Problem Set 2: Written Part

Your Name Goes Here

Details

How to Write Up

The written part of this assignment can be either typeset using latex or hand written.

Grading

5% of your grade on this assignment is for turning in something legible. This means it should be organized, and any Rmd files should knit to pdf without issue.

An additional 15% of your grade is for completion. A quick pass will be made to ensure that you've made a reasonable attempt at all problems.

Across both the written part and the R part, in the range of 1 to 3 problems will be graded more carefully for correctness. In grading these problems, an emphasis will be placed on full explanations of your thought process. You don't need to write more than a few sentences for any given problem, but you should write complete sentences! Understanding and explaining the reasons behind what you are doing is at least as important as solving the problems correctly.

Solutions to all problems will be provided.

Collaboration

You are allowed to work with others on this assignment, but you must complete and submit your own write up. You should not copy large blocks of code or written text from another student.

Sources

You may refer to our text, Wikipedia, and other online sources. All sources you refer to must be cited in the space I have provided at the end of this problem set.

Problem I: M&M's

Starting the week after next, we will be looking at Bayesian inference. The central idea in Bayesian statistics is that probability is used to express our state of knowledge about the value of an unknown parameter. The written part of this problem should take you 1 or 2 minutes, and is intended to introduce this way of thinking about probability.

Consider the question "What proportion of peanut M&M's are blue?" Note that the maker of M&M's has set the proportion of peanut M&M's that are blue, but they have not published this number. We will denote this proportion by θ. Although we don't know the value of θ, we can probably make some reasonable guesses about what it is based on our past experience with eating M&M's.

(1) What is your best guess at the proportion of M&M's that are blue? This should be a number between 0 and 1.

Answers will vary. Personally, my best guess is that the proportion of M&M's that are blue is 0.17.

(2) Give an interval $[a, b]$ so that you think there is a 50% chance the proportion of M&M's that are blue is in that interval. In other words, you are picking numbers a and b between 0 and 1 so that you think $P(\theta \in [a, b]) = 0.5$. This is called a 50% credible interval. We'll return to this idea in class later.

Answers will vary. Personally, I think there's probability 0.5 that the proportion of M&M's that are blue is between about 0.1 and 0.2.

(3) Give an interval $[c, d]$ so that you think there is a 90% chance that the proportion of M&M's that are blue is in that interval. In other words, you are picking numbers c and d between 0 and 1 so that you think $P(\theta \in [c, d]) = 0.9$. This is called a 90% credible interval. We'll return to this idea in class later.

Answers will vary. Personally, I think there's probability 0.9 that the proportion of M&M's that are blue is between about 0.05 and 0.3.

Problem II: Method of Moments

(1) Suppose that $X_i \overset{\text{i.i.d.}}{\sim}$ Geometric(p) for $i = 1, \ldots, n$. Find the method of moments estimator of p.

Note that there are multiple parameterizations of the Geometric distribution. Please use the parameterization as given on the "Common Probability Distributions" handout.

If $X \sim$ Geometric(p) then $E(X) = \frac{1-p}{p}$. To find the method of moments estimator of p, we set this expected value equal to the sample mean and solve for p:

$$\begin{aligned} &\frac{1-p}{p} = \bar{X} \\ &\Rightarrow 1 - p = p\bar{X} \\ &\Rightarrow 1 = p(\bar{X} + 1) \\ &\Rightarrow p = \frac{1}{1+\bar{X}} \end{aligned}$$

Our final estimator is $\widehat{p} = \frac{1}{1+\bar{X}}$. Note that this is an estimator so I have written the result with a capital $\bar{X}$.

Problem III: Wind Speeds

This problem is adapted from an example in "Mathematical Statistics with Resampling and R" by Chihara and Hesterberg (2011), who write:

> "[U]nderstanding the characteristics of wind speed is important. Engineers use wind speed information to determine suitable locations to build a wind turbine or to optimize the design of a turbine. Utility companies use this information to make predictions on engery avaiabilty during peak demand periods (say, during a heat wave) or to estimate yearly revenue.
>
> The Weibull distribution is the most commonly used probability distribution used to model wind speed ... The Weibull distribution has a density function with two parameters, the shape parameter $k > 0$ and the scale parameter $\lambda > 0$."

