

GU4206-GR5206 Final Exam [100 pts]

Student (UNI)

May 4th, 2018

The STAT Spring 2018 GU4206-GR5206 Final Exam is open notes, open book(s), open computer and online resources are allowed. Students are **not** allowed to communicate with any other people regarding the final with the exception of the instructor (Gabriel Young) and TA (Fan Gao). This includes emailing fellow students, using WeChat and other similar forms of communication. If there is any suspicious of students cheating, further investigation will take place. If students do not follow the guidelines, they will receive a zero on the exam and potentially face more severe consequences. The exam will be posted on Canvas at 9:00AM. Students are required to submit both the .pdf and .Rmd files on Canvas (or .html if you must) by 12:00PM. Late exams will not be accepted. If for some reason you are unable to upload the completed exam on Canvas by 12:00PM, then immediately email the markdown file to the course TA.

Part 1: Simulation [40 pts]

In this section, we consider a **mixture** of two normal distributions. Here we assume that our random variable is governed by the probability density $f(x)$, defined by

$$\begin{aligned} f(x) &= f(x; \mu_1, \sigma_1, \mu_2, \sigma_2, \delta) \\ &= \delta f_1(x; \mu_1, \sigma_1) + (1 - \delta) f_2(x; \mu_2, \sigma_2) \\ &= \delta \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp -\frac{1}{2\sigma_1^2}(x - \mu_1)^2 + (1 - \delta) \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp -\frac{1}{2\sigma_2^2}(x - \mu_2)^2, \end{aligned}$$

where $-\infty < x < \infty$ and the parameter space is defined by $-\infty < \mu_1, \mu_2 < \infty$, $\sigma_1, \sigma_2 > 0$, and $0 \leq \delta \leq 1$. The **mixture parameter** δ governs how much mass gets placed on the first distribution $f(x; \mu_1, \sigma_1)$ and the complement of δ governs how much mass gets placed on the other distribution $f_2(x; \mu_2, \sigma_2)$.

In our setting, suppose that we are simulating $n = 10,000$ heights from the population of both males and females. Assume that males are distributed normal with mean $\mu_1 = 70$ [in] and standard deviation $\sigma_1 = 3$ [in] and females are distributed normal with mean $\mu_2 = 64$ [in] and standard deviation $\sigma_2 = 2.5$ [in]. Also assume that each distribution contributes equal mass, i.e., set the mixture parameter to $\delta = .5$. The distribution of males is governed by

$$f_1(x; \mu_1, \sigma_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp -\frac{1}{2\sigma_1^2}(x - \mu_1)^2, \quad -\infty < x < \infty,$$

and the distribution of females is governed by

$$f_2(x; \mu_2, \sigma_2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp -\frac{1}{2\sigma_2^2}(x - \mu_2)^2, \quad -\infty < x < \infty.$$

The goal is to **simulate** from the **mixture distribution**

$$\delta f_1(x; \mu_1, \sigma_1) + (1 - \delta) f_2(x; \mu_2, \sigma_2),$$

where $\mu_1 = 70, \sigma_1 = 3, \mu_2 = 64, \sigma_2 = 2.5, \delta = .5$ using the accept-reject algorithm.

Perform the following tasks:

- 1) [10 pts] Using **ggplot**, graph $f_1(x; \mu_1, \sigma_1)$, $f_2(x; \mu_2, \sigma_2)$ and the mixture $f(x)$ all on the same plot. Make sure the plot includes a legend and is labeled appropriately.

Solution goes here

- 2) [25 pts] Use the **accept-reject** algorithm to simulate from this mixture distribution. To receive full credit:

2.i Clearly identify an **easy to simulate** distribution $g(x)$. I recommend picking a normal distribution or a Cauchy distribution for $g(x)$.

