Homework 5 [60 points] / Updates in magenta

Questions related to the homework can be posted on the Piazza forum site, but do not post answers to homework problems. We will try to respond promptly to posted questions. You may send questions after 9pm the day before the assignment is due, but there is no guarantee they will be answered (on time).

You may discuss the homework problems and computing issues with other students in the class. However, you must write up your homework solution on your own. In particular, do not share your code or homework files with other students. If you collaborate with other students list their names at the beginning of your writeup.

This homework will build upon the material presented in labs and lectures, and will be focused on multivariate root finding / optimization, and on the Bernoulli / Binomial distribution. All programming should be done in `R`. If there are any difficulties starting the homework, please contact the TA.

The homework is split into two parts:

Part 1: Root Finding For Maximum Likelihood Estimation (Word Pairs)

Part 2: Root Finding For Maximum Likelihood Estimation (O Ring Data)

For Part One, text in blue denotes what needs to be done.

Points will be allocated to good code style and (mathematical) clarity. Points *may* be deducted if homework is unreadable, so take note of this.

Files to submit on CMS

Do check to see that you have submitted all these files. Each part will be graded as a whole. The writeups should either be TeXed up or done using R Markdown.

1. Part One [30 points]
    - `return_estimates.r`
    - `writeup_one.html` or `writeup_one.pdf`
2. Part Two [30 points]
    - `oringCA.r`
    - `oringNR.r`
    - `writeup_two.html` or `writeup_two.pdf`

Root Finding For Maximum Likelihood Estimators (Word Pairs)

Maximum likelihood estimators play an important part in statistics, finding the optimal parameters of a model given some observations. In Homework 3, you've computed the likelihood function of a Poisson distribution, and found the maximum likelihood estimator by equating the derivative to zero, and solving for the parameters.

Most of the time however, we can't get a nice form of the derivative, and thus have to resort to root finding methods. In this problem, we will look at the multinomial distribution, with respect to contingency tables.

The Multinomial MLE (warm up)

Suppose we have p random variables $X_1, X_2, \ldots, X_p$ where:

$$\begin{aligned}\mathbb{P}\left[X_1 = n_1, X_2 = n_2, \ldots, X_n = n_p\right] &= \binom{n}{n_1\ n_2\ \ldots\ n_p} \prod_{i=1}^{p} \pi_i^{n_i} \\ &= \frac{n!}{n_1! n_2! \ldots n_p!} \prod_{i=1}^{p} \pi_i^{n_i}\end{aligned}$$

where $\sum_{i=1}^{p} n_i = n,\ n_i \geq 0$ and $\sum_{i=1}^{p} \pi_i = 1,\ \pi_i > 0$.

The joint distribution of $X_1, \ldots, X_p$ is a multinomial distribution.

Note that when $p = 2$, this simplifies to the binomial distribution. More generally, we see the multinomial distribution in situations where we have contingency tables.

For example, suppose we were doing a study on Hepatitis B and tattoo parlors. Suppose we observed n people where we had:

	Has Hepatitis B	No Hepatitis B
Visited Tattoo Parlor in last six months	n_1	n_2
Did not visit Tattoo Parlor at all	n_3	n_4

with $n_1 + n_2 + n_3 + n_4 = n$.

Here, we're interested in the *probabilities* π_i of drawing a person from our population that belongs to one of the four categories. We would then find the MLEs of $\pi_1, \pi_2, \pi_3, \pi_4$.

We will now try to find the maximum likelihood estimates $\widehat{\pi}_1, \widehat{\pi}_2, \ldots, \widehat{\pi}_p$. We proceed to do this by trying to generalize from an easy distribution which we know - the binomial distribution.

Note that we could write the binomial distribution as such:

$$\mathbb{P}[X_1 = n_1, X_2 = n_2] = \binom{n}{n_1\ n_2} \pi_1^{n_1} \pi_2^{n_2} = \frac{n!}{n_1! n_2!} \pi_1^{n_1} \pi_2^{n_2} \tag{0.1}$$

where $n_1 + n_2 = n$, $\pi_1 + \pi_2 = 1$. Conventionally, we would instead write this in terms of k successes and $n - k$ failures, with:

$$\mathbb{P}[X_1 = k, X_2 = n - k] = \binom{n}{k} p^k (1-p)^{n-k} \tag{0.2}$$

Using Equation 0.2, write down the log likelihood function for the binomial distribution, and derive the maximum likelihood estimate $\hat{p}$.

Paying careful attention to notation, argue carefully how your answer helps you find the maximum likelihood estimates of both π_1 and π_2 in Equation 0.1.