If $X \sim$ Weibull(k, λ), then it has pdf

$$f(x|k, \lambda) = \frac{kx^{k-1}}{\lambda^k} e^{-(x/\lambda)^k}$$

We will consider fitting a Weibull distribution to measurements of daily average wind speeds in meters per second at the site of a wind turbine in Minnesota over the course of 168 days from February 14 to August 1, 2010 (there were no data for July 2). Although these data are gathered over time, let's treat the measurements as independent (this is unrealistic but will make the problem more approachable).

We adopt the model $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim}$ Weibull(k, λ).

The likelihood function is

$$\begin{aligned}\mathcal{L}(k,\lambda|x_1,\ldots,x_n) &= \prod_{i=1}^{n} \frac{k x_i^{k-1}}{\lambda^k} e^{-(x_i/\lambda)^k} \\ &= \frac{k^n}{\lambda^{kn}} \exp\left\{-\sum_{i=1}^{n}(x_i/\lambda)^k\right\} \prod_{i=1}^{n} x_i^{k-1}\end{aligned}$$

The log-likelihood is

$$\ell(k,\lambda|x_1,\ldots,x_n) = n\log(k) - kn\log(\lambda) + (k-1)\sum_{i=1}^{n}\log(x_i) - \sum_{i=1}^{n}\left(\frac{x_i}{\lambda}\right)^k$$

The partial derivatives of the log-likelihood with respect to the unknown parameters are:

$$\frac{\partial}{\partial k}\ell(k,\lambda|x_1,\ldots,x_n) = \frac{n}{k} - n\log(\lambda) + \sum_{i=1}^{n}\log(x_i) - \sum_{i=1}^{n}\left(\frac{x_i}{\lambda}\right)^k \log\left(\frac{x_i}{\lambda}\right) \tag{1}$$

and

$$\frac{\partial}{\partial \lambda}\ell(k,\lambda|x_1,\ldots,x_n) = \frac{-kn}{\lambda} + \frac{k}{\lambda^{k+1}}\sum_{i=1}^{n} x_i^k \tag{2}$$

Setting (2) equal to 0 and solving for λ gives

$$\lambda = \left[\frac{1}{n}\sum_{i=1}^{n} x_i^k\right]^{1/k} \tag{3}$$

Substituting this into (1) and setting it equal to 0 gives

$$\frac{1}{k} + \frac{1}{n}\sum_{i=1}^{n}\log(x_i) - \frac{\sum_{i=1}^{n} x_i^k \log(x_i)}{\sum_{i=1}^{n} x_i^k} = 0$$

This equation cannot be analytically solved for k; numerical optimization methods must be used to maximize the log-likelihood. Note that if we plug (3) back into the log-likelihood function, we obtain a function of just k to maximize:

$$\begin{aligned}\tilde{\ell}(k|x_1,\ldots,x_n) &= n\log(k) - kn\log\left[\left\{\frac{1}{n}\sum_{i=1}^{n} x_i^k\right\}^{1/k}\right] + (k-1)\sum_{i=1}^{n}\log(x_i) - \sum_{i=1}^{n}\left(\frac{x_i}{\left\{\frac{1}{n}\sum_{i'=1}^{n} x_{i'}^k\right\}^{1/k}}\right)^k \\ &= n\log(k) - n\log\left[\frac{1}{n}\sum_{i=1}^{n} x_i^k\right] + (k-1)\sum_{i=1}^{n}\log(x_i) - \sum_{i=1}^{n}\frac{x_i^k}{\frac{1}{n}\sum_{i'=1}^{n} x_{i'}^k} \\ &= n\log(k) - n\log\left[\frac{1}{n}\sum_{i=1}^{n} x_i^k\right] + (k-1)\sum_{i=1}^{n}\log(x_i) - n\end{aligned}$$

This is not the log-likelihood function, but if we find the value of k that maximizes $\tilde{\ell}(k|x_1,\ldots,x_n)$, we will have found the value of k that maximizes the log-likelihood.

