STAD80 Analysis of Big Data **Assignment: 1**

Fundamentals of Statistics and Regression

Lecturer: Qiang Sun *Email: stad80.utsc@gmail.com*

You will have to submit '.rmd' R Markdown, '.tex' latex file (can be generated from R markdown) and '.pdf' file (can be generated from R Markdown). I should knit the '.rmd' file and compile '.tex' file without troubles. You can also submit 'Pythonnotebook', '.tex' + '.pdf' files. Rmarkdown does support python too!

Question 1 (Conceptual Challenges (30 points))**.** *Select the answers as instructed (Note: each question* **may have one or more** *correct answers). Each question counts* **5 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* **2 points***. If all the correct answers are chosen, you get full points.*

(1) Let $\{X_i\}_{i=1}^n \overset{i.i.d.}{\sim} p_\theta(x)$ *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.*

(6) Select all the WRONG statements:

A. *R^2 is only used when we are doing linear regression.*

B. *R^2 on training data can be negative if fitting is very bad.*

C. *R^2 on training data is always no smaller than that on testing data.*

D. *Recruiting more variables in the linear model never hurts R^2 on the training data for the OLS estimator.*

E. *The higher R^2 is, the better the fitted model is.*

Question 2 (Maximum Likelihood Estimator (MLE) and Asymptotic Normality (20 points)). *Maximum likelihood is one of the most fundamental principals in parameter estimation. Suppose we have n i.i.d. random samples $\{X_i\}_{i=1}^n$ that have probability density function $p_\theta(x)$. We are interested in estimating the parameter θ. Denote the correspondent MLE by $\widehat{\theta}_n$. In the lecture, we have known that under some regularity conditions, the MLE enjoys the asymptotic normality*

$$\sqrt{n}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0, \frac{1}{I(\theta)}), \tag{0.1}$$

where

$$I(\theta) := \mathbb{E}_\theta\Big(-\frac{\partial^2}{\partial\theta^2}\log p_\theta(X)\Big) = -\int_{\mathcal{X}}\Big(\frac{\partial^2}{\partial\theta^2}\log p_\theta(x)\Big)p_\theta(x)dx$$

is the Fisher information and $\mathcal{X}$ is the range of X_i.

Part I. Let z_α denote the α upper quantile of standard normal distribution. Prove that

$$C_n = \left[\widehat{\theta}_n - \frac{z_{\alpha/2}}{\sqrt{nI(\widehat{\theta}_n)}}, \widehat{\theta}_n + \frac{z_{\alpha/2}}{\sqrt{nI(\widehat{\theta}_n)}}\right]$$

is a $(1-\alpha)$ asymptotic confidence interval for θ, i.e., $\lim_{n\to\infty} P(\theta \in C_n) = 1-\alpha$. We assume $I(\theta)$ is a continuous function.
(Hint 1: According to Slutsky's Theorem, if $X_n \xrightarrow{D} X$ and $Y_n \xrightarrow{P} c$, where c is some constant, then $X_nY_n \xrightarrow{D} cX$. Use this result to prove that $\sqrt{nI(\widehat{\theta}_n)}(\widehat{\theta}_n - \theta) \xrightarrow{D} N(0,1)$.)

Part II. Suppose $\{X_i\}_{i=1}^n$ have the following probability density function:

$$p_\theta(x) = (\theta-1)x^{-\theta}\cdot 1\{x \geq 1\},$$

where $\theta > 1$ is the parameter of interest. Denote this distribution by P_θ and MLE of θ by $\widehat{\theta}_n$.

(a) Derive the MLE $\widehat{\theta}_n$.

(b) From (0.1), we know that the MLE $\widehat{\theta}_n$ is asymptotically normal. Calculate the asymptotic variance in terms of θ.

(c) Derive a 95% asymptotic confidence interval (CI) for the parameter θ.

(d) Use simulations to verify the effectiveness of the CI in (c) when $n = 100$ and $\theta = 2$.
(Hint 1: For the simulation part, we need to figure out how to generate random variables that follow P_θ. Suppose we have obtained the cumulative distribution function $F(x)$ of P_θ, and we generate a random variable U that is uniformly distributed over $[0,1]$. It is true that $F^{-1}(U) \sim P_\theta$.)
(Hint 2: By effectiveness of the CI, we mean whether the constructed CI will cover the true θ with probability around 95%. To verify the effectiveness of the CI, we can independently generate a large number of datasets (e.g., 10,000 datasets) with each having the sample size n. Calculate CI's for all generated datasets respectively and then summarize the frequency of CI's covering the true θ. If the frequency is around 95%, then we can claim that the constructed CI is effective.)