2.ii Identify a **suitable** value of α such that your envelope function $e(x)$ satisfies

$$f(x) \leq e(x) = g(x)/\alpha, \quad \text{where } 0 < \alpha < 1.$$

Note that you must choose α so that $e(x)$ is close to $f(x)$. Show that your α is **suitable** using a plot.

2.iii Simulate 10,000 draws from the mixture distribution using the **accept-reject** algorithm. Display the first 20 simulated values. Also, using **ggplot** or **base R**, construct a histogram of the simulated mixture distribution with the true mixture pdf $f(x)$ overlayed on the plot.

2.i) [5 pts]

Solution goes here

2.ii) [10 pts] By inspection, we see that choosing $\alpha = .44$ allows $e(x)$ to be greater than $f(x)$ for all x and the envelope sits relatively close to the target distribution.

Solution goes here

2.iii) [10 pts] The **Accept-Reject** algorithm is coded below:

Solution goes here

Plot:

Solution goes here

- 3) [5 pts] Slightly change the **Accept-Reject** algorithm from Part (2.iii) to also include the acceptance rate, i.e., how many times did the algorithm accept a draw compared to the total number of trials performed. What proportion of cases were accepted? Compare this number to your chosen α and comment on the result.

Solution goes here

Part II: Maximum Likelihood Estimation and Newton's Method [25 pts]

Recall in logistic regression, the likelihood function is derived by **linking** the mean of Y_i with a linear function.

Logistic Regression Model:

Let Y_1, Y_2, \dots, Y_n be independently distributed Bernoulli random variables with respective success probabilities p_1, p_2, \dots, p_n . Then the **logistic regression model** is:

$$E[Y_i] = p_i = \frac{e^{(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip})}}{1 + e^{(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip})}}, \quad i = 1, 2, \dots, n.$$

Notice with some simple algebra, the above model can be expressed as:

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{i,p} x_i, \quad i = 1, 2, \dots, n.$$

The main idea is to **link** the expected value of Y_i ($E[Y_i] = p_i$) to a linear function. This same principle can be applied to other settings.

Data Description

Consider a geriatrics experiment designed as a prospective study to investigate the effects of two interventions on the frequency of falls. One hundred subjects were randomly assigned to one of the two interventions: education only ($X_1 = 0$) and education plus aerobic exercise training ($X_1 = 1$). Subjects were at least 65 years of age and in reasonably good health. Three variables considered to be important as control variables were gender ($X_2 : 0 = \text{female}, 1 = \text{male}$), a balance index (X_3), and a strength index (X_4). The higher the balance index, the more stable the subject; and the higher the strength index, the stronger the subject. Let Y be the number of falls during the six month study.

```
glm.data <- read.table("https://netfiles.umn.edu/users/nacht001/www/nachtsheim/Kutner/Chapter%2014%20Data%20Set.txt")
names(glm.data) <- c("Y", "X1", "X2", "X3", "X4")
head(glm.data)
```

```
##   Y X1 X2 X3 X4
## 1 1  1  0 45 70
## 2 1  1  0 62 66
## 3 2  1  1 43 64
## 4 0  1  1 76 48
## 5 2  1  0 51 72
## 6 1  1  1 73 39
```

In this setting, the response variable takes on discrete count values, therefore it is reasonable to assume Y_1, Y_2, \dots, Y_{100} are independent Poisson random variables with mean $E[Y_i] = \lambda_i$. Here we can choose a link function that relates λ_i to the linear function $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4}$.

Perform the following task:

- 4) [25 pts] Assume Y_1, Y_2, \dots, Y_{100} are independent Poisson random variables with mean

$$E[Y_i] = \lambda_i = \exp(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4})$$

Note that the link function is $\exp(u)$. Use maximum likelihood estimation to estimate the Poisson regression model. To receive full credit:

- 4.i Define the negative log-likelihood function in R. Name the function **pois.neg.ll**.
- 4.ii Test the negative log-likelihood function at the parameter point **rep(0,5)**.
- 4.iii Use the **Newton's Method** or **Gradient Descent** algorithm from class to estimate coefficients $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$. Display the estimated parameters and the number of iterations the algorithm took to converge. For partial credit, you can use **nlm()**.

