Jerry Cain
CS109

Spring 2023

CS109 Week 4 Exam

This is a closed calculator/computer exam. You are, however, permitted to consult the two double-sided sheets of notes you've prepared ahead of time. You're otherwise not permitted to refer to any other notes.

In the event of an incorrect answer, any explanation you provide of how you obtained your answer can potentially allow us to give you partial credit for a problem. It is fine for your answers to include summations, products, factorials, exponents, and combinations unless stated otherwise.

I acknowledge and accept the letter and spirit of the honor code.

Signature: ____

Last Name [print]: ____

First Name [print]: ____

SunetID [i.e., your @stanford.edu email]: ______________________________

SUID [i.e., your seven or eight-digit student ID number]: ______________________________

1 Summer Study in London [15 points]

Kathleen is planning to spend the summer at Imperial College London (on scholarship!) to partake in an intense three months of one-on-one tutorials with some of the world's leading experts in machine learning. Kathleen's scholarship allows for her to take a total of 7 tutorials, each of which meets just once per week for precisely one hour.

There are so many tutorials to choose from, though! Imperial offers a total of 25 different tutorials—five on Mondays, five on Tuesdays, five on Wednesday, five on Thursday, and five on Fridays. Fortunately, all tutorials are offered at different times, so Kathleen can choose any 7 of the 25 she wants without introducing any conflicts.

a. [2 points] How many different ways can Kathleen choose her seven tutorials if she can choose the seven without restrictions.

b. [5 points] How many different ways can Kathleen choose her seven tutorials so that she has at least one tutorial every weekday?

In order to be sure of your answer to part b, you've decided to count the same number a different way: by using the inclusion-exclusion method to count the number of ways Kathleen can have one or more days without any tutorials. Once you count that, you can subtract your answer here from your answer to part a to replicate the same number you derived for part b.

Let A_i be the set of all schedules that include at least one tutorial on the i^{th} weekday, where A_1 counts the number of schedules that include Monday tutorials, A_2 counts the number of schedules that include Tuesday tutorials, and so forth. To count the number of schedules that give Kathleen off one or more weekdays per week, we need to compute the following:

$$\left|A_1^C \cup A_2^C \cup A_3^C \cup A_4^C \cup A_5^C\right|$$

By applying the inclusion-exclusion principle to this particular problem, we arrive at:

$$\left|A_1^C \cup A_2^C \cup A_3^C \cup A_4^C \cup A_5^C\right| = \sum_i \left|A_i^C\right| - \sum_{i<j} \left|A_i^C \cap A_j^C\right| + \sum_{i<j<k} \left|A_i^C \cap A_j^C \cap A_k^C\right|$$

c. [2 points] Clearly but briefly explain why we don't need to include the four-way or five-way intersections of complements for this particular problem statement?

d. [6 points] Present an expression that counts the number of ways that Kathleen can have one or more weekdays off per week. Your expression that include terms and products that are consistent with the formula for $\left|A_1^C \cup A_2^C \cup A_3^C \cup A_4^C \cup A_5^C\right|$ above. Restated, you can't write down your answer to part a minus your answer to part b. Instead, your approach should present combinatorial expressions for $\left|A_1^C\right|$, $\left|A_1^C \cap A_2^C\right|$, and $\left|A_1^C \cap A_2^C \cap A_3^C\right|$ so you can use them to build up your final answer for the full union of all five complements.

2 Combinatorial Proofs [5 points]

Consider the following combinatorial identity for all integers $n \geq 2$:

$$\sum_{k=2}^{n} \binom{k}{2} \binom{n}{k}^2 = \binom{n}{2} \binom{2n-2}{n-2}$$

Present a **combinatorial** proof of the above identity, without relying on any algebra. As a hint, consider the selection of n members for a Stanford committee, where the n members are selected from a group of n professors and n students, where the co-chairs of the committee must both be professors.

