_{i,1}, x_{i,2}\}$ over n data points, the MSE is calculated with

  $$\text{MSE}(\beta_1, \beta_2) = \sum_{i=1}^{n} (y_i - (\beta_1 x_{i,1} + \beta_2 x_{i,2}))^2. \tag{4.1}$$

  (**Note:** This step takes a large amount of memory. If this causes problems like freezing or crashing, try closing R and restarting.)
- Using the `contour` (built-in) function in `R`, plot level curves of the MSE, and place the MLE coefficients and Lasso coefficients on this plot.
- Draw the L1-ball such that the Lasso coefficients sit on the edge of the L1-ball (diamond-shaped).

Do the same steps but for the Ridge coefficients (and picking the coefficients that have half the L2-norm of the MLE coefficients), and instead of an L1-ball, draw an L2-ball (circle-shaped). Your plots will look like Figure 2.

Answer the following questions: In relationship to the level curves, where do to the MLE coefficients sit? What does this say about the MSE of the MLE coefficients? In relationship to the level curves and the L1-ball (or L2-ball), where do the Lasso (or Ridge) coefficients sit? What does this say about the MSE of the Lasso and Ridge coefficients? Lastly, based on our geometric representation of the L1- and L2-balls, why can we believe that Lasso coefficients favor sparsity over Ridge coefficients? Answer each of the 5 questions with 1-2 sentences. [This question shows you how regularization works geometrically.]

5 A Regression by Any Other Form Will Predict Just As Sweetly. (15 points)

5.1 Constrained and Unconstrained Optimization

Suppose we have $\mathbf{Y} \in \mathbb{R}_{n\times 1}$ and $\mathbf{X} \in \mathbb{R}_{n\times p}$. In this question, we explore the relationship between the following two optimization problems

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \quad \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \tag{5.1}$$

and

$$\begin{aligned} \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \quad & \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\beta}\|_1 \leq C \end{aligned} \tag{5.2}$$

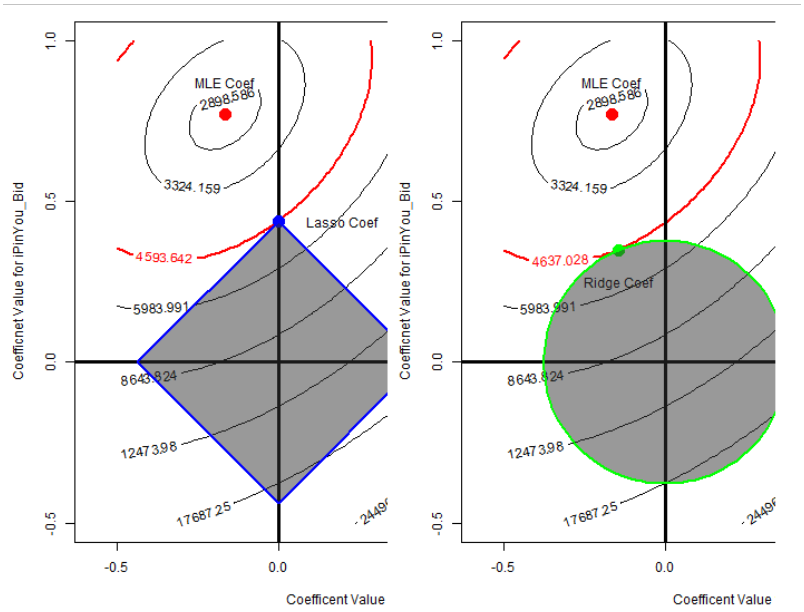


Figure 2: An (truncated) example of what your geometric pictures (one for Lasso, one for Ridge) for the MSE might look like. Your solution to this problem with have two plots, each similar to one shown here.

Show that for any positive λ in (5.1), there exists a constant C such that (5.2) shares the same minimizer with (5.1). Give comments on this conclusion considering the L1-ball and likelihood level curves you draw for Lasso in Question 1.

(Hint 1: Suppose $\widehat{\boldsymbol{\beta}}$ is a minimizer of (5.1). Choose the constant C to be $C = \|\widehat{\boldsymbol{\beta}}\|_1$. Use a contradiction argument to show that $\widehat{\boldsymbol{\beta}}$ must be the minimizer of (5.2) as well.
Hint 2: What must be true if $\widehat{\boldsymbol{\beta}}$ were not a minimizer of (5.2)? How would this lead to a contradiction in your assumptions?)

5.2 Regularized Regression

Consider the elastic-net optimization problem for pre-given $\mathbf{y}$, $\mathbf{X}$, α and λ:

$$\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda[\alpha\|\boldsymbol{\beta}\|_2^2 + (1-\alpha)\|\boldsymbol{\beta}\|_1] \Big\}. \tag{5.3}$$

Show how one can turn this into a standard lasso problem using an augmented version of $\mathbf{X}$ and $\mathbf{y}$. That is, add columns and/or rows to $\mathbf{X}$ and $\mathbf{y}$ with specific values to become $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{y}}$ so the optimal value of (5.3) matches that of the lasso optimization problem:

$$\min_{\boldsymbol{\beta}} \Big\{ \|\widetilde{\mathbf{y}} - \widetilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2 + \widetilde{\lambda}\|\boldsymbol{\beta}\|_1 \Big\}. \tag{5.4}$$

[This question shows you how to reformulate elastic-net regularization as a special case of lasso regularization.]

