CS 109: Probability for Computer Scientists
Problem Set#5

Adonis Pugh

February 28, 2020

1. Let $X =$ the number of requests you receive at your web site per minute, where $X \sim \mathrm{Poi}(12)$. Each request, independently of all other requests, is equally likely to be routed to one of N web servers. Compute the expected number of web servers that will receive at least one request each during a minute. (Hint: there are a few ways to do this problem, but one way you might approach it is to first determine the *conditional* expectation of the number of web servers that receive at least one request each during a minute, conditioned on some fixed number, k, of requests during that minute. Then use that result to compute the *unconditional* expectation of the number of web servers that receive at least one request each during a minute.)

Answer. Let Y_i be an indicator variable that is 1 if the server gets one or more requests in a minute and 0 if it gets no requests in a minute. That is, let Y_i be a server. Let Y be the sum of all the Y_i random variables. We want to find the expected value of Y. We can do this using the law of total expectation to expand $E[Y]$ into calculable terms. This gives us an infinite sum of conditional expectations of Y given X is a fixed value multiplied by the probability that X takes on that value, which is a Poisson distribution. We can then compute our answer.

$$X \sim Poi(12) \quad Y = \sum_{i=1}^{N} Y_i$$

$$E[Y] = E[E[Y \mid X]] = \sum_{k=0}^{\infty} E[Y \mid X = k] P(X = k)$$

$$E[Y \mid X] = N \left(1 - \left(\frac{N-1}{N} \right)^k \right)$$

$$P(X = k) = \frac{12^k}{k!} e^{-12}$$

$$\boxed{E[Y] = \sum_{k=0}^{\infty} N \left(1 - \left(\frac{N-1}{N} \right)^k \right) \frac{12^k}{k!} e^{-12}}$$

2. You go on a camping trip with two friends who each have a mobile phone. Since you are out in the wilderness, mobile phone reception isn't very good. One friend's phone will independently drop calls with 10% probability. Your other friend's phone will independently drop calls with 25% probability. Say you need to make 6 phone calls, so you randomly choose one of the two phones and you will use that *same* phone to make all your calls (but you don't know which has a 10% versus 25% chance of dropping calls). Of the first 3 (out of 6) calls you make, one of them is dropped. What is the conditional expected number of dropped calls in the 6 total calls you make (conditioned on having already had one of the first three calls dropped)?

Answer. Since we know the number of calls dropped out of the first three is one, the number of dropped calls out of six must be at least one. Furthermore, each phone has a probability of being the phone that was chosen. This starts off as 50/50, but our knowledge of the dropped call means that the probability that it is either phone is no longer 50/50 and must be recalculated using Bayes' theorem given that one out of the first three calls were dropped. The answer is then the one known dropped call plus the expected number of calls out of the last three dropped by phone 1 multiplied by its new respective probability plus the expected number of calls out of the last three dropped by phone 2 multiplied by its new respective probability.

$X =$ calls dropped out of 6 $\quad Y = 1$ out of 3 calls dropped $\quad Z =$ calls dropped out of 3

$A =$ phone 1 with 10% drop probability $\quad B =$ phone 1 with 25% drop probability

$$E[X \mid Y] = 1 + E[Z \mid A]P(A) + E[Z \mid B]P(B)$$

$$P(A \mid Y) = \frac{P(Y \mid A)P(A)}{P(Y)} = \frac{P(Y \mid A)P(A)}{P(Y \mid A)P(A) + P(Y \mid B)P(B)}$$

$$P(Y \mid A) = \binom{3}{1}0.1^1(1-0.1)^{3-1} = 0.243$$

$$P(Y \mid B) = \binom{3}{1}0.25^1(1-0.25)^{3-1} \approx 0.4219$$

$$P(A \mid Y) = \frac{(0.243)(0.5)}{(0.243)(0.5) + (0.4219)(0.5)} \approx 0.36547$$

$$P(B \mid Y) = \frac{(0.4219)(0.5)}{(0.243)(0.5) + (0.4219)(0.5)} \approx 0.63453$$

$$E[X \mid Y] = 1 + 3 \times 0.1 \times 0.36547 + 3 \times 0.1 \times 0.63453 \approx 1 + 0.58554$$

$$\boxed{E[X \mid Y] \approx 1.5855}$$

3. You are developing medicine that sometimes has a desired effect, and sometimes does not. With FDA approval, you are allowed to test your medicine on 9 patients. You observe that 7 have the desired outcome. Your belief as to the probability of the medicine having an effect before running any experiments was $\text{Beta}(2, 2)$.

   a. What is the distribution for your belief of the probability of the medicine being effective after the trial?

   b. Use your distribution from (a) to calculate your confidence that the probability of the drug having an effect is greater than 0.5. You may use `scipy.stats` or an online calculator.