The first and second derivatives of $\tilde{\ell}(k|x_1,\ldots,x_n)$ are:

$$\frac{d}{dk}\tilde{\ell}(k|x_1,\ldots,x_n) = \frac{n}{k} - n\frac{\sum_{i=1}^{n} x_i^k \log(x_i)}{\sum_{i=1}^{n} x_i^k} + \sum_{i=1}^{n}\log(x_i)$$

and

$$\frac{d^2}{dk^2}\tilde{\ell}(k|x_1,\ldots,x_n) = \frac{-n}{k^2} - n\frac{\sum_{i=1}^n x_i^k \{\log(x_i)\}^2}{\sum_{i=1}^n x_i^k} - n\frac{\left\{\sum_{i=1}^n x_i^k \log(x_i)\right\}^2}{\left\{\sum_{i=1}^n x_i^k\right\}^2}$$

(1) Write down a complete statement of a Newton algorithm for finding a maximum likelihood estimate of k by maximizing the function $\tilde{\ell}$. Your algorithm will run until either a user-specified maximum number of iterations is reached or a user-specified tolerance is reached for the magnitude of the change in estimates of k on consecutive iterations. Include a specific statement of how the estimate of k will be updated in each iteration of the algorithm. (You do not need to do any new calculus for this – all the results you need are above.)

There are multiple valid ways to write down the algorithm; here is one example:

Our algorithm takes three inputs:

- A starting value for k, denoted k_0.
- A maximum number of iterations to run.
- A tolerance ε used to determine when to stop the algorithm.

Algorithm:

1. Initialize `k_current =` k_0 and `last_change =` ∞ (or anything larger than ε)
2. Repeat until the maximum number of iterations has been reached or `last_change` $< \varepsilon$.
   a. Set `k_previous` = `k_current`
   b. Set `k_current` = `k_previous` - $\frac{\frac{d}{dk}\tilde{\ell}(\text{k\_current}|x_1,\ldots,x_n)}{\frac{d^2}{dk^2}\tilde{\ell}(\text{k\_current}|x_1,\ldots,x_n)}$
   c. Set `last_change` = | `k_previous` − `k_current` |

(2) Once the algorithm you specified in part a has finished running, how could you use the results to find the maximum likelihood estimate of λ? (You can basically just write down one of the equations derived above.)

The above procedure yields a maximum likelihood estimate of k. We could obtain a maximum likelihood estimate of λ by plugging the maximum likelihood estimate of k into the equation below:

$\lambda = \left[\frac{1}{n}\sum_{i=1}^n x_i^k\right]^{1/k}$

(3) Write a 1 or 2 paragraph explanation of what Newton's method is in the context of optimization. Please explain what the goal of the method is, how the method works, and how it relates to Taylor's theorem. You should include a supporting illustration (this can be drawn by hand).

Newton's method can be characterized in either of two ways:

1. An algorithm for finding the *roots* of a function $f(z)$; or
2. An algorithm for finding the *maximum* or *minimum* of a function $g(z)$.

In optimization, the second view of the algorithm is usually taken; however, note that if z^* is a maximum or minimum of the function g, then $\frac{d}{dz}g(z^*) = 0$. If we set $f(z) = \frac{d}{dz}g(z)$, the problem of finding a maximum or minimum of a function can be framed in terms of finding a root of its derivative. The two ways of framing Newton's method can both be used to solve an optimization problem.

In our applications in statistics, $g(z)$ has been a log-likelihood function $\ell(\theta|\mathbf{x})$ and $f(z)$ would correspond to the first derivative of the log-likelihood function, $\frac{d}{d\theta}\ell(\theta|\mathbf{x})$. To be more firmly rooted in our application, I will switch now to talking about optimizing the log-likelihood or finding roots of its first derivative.

To be even more concrete, let's consider the example we used when we explored this method in lab 3. There, we had a binomial model for the count of children in a sample of size $n = 2700$ who were deficient in vitamin D. We observed $x = 540$ children with the deficiency, and our goal is to estimate the parameter θ in the model $X \sim \text{Binomial}(2700, \theta)$ for these data. The likelihood function is $\ell(\theta|x = 540, n = 2700) = \log\left\{\binom{2700}{540}\right\} + 540\log(\theta) + (2700 - 540)\log(1 - \theta)$. Its first derivative is $\frac{d}{d\theta}\ell(\theta|x = 540, n = 2700) = \frac{540}{\theta} + (2700 - 540)\frac{1}{1-\theta}(-1)$.