Question 3 (Law of Large Numbers and Central Limit Theorem (20 points)). *This question helps you to better understand the concepts of convergence in probability and convergence in distribution. In particular, you will visualize the Law of Large Numbers and Central Limit Theorem for a Discrete Distribution:*

$$\mathbb{P}(X = 1) = \mathbb{P}(X = -1) = 1/2.$$

Generate $N = 10,000$ datasets, each of which has n data points.
(Hint: Write a function in R that samples from the uniform distribution between 0 and 1 using the built-in `runif` *function. If the result is less than 0.5, set it to -1. Otherwise, set it to 1.)*
Let $\overline{X}_n^{(i)}$ be the average of i^{th} dataset, $\mu = EX$ and $\sigma^2 = \mathrm{Var}(X)$. We consider $n = \{10, 100, 1000, 10000\}$ for this simulation.
(Hint: You will not need the individual data points from each dataset. Therefore, to save memory, you need only store the $\overline{X}_n^{(i)}$ rather than all the data points. It is highly recommended that you do this to avoid freezing or crashing your computer.)

Plot and intepret the following:

(a) $\log_{10}(n)$ *v.s.* $\overline{X}_n^{(1)} - \mu$*;*
(Hint: This plot illustrates how the deviation $\overline{X}_n^{(1)} - \mu$ converges to 0 as n goes to infinity).

(b) Draw $\log_{10}(n)$ *v.s.* $\frac{1}{N}\sum_{i=1}^{N} 1\{|\overline{X}_n^{(i)} - \mu| > \epsilon\}$ *for* $\epsilon = 0.5, \epsilon = 0.1, \epsilon = 0.05$*;*
(Hint 1: This plot illustrates the law of large numbers. Please explain why.)
(Hint 2: For some statement S, the indicator function $1\{S\}$ is defined as $1\{S\} = 1$ if S is true and $1\{S\} = 0$ otherwise.)

(c) Draw histograms and Q-Q plots of $\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma$ for N datsets for $n = 10, n = 1,000, n = 10,000$. You may choose your histogram bins or you may let R choose automatically–any meaningful plot will do.
(Hint: This plot illustrates the Central Limit Theorem. Please explain why.)

(d) Generate i.i.d. standard normal $Y_1, \ldots, Y_N$ that are independent to previous random variables. Plot

$$\log_{10}(n) \text{ v.s. } \frac{1}{N}\sum_{i=1}^{N} 1\{|\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma - Y_i| > \epsilon\} \text{ for } \epsilon = 0.001.$$

(Hint: This plots illustrates the difference between convergence in probability and convergence in distribution. Explain why this plot shows that $\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma$ does not converge to Y in probability.)

Instructions: *This problem tries to familiarize you with* `R` *programming. You should produce 11 graphics in total – one plot for (a), three plots for (b), six plots for (c), and one plot for (d). You may format and combine the plots together for each part in whichever way you like, but we must be able to clearly read and interpret your results.*

Question 4 (Basic R Programming for Big Data (20 points))**.** *This problem will help you practice coding in* `R` *under a big data setting. The dataset comes from a dating website "LibimSeti." It consists of two files,* `ratings.dat` *and* `users.Rdata`*. The* `ratings.dat` *contains 3,000,000 ratings of profiles made by LibimSeTi users. It is organized as a comma-separated matrix with 3,000,000 rows and 3 columns. The* `users.Rdata` *contains the basic information of the users. It has $135,358$ rows and 7 columns.* `Readme.txt` *contains a detailed description of the data.*

(a) Load `ratings.dat` *into* `R` *using the package* `bigmemory`*, and name the columns by UserID, ProfileID, Rating. To evaluate every profile's score, we use the weighted rank (also used by IMDB):*

$$\textit{Weighted Rank (WR)} = (v \div (v+m)) \times R + (m \div (v+m)) \times C, \textit{ where}$$

$$\begin{array}{rcl} R & = & \textit{average rate for the profile} \\ v & = & \textit{number of votes for the profile} \\ m & = & \textit{4182 (the 250th largest number of ratings for a single profile)} \\ C & = & \textit{the mean rate across the whole data.} \end{array}$$

Write an `R` function `weighted.rank(ProfileID)` returning the weighted rank of the profile with its ID as input. Use the functions `which` and `apply` to compute the weighted ranks of all the profiles who were rated by UserID 100. Report your result by plotting a histogram of those scores you obtained. You can also use the `mwhich` and `lapply` functions. [This teaches you how to use bigmemory to load big data and how to write a R function.]