Answer.

a. To update our probability after a number of trials, we add the number of success and failures of the trials.

$$\text{Beta}(2+7, 2+2) = \boxed{\text{Beta}(9, 4)}$$

b.

$$P(X \geq 0.5) = 1 - P(X < 0.5) \approx 1 - 0.072998 \approx \boxed{0.927}$$

4. You are designing a randomized algorithm that delivers one of two new drugs to patients who come to your clinic—each patient can only receive one of the drugs. Initially you know nothing about the effectiveness of the two drugs. You are simultaneously trying to learn which drug is the best and, at the same time, cure the maximum number of people. To do so we will use the Thompson Sampling Algorithm.

   Thompson Sampling Algorithm: For *each* drug we maintain a Beta distribution to represent the drug's probability of being successful. Initially we assume that drug i has a probability of success: $\theta_i \sim \mathrm{Beta}(1, 1)$.

   When choosing which drug to give to the next patient we **sample** a value from each Beta and select the drug with the largest **sampled** value. We administer the drug, observe if the patient was cured, and update the Beta that represents our belief about the probability of the drug being successful. Repeat for the next patient.

   a. Say you try the first drug on 7 patients. It cures 5 patients and has no effect on 2. What is your belief about the drug's probability of success, θ_1? Your answer should be a Beta.

Method	**Description**
`V = sampleBeta(a, b)`	Returns a real number value in the range [0, 1] with probability defined by a PDF of a Beta with parameters a and b.
`R = giveDrug(i)`	Gives drug i to the next patient. Returns a True if the drug was successful in curing the patient or False if it was not. Throws an error if $i \notin \{1, 2\}$.
`I = argmax(list)`	Returns the index of the largest value in the list.

   b. Write pseudocode to administer either of the two drugs to 100 patients using Thompson's Sampling Algorithm. Use functions from the table above. Your code should execute `giveDrug` 100 times.

   c. After running Thompson Sampling Algorithm 100 times, you end up with the following Beta distributions:
   $\theta_1 \sim$ Beta$(11, 11)$,
   $\theta_2 \sim$ Beta$(76, 6)$.
   What is the expected probability of success for each drug?

Answer.

a. Since we had no prior belief, our distribution started as $\mathrm{Beta}(1, 1)$. However, once five cures were observed from 7 trials (5 success and 2 failures), our belief about the effectiveness of the drug is updated to be $\mathrm{Beta}(1 + 5, 1 + 2) = \mathrm{Beta}(6, 3)$.

b.

```
drug_A = [1, 1]
drug_B = [1, 1]
for i in range(0, 100):
    first = sampleBeta(drug_A[0], drug_A[1])
    second = sampleBeta(drugB[0], drug_B[1])
    choice = argmax([first, second])
    cured = giveDrug(choice + 1)
    if cured == true:
        if choice == 0:
            drug_A[0] += 1
        else:
            drug_B[0] += 1
    else:
        if choice == 0:
            drug_A[1] += 1
        else:
            drug_B[1] += 1
```

c.

$$\boxed{E[\theta_1] = \frac{11}{22} = 0.5}$$

$$\boxed{E[\theta_2] = \frac{76}{82} = 0.927}$$

5. A fair 6-sided die is repeatedly rolled until the total sum of all the rolls exceeds 300. Approximate (using the Central Limit Theorem) the probability that *at least* 80 rolls are necessary to reach a sum that exceeds 300.