This gives us an idea on how to proceed to find the maximum likelihood estimates of a multinomial distribution. To check if our idea has any merit, we should consider one more case - the *trinomial* distribution, where we write:

$$\mathbb{P}[X_1 = n_1, X_2 = n_2, X_3 = n_3] = \binom{n}{n_1\ n_2\ n_3} \pi_1^{n_1} \pi_2^{n_2} \pi_3^{n_3}$$

where $n = n_1 + n_2 + n_3$, $n_i \geq 0$, and $\sum_{i=1}^{3} \pi_i = 1$, $\pi_i > 0$.

By setting $\pi_j = 1 - \sum_{i \neq j} \pi_i$, find the maximum likelihood estimates $\widehat{\pi}_1, \widehat{\pi}_2$, and $\widehat{\pi}_3$.

We should be able to take on the general multinomial distribution now.

For the general multinomial distribution, carefully derive the maximum likelihood estimates $\widehat{\pi}_i$ using what you have shown in the previous part, defining the notation you use and any assumptions you make.

Computing The MLE When Margins Are Known

We will now be considering the MLE when we know the margins for a 2×2 contingency table. This method is used in many modern day applications, and it all stems from this 1940 paper: `On A Least Squares Adjustment Of A Sampled Frequency Table When The Expected Marginal Totals are Known` by Deming and Stephan. You may check the paper out here `https://www.jstor.org/stable/pdf/2235722.pdf` for further reading, but this is not required.

The Set Up

We will utilize this method in the *document* - *words* scenario, similar to HW3. Suppose we had this extremely large matrix D filled with 1s and 0s.

```
                    w1  w2  w3  w4  ...
              doc1  0   1   1   1
              doc2  1   1   1   0
D     =       doc3  0   0   0   1
              doc4  0   1   0   0
               .
               .
               .
```

A 1 in the $(i, j)^{\text{th}}$ entry means that word j appears in document i. Without loss of generality, we will look at the contingency table for `Word1` and `Word2`.

	`Word2`	`No Word2`
`Word1`	N_1	N_2
`No Word1`	N_3	N_4

Supposing we had the matrix D, explain briefly how we would use this matrix to compute the exact values N_1, N_2, N_3, and N_4. Do you need the values of N_1, N_2, N_3, N_4 in order to find $N := N_1 + N_2 + N_3 + N_4$? Explain your answer.

The theory says that we pre-compute the *margins* of the table. We denote them to be M_1 and M_2, where $M_1 = N_1 + N_2$, and $M_2 = N_1 + N_3$.

Suppose we are interested in estimating $\widehat{N}_1, \widehat{N}_2, \widehat{N}_3$ and $\widehat{N}_4$, and we drew a random sample of documents

	Word2	No Word2
Word1	n_1	n_2
No Word1	n_3	n_4

from our universe of all documents (i.e. from the matrix D). Then we just need to find an estimate $\widehat{N}_1$, because we can compute $\widehat{N}_2, \widehat{N}_3$ and $\widehat{N}_4$ from the margins.

By referring to the previous contingency table, and the maximum likelihood estimates of the multinomial distribution, derive an expression for the maximum likelihood estimate of $\widehat{N}_1$ given that you have observed n_1, n_2, n_3, n_4, and know M_1 and M_2. Your answer should carefully explain every step you take.

Bonus: Write this expression as a polynomial and directly solve for the root without any numerical methods.

A Practical Scenario

We will now test out how useful our estimates are. While we do not have a matrix D, we do have our words from HW3, so let's use them.

Write a function:

```
return_estimates<-function(word1, word2, seed, sample_size){

  # word1 from our words in HW3
  # word2 from our words in HW3
  # seed to be the seed we start with (for comparison)
  # sample_size the number of documents sampled

  # The function should do the following:
  # 1. Randomly choose a subset of documents (given below)
  # 2. For any word1 and word2 vector
  #      + Compute n1, n2, n3, n4
  #      + Compute M1, M2
  # 3. Use your favourite root finding algorithm to
  #    find an estimate of N_1_hat
  # 4. Return a vector (N_1_hat, N_2_hat, N_3_hat, N_4_hat)

  ## Leave below unchanged
  set.seed(seed)
```

```
  docs = sample(0:65535,sample_size)
  ## Leave above unchanged

  ## Put code here

  return(c(N1_hat, N2_hat, N3_hat, N4_hat))
}
```

Submit `return_estimates.r` to CMS, and put a copy of this in your writeup. You may embed helper functions *within* this function. Intuitive variable names / commenting code is optional, but may result in partial credit.