(b) *Load `users.Rdata` into `R`. Then you will have a matrix `User` in your working environment. Based on `users.Rdata` and `ratings.dat`, plot the boxplots of ratings from 1) male users coming from New York State and 2) female users coming from California. (Hint: you can use the `unique` or `grep` functions to find out the different ways each state is recorded in the dataset.)*

(c) *In order to predict a user's rating for a profile, fit a linear regression using the average rating given by the user and the average rating of the profile from all users as predictors. Report your results. Specifically, report the R-squared value and the three coefficients (intercept, coefficient for average rating given by a user and coefficient for average ratings given to a profile). Informally, the model is*

$$\textit{(User } i\textit{'s rating on Profile } j\textit{)} = \theta_1 + \theta_2 \textit{(Average Rating given by User } i\textit{)} + \theta_3 \textit{(Average Rating given to Profile } j\textit{)} + \epsilon_{ij}$$

where $\{\theta_1, \theta_2, \theta_3\} \in \mathbb{R}^3$ and ϵ_{ij} is a Gaussian noise term. You are required to 1) run the regression over the entire dataset using the the `R` function `biglm` in the package of `biganalytics`; 2) apply the subsampling technique, which means you do regression over multiple small subsamples respectively and take the mean of coefficients fitted by all subsamples as the aggregated estimand of coefficients. You can refer to Hint 3 below for further instructions on the subsampling technique. You are encouraged to compare the difference between the two methods in terms of the coefficient values, the R-squared value and the CPU time. You might need to recall how to calculate the R-squared value from previous courses.

Hint 1: Simple things become painful under the big data setting. The most time-consuming part is to calculate average ratings involved in the linear model. The most natural way of doing this is to use `mwhich` for each user (profile) to locate the ratings associated with the given user (profile) and then do the average, but once you run the code you will feel the pain: the progress is very slow. Here we suggest another way of doing this, which we hope can make your life easier. Try to understand the following code for calculation of average ratings, and if you like it, you can copy the following code from the given `R` script file `AveRating.r`. :)

```
N=3000000   # number of rating records
Nu=135359   # maximum of UserID
Np=220970   # maximum of ProfileID
user.rat=rep(0,Nu)  # user.rat[i] denotes the sum of ratings given by user i
user.num=rep(0,Nu)  # user.num[i] denotes the number of ratings given by user i
profile.rat=rep(0,Np)  # profile.rat[i] denotes the sum of ratings given to profile i
profile.num=rep(0,Np)  # user.rat[i] denotes the number of ratings given to profile i
for (i in 1:N){ # In each iteration, we update the four arrays, i.e. user.rat,
    user.num, profile.rat and profile.num, using one rating record.
  user.rat[X[i,'UserID']]=user.rat[X[i,'UserID']]+X[i,'Rating']   # The matrix X here
      comes from the file 'ratings.dat'
  user.num[X[i,'UserID']]=user.num[X[i,'UserID']]+1
  profile.rat[X[i,'ProfileID']]=profile.rat[X[i,'ProfileID']]+X[i,'Rating']
  profile.num[X[i,'ProfileID']]=profile.num[X[i,'ProfileID']]+1
  if (i %% 10000==0) print(i/10000)
}
user.ave=user.rat/user.num
profile.ave=profile.rat/profile.num
X1=X
colnames(X1)=c('UsrAveRat','PrfAveRat','Rat')
X1[,'UsrAveRat']=user.ave[X[,'UserID']]
```

```
X1[,'PrfAveRat']=profile.ave[X[,'ProfileID']]  # X1 is the new data matrix we will
    work with in regression.
```

With the code given above, we can finish calculating all the average ratings by only one for-loop over the entire dataset.

Hint 2: Using `biglm` *on the entire dataset can take around one hour on a descent laptop. Be patient. Here are several tips for running codes on big data: a) Test you code first on a small dataset; b) In order to know that your codes are still working, you should have your code display the current progress of your program (the percentage of work finished, the number of loops it is working on, etc.) For example, if there is a huge for-loop in your code, print an asterisk('*') after every 10,000 iterations; c) Have your code automatically save your results as an rda file every so often. This allows you to check your results in a different R console while the program is running and retain your results if your R program prematurely terminates; d) You can run R from the command line to avoid using the R Studio interface. This will typically make your code faster.*

Hint 3: We first need to figure out what the data matrix $\mathbf{X}_1$ *is for our regression task. It is worth noting that* $\mathbf{X}_1$ *is now different from the matrix* $\mathbf{X}$ *that we load from* `ratings.dat`*.* $\mathbf{X}_1$ *is of the same dimensions as* $\mathbf{X}$*, but to get* $\mathbf{X}_1$*, we need to replace the ProfileID's and UserID's in* $\mathbf{X}$ *with the associated average ratings as explained in the model. (See the code given in Hint 1 for reference.) The subsampling technique essentially means that we randomly choose a small number rows of* $\mathbf{X}_1$*, whose row indices form a set* I*, and do regression only over* X_{1_I}*, which is a sub-matrix of* $\mathbf{X}_1$ *that consists of only rows* I*. Since we use only a small number of samples, the regression coefficients we get will not be accurate. A natural way to solve this problem is to apply this subsampling trick for multiple times and average the coefficients we get. In this way, we expect the result to be more stable and accurate.*