- 4.i) [10 pts]

```
# Solution goes here
```

- 4.ii) [5 pts]

```
# Solution goes here
```

- 4.iii) [10 pts]

```
# Solution goes here
```

Check the result with `glm()` (optional)

```
glm(Y~X1+X2+X3+X4,data=glm.data,family="poisson")
```

```
##
## Call:  glm(formula = Y ~ X1 + X2 + X3 + X4, family = "poisson", data = glm.data)
##
## Coefficients:
## (Intercept)          X1          X2          X3          X4
##    0.489467   -1.069403   -0.046606    0.009470    0.008566
##
## Degrees of Freedom: 99 Total (i.e. Null);  95 Residual
## Null Deviance:      199.2
## Residual Deviance: 108.8    AIC: 377.3
```

Part III: Cross Validadtion and Linear Regression [35 pts]

The goal of this section is to illustrate how the **degree** of a polynomial model can be thought of as a tuning parameter.

Perform the follwoing task:

- 5) [5 pts] Upload the dataset **finalexamtrain.csv** and plot Y versus x using **ggplot**. Change the points to blue in the plot.

```
# Solution goes here
```

- 6) [5 pts] In this setting, we regress the response variable Y against a single predictor x using five different models:

Linear Model: degree=1

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

Quadratic Model: degree=2

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

Cubic Model: degree=3

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

Quartic Model: degree=4

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

Quintic Model: degree=5

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

Fit each model using the training data. To make your life easier, I recommend using the Inhibit Interpretation function **I()**. Display the estimated coefficients for each model.

```
# Solution goes here
```

- 7) [10 pts] Consider the dataframe **finalexamtest.csv** that contains four validation (test) sets each consisting of 50 cases. Notice that the variable **ValSet** contains four levels. More specifically, the first 50 cases belong to **TestSet1**, the next 50 cases belong to **TestSet2**, ...etc.

```
data.test <- read.csv("finalexamtest.csv")
head(data.test)
```

```
##      x      Y  ValSet
## 1 0.67 45.00438 TestSet1
## 2 3.23 386.60818 TestSet1
## 3 1.54 284.04469 TestSet1
## 4 2.42 391.53059 TestSet1
## 5 0.50  1.37337 TestSet1
## 6 1.18 262.60680 TestSet1
```

```
tail(data.test)
```

```
##      x      Y  ValSet
## 195 1.00 147.2985 TestSet4
## 196 3.12 296.8062 TestSet4
## 197 3.42 302.9193 TestSet4
## 198 2.09 263.6752 TestSet4
## 199 1.69 212.8365 TestSet4
## 200 1.65 303.2247 TestSet4
```

```
levels(data.test$ValSet)
```

```
## [1] "TestSet1" "TestSet2" "TestSet3" "TestSet4"
```

Write a function named **test.error** that inputs a dataframe and outputs the **test error** for each model. The function should give four predictions errors corresponding to each model, i.e., linear, quadratic, cubic, quartic, and quintic. Note that you are still using the trained models from Part (6). You are allowed to hard code some of this function. Test the **test.error** function on the full training data.

Hint: `y.test <- predict(model,newdata = data.test[“x”])`

```
# Solution goes here
```

- 8) [10 pts] Use the **Split/Apply/Combine** model to compute the test error for each test set: **TestSet1**, **TestSet2**, **TestSet3**, **TestSet4**. You can exhaustively perform this task for partial credit. Display the test errors for each test set and each model. Also display the average test error over each validation set.

```
# Solution goes here
```

- 9) [5 pts] Create a plot (base R or ggplot) showing both the **average test error** and the **training error** as a function of the polynomial's **degree**. Briefly comment on this plot.

```
# Solution goes here
```