CSE 312: Foundations of Computing II Summer 2024

Homework 3

Due: Wednesday, July 10th, by 11:59pm

Instructions

See the instructions and FAQ for homeworks on the course website for important notes on the submission format!

Solutions submission. You must submit your solution via Gradescope. In particular:

- Submit a *single* PDF containing the solution to all of Tasks 1-6 to Gradescope under "**HW 3 [Written]**".
- Submit your code for Task 7 to "**HW 3 [Coding]**".
- Each numbered task should be solved on its own page (or pages). Follow the prompt on Gradescope to link tasks to your pages.
- Do not write your name on the individual pages – Gradescope will handle that.

Task 0 – Collaborators [0 pts]

List the full names of anyone you collaborated with on this homework. If you did not collaborate with anyone, write "None" in this section.

Task 1 – Lazy Weather Forecast [10 pts]

You're a weather forecaster with a very accurate but unreliable system. Each day, you will forecast one of 5 possible weather conditions: sunny, cloudy, rainy, snowy, or windy. With probability p, the system you use is working. If the system is working, you predict the weather forecast correctly with probability 0.9. If the system is broken, you randomly select one of the 5 conditions to announce.

Let C be the event that you forecast the correct weather condition today, and S be the event that the system is working today.

a) What is the probability that you announce the correct weather forecast?

b) What is the probability that the system was working, given your weather forecast was correct?

Task 2 – A Pond-erous Pursuit [10 pts]

Tamara is leading a field expedition to locate a rare species of salamander in a remote wetland area. This salamander species is found in approximately 3% of ponds in the area they are investigating. They have two tests (A and B) to help in detecting if the pond contains the salamander. We know the following:

- If the salamander is in the pond, the probability of test A being positive is 0.85 and the probability of test B being positive is 0.9.
- If the salamander is **not** in the pond, the probability of test A being positive is 0.2 and the probability of test B being positive is 0.25.

The events of tests A being positive and test B being positive are **conditionally independent** given the presence of the salamander and **conditionally independent** given the absence of the salamander.

a) What is the probability that test A is positive, test B is positive, **and** the salamander is in the pond?

b) What is the probability the salamander is in the pond given that both tests are positive?

Task 3 – Charlie in the Chocolate Factory [15 pts]

Charlie has two buckets of chocolate that Willy Wonka gifted him. Bucket A contains 14 Toblerone chocolates and 6 Lindor truffles. Bucket B contains 10 Toblerone chocolates and 10 Lindor truffles.
Charlie will perform the following experiment: he flips a fair coin. If the coin is heads, he goes to bowl A. If the coin is tails, he goes to bowl B. Then from whichever bowl he is standing in front of, he draws two chocolates independently with replacement (that is, he will put the first candy back before drawing the second). Let G_1 be the event that the first candy drawn is a Toblerone and G_2 be the event that the second candy drawn is a Lindor chocolate.

a) Calculate $\mathbb{P}(G_1)$. [4 points]

b) Calculate $\mathbb{P}(G_1 \cap G_2)$. [4 points]

c) Calculate $\mathbb{P}(G_1 \mid G_2)$. [4 points]

d) Based on your calculations so far, are G_1 and G_2 independent? Using your calculations and the definition of independence, justify your assertion. [1 point]

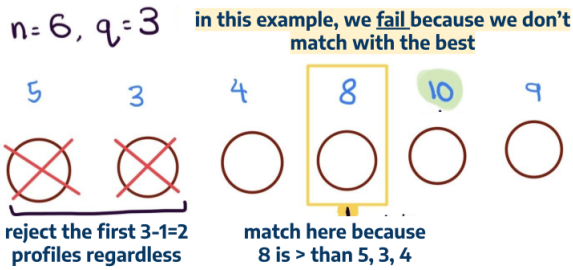
e) The result might be counter-intuitive. Explain the result **intuitively** (that is, you should not just refer directly to the numbers and the definition of independence; instead explain the result conceptually). [2 points]

Task 4 – Secret Admirers [20 pts]

You're using a dating app that shows you n profiles one at a time in random order. You can only view each profile once and can't look ahead or back. For each profile, you can either match (commit to that person and stop viewing other profiles) or pass (go to the next profile). The app guarantees a date with anyone you match with. Your goal is to maximize your chances of finding the best date among the n profiles, . In this problem, we want to analyze strategies for finding the best profile. (*You may assume that* $n \geqslant 1$.)

a) First, for a baseline, suppose your strategy were instead to match with the third profile no matter what. What is your probability of matching with your favorite among the n profiles? [3 points]

You don't know much about your potential dating pool, but you can immediately rank a new profile *relative to those you have already seen*. A probability expert tells you that the optimal strategy is as follows:
1. Reject the first $q-1$ profiles (regardless of how good you think they are) for some number q.
2. Starting with profile q, you will match with the first profile who is better than everyone you have seen so far.
In this problem, we'll analyze this strategy compute the best value of q.
(below is a picture of an example outcome of this strategy)



b) Let's start analyzing this strategy! For two natural numbers $q \leqslant i$, compute the probability that the best profile among the first $q-1$ is also the best profile among the first $i-1$ (so the $\max[1,i) = \max[1,q)$). You may assume $1 < q \leqslant i \leqslant n$. [5 points]

c) You match with the first profile at index q or later that is better than all the prior profiles you have seen. Supposing that the best profile is at index i, what is the probability that you will match with the best profile? Unlike in the previous part, for this part you will also need to handle the case that $i < q$; you may still assume that $1 < q$. (Hint: use part(b)!) [5 points]

d) We now set up a formula for the probability of selecting the best match if we ignore everyone before an arbitrary point q (i.e., we only start considering matching with someone if they are the q^{th} person we see or above). Use the Law of Total Probability to express the quantity as a summation over all possible placements of the best match. You will need to reason about the definition of our events to come up with the final result. Previous parts may be helpful here.