We will now consider the following word pairs:

- `primary election`
- `south dakota`
- `north carolina`

Run this function and display the estimates of $\widehat{N}_1, \widehat{N}_2, \widehat{N}_3, \widehat{N}_4$ for these word pairs as well as the original N_1, N_2, N_3, N_4. Set `sample_size` to be 10,000, the `seed` to be the digits of your netID[1]. Comment on the estimates you get and the true values.

Computing the marginal values M_1, M_2 as well as running a root finding algorithm to get $\widehat{N}_1, \widehat{N}_2, \widehat{N}_3$ and $\widehat{N}_4$ seem to be more work than just computing the maximum likelihood estimates directly. Are there any reasons why we would prefer the former in practice? If yes, justify them. If no, explain why we would prefer the latter[2]. Your answer should focus more on practical cases rather than academic cases[3].

[1] If your netID is `pfv2`, then set the seed to be 2. If your netID is `pa338`, then set the seed to be 338.
[2] Eg, what is "more work"?
[3] See discussion in the second footnote in the solutions for HW4 as an example.

Root Finding For Maximum Likelihood Estimators (O Ring Data)

This problem involves data from the Challenger space shuttle which exploded after takeoff on January 28, 1986.

`https://en.wikipedia.org/wiki/Space_Shuttle_Challenger_disaster`

The data in "shuttle.csv" contains three variables: flight, ndo and temp. The variable flight is simply a code for the launch numbers (prior to Jan. 28, 1986) and can be ignored in what follows. The variable ndo is the number of damage o-rings, which was determined after the solid rocket boosters were retrieved from the Atlantic ocean following each launch. The variable temp is the temperature at the time of each launch.

Let y_i be an indicator of o-ring damage during the ith launch and let $\pi_i = P(y_i = 1)$. Consider the following logistic regression model for the probability of o-ring damage:

$$\ln \frac{\pi_i}{1-\pi_i} = \beta_0 + \beta_1 x_i \,,$$

where x_i is the temperature at the ith launch.

1. Assuming all the launches are independent (which seems reasonable), derive the likelihood and log-likelihood functions for the regression parameter vector $\beta = (\beta_0, \beta_1)$. Show that a sufficient statistic is $(\sum_i y_i, \sum_i x_i y_i)$.

2. Modify the coordinate ascent and golden section code from Lecture 10 to determine the ML estimates of the regression coefficients. Use the starting values $\beta_0 = \ln \frac{\bar{y}}{1-\bar{y}}$ and $\beta_1 = 0$, and report the number of iterations required for the algorithm to converge. Submit your code as "oringCA.R".

   The submission of your function should be of this form:

   ```
   oringCA<-function(filename){

     data = read.csv(filename)
     # Code here
     return(c(num_iter,b0,b1))
   }
   ```

   You can assume that `filename` is a name of a CSV file. Function checking will comprise of using CSV files *similar to* the provided `shuttle.csv` file. However, points for this

question will also be awarded for a well documented function (intuitive variable names, comments, etc). A function which is not commented / has no intuitive variable names and has a *off by one* error or small typo would get zero credit, but a function that is well documented and has the same errors would get near full credit.

3. Derive formulas for the gradient vector and the Hessian matrix.

4. Write a Newton-Raphson algorithm to determine the maximum likelihood estimates of the regression coefficients. Report the iteration history using the same starting values for coordinate ascent. Submit your R code as "oringNR.R".

   The submission of your function should be of this form:

```
oringNR<-function(filename){
  data = read.csv(filename)

  # Code here
  return(histOR)
}
```

   You can assume that `filename` is a name of a CSV file. Function checking will comprise of using CSV files *similar to* the provided `shuttle.csv` file.

   `histOR` should be a matrix with 3 columns. The first column should be the iteration number, second column the values of β_0 at each iteration, and the third column the values of β_1 at each iteration.

   Points for this question will also be awarded for a well documented function (intuitive variable names, comments, etc). A function which is not commented / has no intuitive variable names and has a *off by one* error or small typo would get zero credit, but a function that is well documented and has the same errors would get near full credit.

5. Construct a plot of the probability of o-ring damage as a function of temperature. What is the predicted probability of o-ring damage when temperature at launch is 32°F? Include code and plot in your writeup.

6. Fit the logistic regression model using the **glm** function in R and compare the estimates to those obtained using the CA and NR methods. Include the code in your writeup.

7. Construct a histogram of the permutation distribution of $\sum_i x_i y_i$, by permuting the vector of y-values. Indicate where the observed value falls on the histogram. Determine the proportion of permutation values that are less than or equal to the observed value.

   Include the code and plot in your writeup.

   original value in data of y

8. Determine the exact permutation probability of a value less than or equal to the observed. (You may write your own code or use an existing R program to do this.)

   Include the code in the writeup.