In both views of the algorithm, the idea is to start with an initial guess θ_0 of the value that solves the problem at hand. We will iteratively update our guess until we arrive at either a root of the function $f(\theta) = \frac{d}{d\theta}\ell(\theta|x = 540, n = 2700)$ or an maximum of $g(\theta) = \ell(\theta|x = 540, n = 2700)$. The way we derive the update is different in the two views of the algorithm, but the resulting update is the same either way.

In the first view of the algorithm, where we are finding a root of $f(\theta)$, the method works by forming a *first-order* Taylor series approximation (that is, a linear approximation) to $f(\theta)$ at our current guess. That first-order approximation is given by

$$P_1(\theta) = f(\theta_0) + \frac{d}{d\theta}f(\theta)|_{\theta=\theta_0}(\theta - \theta_0).$$

Locally, this line is a good approximation to $f(\theta)$; ideally, its root will be near the root of f, or at least closer than we started. Setting $P_1(\theta) = 0$ and solving for θ, we obtain $\theta = \theta_0 - \frac{f(\theta_0)}{\frac{d}{d\theta}f(\theta_0)}$. We then repeat this process again and again until our updates are small enough that we declare the algorithm has converged to a root of f. Recalling that $f(\theta) = \frac{d}{d\theta}\ell(\theta|x = 540, n = 2700)$, I'll note that the update can be written in terms of derivatives of the log-likelihood as $\theta = \theta_0 - \frac{\frac{d}{d\theta}\ell(\theta_0|x=540,n=2700)}{\frac{d^2}{d\theta^2}\ell(\theta_0|x=540,n=2700)}$. A more formal statement of the algorithm was given in part (a).

The second view of the algorithm is similar, but we now form a *second-order* Taylor series approximation to $g(\theta)$ around our current guess at a maximum of g, and identify a maximum of that second-order approximation as our updated guess. Specifically, our second-order approximation is given by

$$P_2(\theta) = g(\theta_0) + \frac{d}{d\theta}g(\theta)|_{\theta=\theta_0}(\theta - \theta_0) + \frac{1}{2}\frac{d^2}{d\theta^2}g(\theta)|_{\theta=\theta_0}(\theta - \theta_0)^2.$$

To maximize P_2, we set its derivative equal to 0 and solve for θ (if g was locally concave at θ_0, this critical point will be a maximum of P_2):

$$\begin{aligned}
0 &= \frac{d}{d\theta}P_2(\theta) \\
\Rightarrow 0 &= \frac{d}{d\theta}g(\theta)|_{\theta=\theta_0} + \frac{d^2}{d\theta^2}g(\theta)|_{\theta=\theta_0}(\theta - \theta_0) \\
\Rightarrow 0 &= \frac{d}{d\theta}g(\theta)|_{\theta=\theta_0} + \frac{d^2}{d\theta^2}g(\theta)|_{\theta=\theta_0}(\theta - \theta_0) \\
\Rightarrow \theta &= \theta_0 - \frac{\frac{d}{d\theta}\ell(\theta_0|x = 540, n = 2700)}{\frac{d^2}{d\theta^2}\ell(\theta_0|x = 540, n = 2700)}.
\end{aligned}$$

Note that this is exactly the same update that we obtained from the root-finding view of the process above. Again, we iterate this process until our updates are small enough that we declare convergence.

Below is a picture (taken from the slides introducing the method) showing the first step in this procedure. The initial value of the parameter is shown as λ_0 along the horizontal axis. In the top panel we approximate the log-likelihood by a second-order Taylor series, and the next update (labeled as λ_1) maximizes this Taylor approximation. In the lower panel we approximate the first derivative of the log-likelihood by a first-order Taylor series, and the next update is the value λ_1 at which this approximation crosses the horizontal axis.