6 Modeling The Decisions of Justice Stevens (10 points)

The Supreme Court of the United States (SCOTUS) makes some of the most controversial decisions in the United States. The SCOTUS has the same set of justices during 1994 and 2001. We will model Justice John Paul Stevens's decision (whether or not Justice Sevens voted to reverse the lower court decision) during that time period using tree models. Justice Stevens is a self-proclaimed conservative but he's usually considered liberal during his time on the Supreme Court. Let's use classification and regression tree and random forest to model the decisions based on six explanatory factors: Circuit, Issue, Petitioner, Respondent, LowerCourt, Unconst. The variable Circuit refers to circuit or lower court where the case came from. The issue area is a categorical description of the case, like economic activity or criminal procedure. Petitioner and Respondent are two parties in the case. The LowerCourt variable indicates whether the lower court's decision is a liberal or conservative decision. The Unconst variable means whether or not the petitioner argues that a law or practice is unconstitutional. The dependent variable Reverse is 1 if Justice Stevens decided to reverse the lower court's decision and 0 vice versa.

Use the script `scotus.R` to load the dataset, split the dataset into training and testing data.

Figure 3: The Rehnquist Court in 2005, from Wikimedia Commons

6.1 We aim to train a classification and regression tree (CART) model in the script `scotus.R` with each leaf node having at least 25 data points. Fill in the missing parts of the code. Use the `prp` function to print the trained model. Calculate and report the classification error on testing data.

ntree = 10 not 200!

6.2 We also fitted a random forest model in the script. Let's fix the parameter `ntree` as 200 and find the best `nodesize` parameter using 10-fold cross validation on training set with the random seed set as 350. Read the code and fill in the missing parts of the code. Report the cross validation error in terms of the misclassification using the best `nodesize` parameter.

Acknowledgement

We teaching staff greatly appreciate your insightful questions on PIAZZA, which give lots of inspiration in creating the multiple choice problems. We also appreciate the MIT online course *Analytics in Edge*, which provides interesting stories on power of statistical analytics in real applications and the relevant datasets.

A Under-the-Hood Details

Generally speaking, the `glmnet` package minimizes the sum of negative log-likelihood and l1-penalization of β. Under different model families, the objective function has different specific forms.

Additional Details for Question 4.1. Here, we want to form predictions that follow a Binomial distribution. We do not need to worry about this ourselves, but just so we're aware of what's going on, `glmnet` uses a regularized logistic regression to handle a binomial response. The statistical model utilizes a conditional probability of the following form

$$\mathbb{P}_{\beta_0,\boldsymbol{\beta}}(Y_i = y_i \mid \mathbf{X}_i = \mathbf{x}_i) = \left(\frac{1}{1+e^{-\beta_0-\mathbf{x}_i^\intercal\boldsymbol{\beta}}}\right)^{y_i}\left(1-\frac{1}{1+e^{-\beta_0-\mathbf{x}_i^\intercal\boldsymbol{\beta}}}\right)^{(1-y_i)} \tag{A.1}$$

where β_0 is our intercept, $\boldsymbol{\beta}$ is our vector of p coefficients, n is the number of data points, y_i is the Bernoulli variable indicating a click or not for data point i, $\mathbf{x}_i$ is a vector of 21 features for data point i. Notice that this looks like the form $p_i^{y_i}(1-p_i)^{(1-y_i)}$ for some p_i. The objective function `glmnet` is solving is

$$\min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}} -\frac{1}{n}\sum_{i=1}^{n}\left[y_i(\beta_0+\mathbf{x}_i^\intercal\boldsymbol{\beta}) - \log(1+e^{(\beta_0+\mathbf{x}_i^\intercal\boldsymbol{\beta})})\right] + \lambda[(1-\alpha)||\boldsymbol{\beta}||_2^2 + \alpha||\boldsymbol{\beta}||_1] \tag{A.2}$$

where α is a parameter.

Additional Details for Question 4.2. Here, we want to form predictions that follow a Gaussian distribution. In this setting, `glmnet` uses a regularized linear regression. The statistical model is

$$\mathbb{P}_{\beta_0,\boldsymbol{\beta},\sigma^2}(Y_i = y_i \mid \mathbf{X}_i = \mathbf{x}_i) = N(y_i - \beta_0 - \mathbf{x}_i^\intercal\boldsymbol{\beta}, \sigma^2) \tag{A.3}$$

where we're only interested in estimating β_0 and $\boldsymbol{\beta}$. The objective function `glmnet` is solving is

$$\min_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{p+1}} \frac{1}{2n}\sum_{i=1}^{n}(y_i-\beta_0-\mathbf{x}_i^\intercal\boldsymbol{\beta})^2 + \lambda[(1-\alpha)||\boldsymbol{\beta}||_2^2 + \alpha||\boldsymbol{\beta}||_1]. \tag{A.4}$$