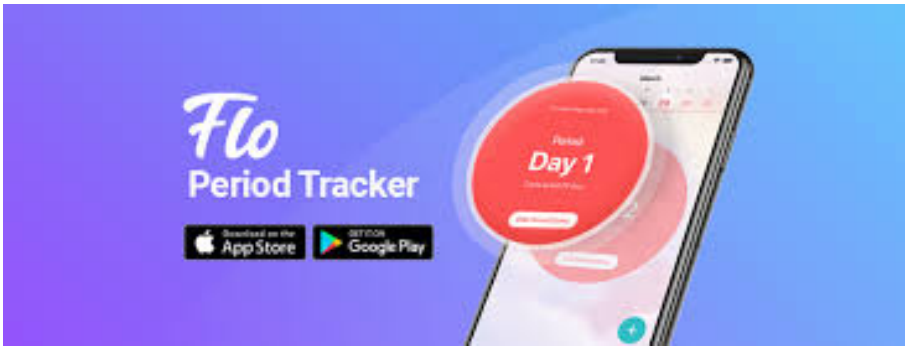
Juliette Woodrow & CS 109 TAs
CS 109

Section #9
March 12th, 2025

Section 9 Solutions

1 MLE Flo



Let X represent the length of a menstrual cycle: the number of days, as a continuous value, between the first moment of one period to the first moment of the next, for a given person. X is parameterized by α and β with probability density function:

$$f(X = x) = \beta \cdot (x - \alpha)^{\beta - 1} \cdot e^{-(x-\alpha)^{\beta}}$$

1. For a particular person, $\alpha = 27$ and $\beta = 2$. Write an expression for the probability that they have their period on day 29. In other words, what is $P(29.0 < X < 30.0)$? You do not need to simplify.

$$P(29.0 < X < 30.0) = \int_{29.0}^{30.0} 2 * (x - 27) * e^{-(x-27)^2} dx$$

2. For a particular person, $\alpha = 27$ and $\beta = 2$. How many times more likely is their cycle to last **exactly** 28.0 days than exactly 29.0 days? Simplify your expression.

This question is asking for the ratio between the probability density at $X = 28$ vs. the probability density as $X = 29$.

$$\frac{f(X = 28)}{f(X = 29)} = \frac{2 * (28 - 27) * e^{-(28-27)^2}}{2 * (29 - 27) * e^{-(29-27)^2}} = \frac{e^3}{2}$$

3. A person has recorded their cycle length for 12 cycles stored in a list: $m = [29.0, 28.5, \ldots, 30.1]$ where m_i is the recorded cycle length for cycle i. Use MLE to estimate the parameter values for α and β. Assume that cycle lengths are IID.

   You don't need a closed-form solution. Derive any necessary partial derivatives and write up to three sentences describing how a program can use the derivatives in order to choose the most likely parameter values.

   Define our likelihood function:

$$L(\alpha, \beta) = \prod_{i=1}^{12} f(m_i)$$

   We'll use log likelihood to make the math easier later:

$$LL(\alpha, \beta) = \sum_{i=1}^{12} \log f(m_i)$$

$$(\alpha, \beta) = \arg\max_{\alpha,\beta} LL(\alpha, \beta)$$

   We can simplify the log of the PDF to:

$$\log f(m) = \log \beta + (\beta - 1) \log(m - \alpha) - (m - \alpha)^{\beta}$$

   Now we take partial derivative w.r.t α and β:

$$\frac{\partial}{\partial \alpha} LL(\alpha, \beta) = \sum_{i=1}^{12} 0 - \frac{\beta - 1}{m_i - \alpha} + \beta(m_i - \alpha)^{\beta - 1}$$

$$\frac{\partial}{\partial \beta} LL(\alpha, \beta) = \sum_{i=1}^{12} \frac{1}{\beta} + \left(1 - (m_i - \alpha)^{\beta}\right) \log(m_i - \alpha)$$

   We can use gradient ascent to maximize LL, using code like the snippet below. This code computes the gradient of the likelihood w.r.t. each parameter, then moves the parameters a small step in the direction of the gradient each iteration.

   Note: Flo is a real "AI based" app that helps people track their period lengths. The real-world distribution is thought to be a mixture between a normal and a Weibull distribution; this problem only has you estimate parameters for a simplified Weibull.

```
num_iterations = 100000

learning_rate = 0.001  # step size
```

```
# Initialize parameters randomly
alpha = some random number
beta = some random number

for i in range(num\_iterations):
    gradient_alpha = 0
    gradient_beta = 0

    for m_i in m:  #  where m is a list of training datapoints
        gradient_alpha += -((beta-1)/(m_i-alpha)) + beta*(m_i-alpha)**(beta-1)
        gradient_beta += (1/beta) + (1-(m_i-alpha)**beta)*log(m_i-alpha)

    alpha = alpha + (learning_rate * gradient_alpha)
    beta = beta + (learning_rate * gradient_beta)
```

2 Evaluating Classifier Performance under Distribution Shift

In general binary classification (for i.i.d. samples), for every sample we draw, we get a set of features X, and a binary label $Y \sim \text{Bern}(p_y)$ for some value of p_y. There also exist distributions D_0 and D_1 such that

$$X \mid Y = 0 \sim D_0 \qquad \text{and} \qquad X \mid Y = 1 \sim D_1$$

Notice that if D_0 and D_1 were identical, then we would have no way of guessing y from x. Training a classifier teaches it to identify differences between samples from D_0 and samples from D_1.

1. Suppose that $p_y = 0.6$, the distribution D_0 has PDF $f_0(x)$, and the distribution D_1 has PDF $f_1(x)$. what is the PDF of the random variable X?