3 The EuroMillions Lottery [15 points]

EuroMillions is a lottery where residents from participating European countries can play with hopes of winning the jackpot, which is generally tens (and often hundreds) of millions of euros. When purchasing a single lottery ticket, the player must

- choose five distinct integers between 1 and 50, inclusive, as your **main** numbers, and independently...
- choose two distinct integers between 1 and 12, inclusive (though the numbers chosen here may repeat one or two numbers chosen as part of the main five). These additional two numbers are known as the **Lucky Stars**.

Unsurprisingly, you win the jackpot (or at least share it in the event of multiple winners) when your five numbers match those drawn in the lottery and your two Lucky Stars match those drawn in the lottery as well. The order of the numbers are drawn doesn't matter.

Here are the results of EuroMillions from last Friday:

Note that last Friday's lottery just happened to draw an 8 two times. That's totally legitimate, since one of the 8's was drawn as part of the first five numbers and the second 8 was drawn separately as one of the two Lucky Stars.

a. [2 points] Assuming all size-5 subsets of $\{1, 2, \ldots, 50\}$ are equally likely to be drawn and all size-2 subsets of $\{1, 2, \ldots, 12\}$ are equally likely to be drawn (independently of the size-5 subsets), what is the probability that you match all seven numbers?

b. [4 points] What is the probability that you match precisely three of the five main numbers given that you match at least one?

c. [5 points] What is the probability that all seven numbers are different? Note that last Friday's drawing is legal, but because the 8 is repeated twice (once among the five and a second time as one of the two Lucky Stars), that particular outcome wouldn't contribute to the probability you're computing here.

d. [4 points] Present a pseudo-Python implementation of a program that estimates the expected number of **consecutive** lotteries needed before all 50 main numbers appear at least once **and** all 12 Lucky Star numbers appear once as well. Your implementation should compute this estimation by running the same experiment 100000 times, where each experiment simulates the lottery process as many times as needed until all numbers show up. Your implementation doesn't need to be all that efficient, but it needs to be correct. You needn't use `scipy` or `numpy` or anything fancy unless you really want to. We expect you to rely on simple Python data structures like Python lists and/or dictionaries and the `random.choice` function, as illustrated below.

Again, don't worry about syntax. We're perfectly happy with pseudo-code as long as there's a clear Python equivalent for each line of your implementation.

```
from random import choice
# choice(range(1, 13)) returns a random integer between 1 and 12 inclusive,
# all being equally likely
def estimate_expectation():
    num_lotteries_needed = 0
    for count in range(100000):
        num_lotteries_needed += run_one_simulation()
    return num_lotteries_needed/100000

def run_one_simulation():
    # place your implementation in the space below
```

4 Spam Detection [10 points]

In an attempt to reduce the amount of spam reaching your inbox, you've installed two separate anti-spam browser extensions. Any single email is either spam (S) or not (S^C), and each of the two programs either marks an email as spam (M_k) or legitimate (M_k^C), for $k = 1, 2$. Assume that 75% of all email is spam—i.e., $P(S) = \frac{3}{4}$, that the first browser extension correctly classifies as spam or legitimate with probability p_1–i.e., $P(M_1|S) = P(M_1^C|S^C) = p_1$, and that the second browser extension is accurate with probability p_2—i.e., $P(M_2|S) = P(M_2^C|S^C) = p_2$. For simplicity, assume the two extensions are conditionally independent of each other, regardless of whether the email is spam or not. Also assume that $0 < p_1 < p_2 < 1$, which among other things, says that the second browser extension correctly classifies emails more often than the first one does.

a. [2 points] Does the order in which the two browser extensions are applied matter? Restated, might our belief that an email is spam be different depending on whether or not the first extension is applied before the second versus the second being applied before the first? Explain your answer.

b. [4 points] What is the probability that a single email is spam even though the first browser extension identifies it as legitimate?

c. [4 points] What is the probability that a single email is spam when the two extensions disagree on whether or not the email is spam?