The final answer may still have a summation in it; simplify as far as you can, but don't expect a clean final answer. You also might need to have a separate formula for very small values of q or n (we have a special case when $q = 1$. If you have a separate case, you should explain where it comes from). [5 points]

To help you confirm if your answer is correct, when $n = 10$ and $q = 5$, the probability is approximately 0.3983, when $n = 10$ and $q = 4$ the probability approximately 0.3987. When $n = 100$, the best value of q is 38, and when $n = 1000$, the best value of q is 369.

Task 5 – Laundry... [10 pts]

The content in this question will be covered in lecture on Friday.
CSE 312 students sometimes delay laundry for a few days (to the chagrin of their roommates). Suppose a busy 312 student must complete 3 problem sets before doing laundry. Each problem set requires 1 day with probability $2/3$, and 2 days with probability $1/3$. Let B be the number of days a busy student delays laundry.

a) What is the range of B?

b) What is the PMF (probability mass function) for B?

c) What is the CDF (cumulative distribution function) for B?

d) What is $E[B]$?

Task 6 – Real-World: Bayes Theorem [10 pts]

The tools of this class are useful to computer scientists, but many of them are useful beyond just "classic" computer science. In this assignment you'll consider an application of Bayes' Rule in the real-world.

In this part, you'll use an application of Bayes Rule to make an argument about whatever real-world scenario you would like. Your scenario can be close-to-home (say something about an RSO you're involved in), a political issue, something computer science related, or anything else, as long as it's based in the "real-world"[1]. You are allowed (and encouraged!) to do your own research toward this question, but can also fall back on reasonable estimates.

The purpose of this question is for you to explore how conditional probability is used in the real world. As long as you put in effort to this questions, and show a valid application of Bayes' rule, you will get full credit. Have fun :)

a) Define events A and B on which you'll apply Bayes' Rule (along with any other events you need). [1 points]

b) State probabilities (or probability estimates) for three of the four quantities you need to use Bayes' rule and apply Bayes' rule. [1 point]

[1] We will be quite lenient about what counts as real world – the hope is that you will pick something you care about. If it's just the probability that the second and third card of a deck of cards are the same value, it's probably not "real-world." But if you're an avid poker player, and you want to use Bayes' Rule to analyze a particular game scenario, that would definitely count.

c) For those estimates, either cite a source for the numbers that you think is reliable or give a justification for your estimate. [1 point]

d) Apply Bayes' rule using the probabilities from part (b) [3 points]

e) What is your takeaway from this calculation? This needs to be more than just restating what your calculation in part b found. [2 points]

f) Discuss at least one limitation of your calculation/application (e.g. factors that didn't go into your estimates, or assumptions you are making that might not be correct). [2 points]

Some Ideas
We hope you'll think of something on your own! If you can't, here are some you might think about:

- We saw in class that routine medical tests can lead to false positives/negatives. Some tests you might consider looking into are over-the-counter pregnancy tests, colon cancer tests, and paternity tests. Here's a potential source of data you can use.
- How hard should Captchas and other "I'm not a robot" tests be to stop the robots from random guessing, but allow through fallible humans?
- How reliable is the rain prediction in weather apps for Seattle?

Sample Solutions
Sample solutions are posted on Ed so you have a sense of what to expect.

Task 7 – Naive Bayes [Coding] [25 pts]

Use the Naive Bayes Classifier to implement a spam filter that learns word spam probabilities from our pre-labeled training data and then predicts the label (ham or spam) of a set of emails that it hasn't seen before. **See this edstem lesson for an introduction to the Naive Bayes Classifier** and details on implementation, and also Section 9.3 from the book.

To solve the task, we have set up an edstem lesson. In particular, write your code to implement the functions `fit` and `predict` in the provided file, `cse312_pset3_nb.py`.

You will be able to run your code directly within edstem, and to test it, using the "Mark" option. This, however, will not evaluate your solution. Instead, once you're ready to submit, you can right-click the files in the directory to download them. Please upload your completed `cse312_pset3_nb.py` to Gradescope under "PSet3 [Coding]".

Some notes and advice:

- Read about how to avoid floating point underflow using the log-trick as described in these notes.
- Make sure you understand how Laplace smoothing works.
- Remember to remove any debug statements that you are printing to the output.
- **Do not directly manipulate file paths or use hardcoded file paths.** A file path you have hardcoded into your program that works on your computer won't work on the computer we use to test your program.
- Needless to say, you should practice what you've learned in other courses: document your program, use good variable names, keep your code clean and straightforward, etc. Include comments outlining what your program does and how. We will not spend time trying to decipher obscure, contorted code. Your score on Gradescope is your final score, as you have unlimited attempts. **START EARLY**.
- We will evaluate your code on data you don't have access to, in addition to the data you are given.

- Remember, it is not expected that Naive Bayes will classify every single test email correctly, but it should certainly do better than random chance! As this algorithm is deterministic, you should get a certain specific test accuracy around 90-95%, which we will be testing for to ensure your algorithm is correct. Note that we will run your code on a test dataset you haven't seen, but you will know immediately if you got full score.