Answer. The probability that at least 80 rolls are necessary to reach a sum greater than 300 is the same as the probability that the sum of 79 rolls is less than 300. In this case, we can use the Central Limit Theorem to sum 79 die rolls. We also have to use 300.5 rather than 300 for a continuity correction.

$$
\begin{aligned}
&Y = \text{sum of 79 rolls is less than 300} \\
&Y \sim \mathcal{N}(79(3.5), 79(3512)) = \mathcal{N}(276.5, 230.4167) \\
&P(Y \leq 300.5) = F(300.5) = \Phi\left(\frac{300.5 - 276.5}{\sqrt{230.4167}}\right) \approx \Phi(1.58) \approx \boxed{0.943}
\end{aligned}
$$

6. Program A will run 20 algorithms in sequence, with the running time for each algorithm being independent random variables with mean = 46 seconds and variance = 100 seconds^2. Program B will run 20 algorithms in sequence, with the running time for each algorithm being independent random variables with mean = 48 seconds and variance = 200 seconds^2.

   a. What is the approximate probability that Program A completes in less than 900 seconds?
   b. What is the approximate probability that Program B completes in less than 900 seconds?
   c. What is the approximate probability that Program A completes in less time than Program B?

Answer. These problems can all be solved by applying the Central Limit Theorem.

a.

$$A \sim \mathcal{N}(20(46), 20(100)) = \mathcal{N}(920, 2000)$$
$$P(A < 900) = F(900) = \Phi\left(\frac{900 - 920}{\sqrt{2000}}\right) \approx \Phi(-0.447) \approx 1 - \Phi(0.447) \approx 1 - 0.67364$$
$$\boxed{P(A < 900) \approx 0.326}$$

b.

$$A \sim \mathcal{N}(20(48), 20(200)) = \mathcal{N}(960, 4000)$$
$$P(B < 900) = F(900) = \Phi\left(\frac{900 - 960}{\sqrt{4000}}\right) \approx \Phi(-0.949) \approx 1 - \Phi(0.949) \approx 1 - 0.82894$$
$$\boxed{P(B < 900) \approx 0.171}$$

c.

$$P(T_A < T_B) = P(T_A - T_B < 0)$$
$$T_A - T_B \sim \mathcal{N}(920 - 960, 2000 + 4000) = \mathcal{N}(-40, 6000)$$
$$P(T_A < T_B) = F(0) = \Phi\left(\frac{0 - (-40)}{\sqrt{6000}}\right) \approx \Phi(0.5164)$$
$$\boxed{P(T_A < T_B) \approx 0.697}$$

7. **[Coding + Written]** Stanford's HCI class runs a massive online class that was taken by ten thousand students. The class used peer assessment to evaluate students' work. We are going to use their data to learn more about peer graders. In the class, each student has their work evaluated by 5 peers and every student is asked to evaluate 6 assignments: five peers and the "control assignment" (the graders were unaware of which assignment was the control). All 10,000 students evaluated the same control assignment, and the scores they gave are in the file `peerGrades.csv`. You may use simulations to solve any part of this question.

   Here are some rules that apply to all the coding questions:

   - Write your answers in the relevant functions of the file `cs109_pset5_hci.py`, which you can download from the course website. Do not rename this file. For this question, submit only this file.
   - Your code will be autograded.
   - Do not use global variables.
   - You may define helper functions if you wish.

   For written questions, write your answer in the PDF that you upload to Gradescope.

   a. **[Coding]** What is the sample mean of the 10,000 grades to the control assignment? Implement the `part_a` function, which should return this quantity as a float.

   For parts (b) and (c), you'll need to run some simulations. To get credit from the autograder, you're ***required*** to abide by the following guidelines:

   - Run the algorithm for exactly 10,000 iterations.
   - You'll need to draw random samples with replacement from an array of grades. To do so, you must use the `np.random.choice` function, which you can call like so:
     `sample = np.random.choice(`*name-of-array*`,` *size-of-random-sample*`, replace=True)`.
     Do not use any other function to generate random samples.
   - Use the `np.mean`, `np.median`, and `np.var` functions to calculate the mean, median, and variance of a list or numpy array.

   (b) **[Coding]** Students could be given a final score which is the *mean* of the 5 grades given by their peers. Imagine the control experiment had only received 5 peer-grades. What is the variance of the mean grade that the control experiment would have been given? Implement the `part_b` function, which should return this quantity as a float.

   (c) **[Coding]** Students could be given a final score which is the *median* of the 5 grades given by their peers. Suppose the control experiment had only received 5 peer-grades. What is the variance of the median grade that the control experiment would have been given? Implement the `part_c` function, which should return this quantity as a float.