   We use LoTP over the value of Y:

   $$f(X) = f(X \mid Y = 0)P(Y = 0) + f(X \mid Y = 1)P(Y = 1) = \boxed{(1 - p_y)f_0(x) + p_y f_1(x)}$$

2. You have trained a classifier. To evaluate it, you construct an evaluation dataset by using rejection sampling to get 500 samples from the population with $y = 0$ and 500 samples from the population with $y = 1$. Evaluating the classifier on this dataset gives you a prediction $\hat{y}$ for each sample, with the following table. In the table, each cell is the fraction of all evaluation examples for which the true label is the column value and the classifier's prediction is the row value. What is p_y for this evaluation set? What is the accuracy of this classifier on this evaluation set?

	$y = 1$	$y = 0$
$\hat{y} = 1$	0.4	0.2
$\hat{y} = 0$	0.1	0.3

   We defined p_y to be the probability that $Y = 1$, which here is $500/1000 = 0.5$.

   We sum the true positive and true negative rates to get the accuracy: $0.4 + 0.3 = 70\%$.

3. You learn that the true distribution of samples has $p_y = 0.2$. If you use the same classifier, what is the True Positive rate over the true sample distribution? (Assume that $X \mid Y = 0$ and $X \mid Y = 1$ remain the same.)

   Notice that $P(Y = 1)$ has changed, but $P(X = x \mid Y = y)$ remains the same. Since our prediction $\hat{Y}$ is a function of X, this also means that $P(\hat{Y} \mid Y)$ has not changed.

   Thus, the new True Positive rate will be

   $$P(\hat{Y} = 1, Y = 1) = P(\hat{Y} = 1 \mid Y = 1)P(Y = 1)$$

where we use the new $P(Y = 1) = 0.2$, and we have to compute $P(\hat{Y} = 1 \mid Y = 1)$ from the confusion matrix.

Notice that the confusion matrix acts as a Joint Probability Table, which means we can read

$$P(\hat{Y} = 1 \mid Y = 1) = \frac{P(\hat{Y} = 1, \mid Y = 1)}{P(Y = 1)} = \frac{P(\hat{Y} = 1, \mid Y = 1)}{P(\hat{Y} = 1, Y = 1) + P(\hat{Y} = 0, Y = 1)} = \frac{0.4}{0.4 + 0.1} = 0.8$$

so the final true positive rate we get is $0.2 \cdot 0.8 = 0.16$.

Alternatively, we observe from the table that the classifier has 80% accuracy on $y = 1$ samples and 60% accuracy on $y = 0$ samples, so its accuracy will therefore be

$$0.8p_y + 0.6(1 - p_y).$$

Plugging in $p_y = 0.5$ gives the accuracy from Part 2, and plugging in $p_y = 0.2$ gives the accuracy for this part.

4. **(Optional)** Fill out the rest of the updated table. Does the model's accuracy go up or down after the distribution shift?

We fill in the remaining cells similarly to the previous part, to get:

	$y = 1$	$y = 0$
$\hat{y} = 1$	0.16	0.32
$\hat{y} = 0$	0.04	0.48

The overall accuracy is now $0.16 + 0.48 = 64\%$, a reduction from earlier. (This makes sense, since the model was more accurate on $y = 1$ samples than on $y = 0$ results, and the percentage of $y = 1$ samples was decreased.

3 Entropy-Maximizing Distributions

It turns out that, for continuous distributions with infinite support and a fixed mean and variance μ, σ^2, the normal distribution $N(\mu, \sigma^2)$ has maximal entropy. The proof of this is beyond the scope of this course, but this is one way to explain the "why" behind the Central Limit Theorem (adding many IID RVs X_i guarantees the mean and variance of the sum will be specific values, but independence ensures maximum entropy) and also explains why the Normal distribution appears so often in nature.

Let's prove a similar result for discrete distributions with finite support: Prove that, among discrete distributions on $\{1, 2, \ldots, n\}$, the discrete uniform distribution has maximal entropy.

Hint 1: Note that $\log_2 x = \frac{1}{\log 2} \log x$, so maximizing the $\log_2$ version of entropy (which is the default definition we use in this course) will be mathematically equivalent to maximizing the natural-log version of entropy.

Hint 2: To enforce that $\sum_{i=1}^{n} p_i$ must be 1, you can substitute $p_n = 1 - \sum_{i=1}^{n-1} p_i$.

The entropy formula is:

$$-\sum_{i=1}^{n} p_i \log_2 p_i = -\frac{1}{\log 2} \sum_{i=1}^{n} p_i \log p_i$$

so we can ignore the $\log 2$ factor.

Following Hint 2, we rewrite this (minus the $\log 2$ factor) as

$$-\sum_{i=1}^{n-1} p_i \log p_i - \left(1 - \sum_{i=1}^{n-1} p_i\right) \log \left(1 - \sum_{i=1}^{n-1} p_i\right)$$

Taking the derivative with respect to p_1, we get

$$-(1 + \log p_1) - \left(-\log \left(1 - \sum_{i=1}^{n-1} p_i\right) + (-1)\right) = -\log p_1 + \log \left(1 - \sum_{i=1}^{n-1} p_i\right) = -\log p_1 + \log p_n$$

and setting this to 0 gives $p_1 = p_n$. Similarly, we get $p_i = p_n$ for all $i = 1, \ldots, n-1$, so in order to add to 1 all of the probabilities must be $1/n$, i.e. the discrete uniform.

(For those wondering, the equivalent to the CLT in this context is: "If I add many IID RVs with discrete support over $\{0, \ldots, n-1\}$ and take the result mod n, the result will be approximately a discrete uniform over the set of values that have nonzero probability.")