   (d) **[Written]** Would you use the mean or the median of 5 peer grades to assign scores in the online version of Stanford's HCI class? Hint: it might help to visualize the scores. Feel free to write code to help you answer this question, but for this question we'll solely evaluate your written answer in the PDF that you upload to Gradescope.

Answer.

a. CODE

b. CODE

c. CODE

d. It would be more shrewd to use the median of the 5 peer grades to assign scores because the variance of the median is significantly less than the variance of the mean. In fact, when plotting histograms of the means and medians of 5 peer grades, the mean has significant outliers whereas the median is not sensitive to outliers.

8. **[Coding + Written]** In this problem you are going to learn how to use and misuse p-values for experiments that are called *A/B tests*. These experiments are ubiquitous. They are a staple of both scientific experiments and user interaction design.

   Suppose you are working at Coursera on new ways of teaching a concept in probability. You have two different learning activities `activity1` and `activity2` and you want to figure out which activity leads to better learning outcomes.

   Over a two-week period, you randomly assign each student to be given either `activity1` or `activity2`. You then evaluate each student's learning outcomes by asking them to solve a set of problems. The data (the activity shown to each student and their measured learning outcomes) are found in the file `learningOutcomes.csv`.

   For coding problems, write your answer in the relevant function of `cs109_pset5_coursera.py`. Follow the same coding guidelines as the previous coding problem (e.g. do not use global variables). For written problems, write your answer in the PDF that gets submitted to Gradescope.

   a. **[Coding]** What is the difference in sample means of learning outcomes between students who were given `activity1` and students who were given `activity2`? Write your answer in the `part_a` function, which should return a float (i.e. the difference in sample means).

   b. **[Coding]** Write code to estimate the p-value (using the bootstrap method) for the observed difference in means reported in part (a). In other words: assuming that the learning outcomes for students who had been given `activity1` and `activity2` were identically distributed, what is the probability that you could have sampled two groups of students such that you could have observed a difference of means as extreme, or more extreme, than the one calculated from your data in part (a)? Write your answer in the `part_b` function, which should return a float. Here are some guidelines to follow:

      - Just like in the previous problem, you are ***required*** to use the `np.random.choice` method with `replace=True` to generate random samples.
      - For the bootstrap algorithm, you should use 10,000 iterations, i.e. you should resample 10,000 times.
      - If you have two lists `a` and `b`, you can create a new list containing all the elements of `a` followed by all the elements of `b` by writing `a + b`

   Scientific journals have traditionally accepted an experiment's result as "statistically significant" if the p-value is below 0.05. By definition, this standard means that 5% of findings published in these journals are in fact not true, but just false positives. The scientific community is beginning to move away from using arbitrary p-value thresholds to determine whether a result is publishable. For example, see this 2019 editorial in the journal *Nature*: `https://www.nature.com/articles/d41586-019-00874-8`.

   You are now troubled by the p-value you obtained in part (b), so you decide to delve deeper. You investigate whether learning outcomes differed based on the background experience of students. The file `background.csv` stores the background of each student as one of three labels: more experience, average experience, less experience.

c. **[Written]** For each of the three backgrounds, calculate a difference in means in learning outcome between `activity1` and `activity2`, and the p-value of that difference.

You'll almost certainly need to write code in this question, and we've provided an `optional_function` that you can use, which gets called by our provided main method. However, we won't grade any code for this part. We'll only grade what you include in your answer PDF.

d. **[Written]** Your manager at Coursera is concerned that you have been "p-hacking," which is also known as data dredging: `https://en.wikipedia.org/wiki/Data_dredging`. In one sentence, explain why your results in part (c) are not the result of p-hacking.

Answer.

a. CODE

b. CODE

c. From a function coded using a similar implementations as in parts (a) and (b), the difference in means in learning outcome between activity 1 and activity 2 for less, average, and more experienced students, respectively, was computed to be approximately 26.02, -24.98 (absolute difference 24.98), and -28.42 (absolute difference 28.42). The p-values of the differences for less, average, and more, respectively, were 0.0003, 0.0073, and 0.0.

d. The results in part (c) are not the result of p-hacking because all sampled background experiences — less, average, and more — were considered in their entirety and reported as such and yielded extremely small p-values; the data was not cherry-picked to bias any result.