Chris Piech
CS109

Fall 2022

CS109 Final Exam

This is a closed calculator/computer exam. You are, however, allowed to use notes in the exam.

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. For example, describe the distributions and parameter values you used, where appropriate. It is fine for your answers to include summations, products, factorials, exponentials, and combinations.

You can leave your answer in terms of Φ (the CDF of the standard normal) or Φ^{-1} (the inverse CDF). For example $\Phi\left(\frac{3}{4}\right)$ is an acceptable final answer. Recall that the exam is going to be "curved" according to the difficulty of the questions and as such hard questions will not translate to lower grades.

I acknowledge and accept the letter and spirit of the honor code. I pledge to write more neatly than I have in my entire life:

Signature: ______________________________

Family Name (print): ______________________________

Given Name (print): ______________________________

Email (preferably your gradescope email): ______________________________

1 Short Answer (22 points)

Answer each of the following questions. You must give a brief justification for your answer.

a. (5 points) What is the probability that a randomly chosen three-digit integer (from 0 to 999 inclusive) will be divisible by 5? Note that 0 is divisible by 5.

b. (9 points) Suppose X is a random variable that is normally distributed with a mean of 100 and a standard deviation of 15. What is the probability that a random sample of size 10 from this distribution will have a mean between 95 and 105?

c. (8 points) Each child in a daycare has a 0.2 probability of having disease A and has an independent 0.4 probability of having disease B. A child is sick if they have either disease A or disease B. If there are 10 children in a daycare what is the probability that 2 or more are sick?

2 Machine Learning (21 points)

a. (5 points) When implementing logistic regression, a student decides to add a second intercept value. To do so they add an extra feature with value 0 to each datapoint. How will this impact training?

b. (8 points) A Naive Bayes classifier is trained on a dataset with 100 examples, 30 of which are labeled as positive and 70 of which are labeled as negative. Instead of using a Laplace prior, you use a Beta($a = 3$, $b = 4$) prior. What is your estimate for the probability $Y = 1$?

c. (8 points) Sometimes, we would like to incorporate additional terms that represent interactions between different features into a logistic regression model. Imagine a dataset with two features, x_1 and x_2, and a corresponding label y for each datapoint. We will add the second-order feature x_1x_2, so that our model is $P(Y = y|X = x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 \cdot x_1 x_2)$. Explain in 1 or 2 sentences how you would change your logistic regression code in order to train this model.

3 Night Sight (20 points)

In this problem we explore how to use probability theory to take photos in the dark. Digital cameras have a sensor that capture photons over the duration of a photo shot to produce pictures. However, these sensors are subject to “shot noise” which are random fluctuations in the amount of photons that hit the lens. In the scope of this problem, we **only consider a single pixel**. The arrival of shot noise photons on a surface is independent with constant rate.

a. (6 points) Shot noise photons land on a particular pixel at a rate of 10 photons per microsecond (μs). If the time duration of a photo shot is 1000 μs, what is the variance of the amount of photons captured by the pixel during a single photo?

b. (14 points) To mitigate shot noise, Stanford graduates realized that you can take a shutter shot (many camera shots in quick succession) and average the number of photons captured. The largest number of photos a camera can take in 1000μs is 15 photos, each with a duration of 66μs. Let X be the average quantity of shot noise photons across the 15 photos, captured by the single pixel. What is $\text{Var}(X)$?

4 Penalty Shootout (20 points)

Soccer games may end up with a penalty shootout. Use probability to estimate the probability that a particular team will win. In a penalty shoot out each team takes 5 shots. If after 5 shots both teams have the same number of goals, they repeat taking one more shot each, until one team has more goals.

Assume that: Players on team A have a 0.8 probability of scoring on each penalty shot. Players on team B have a 0.7 probability of scoring on each penalty shot. Assume that each shot is independent. What is the exact probability that team A wins?

5 Better (20 points)

Write a function `better` which returns the approximate probability that video A has a higher probability of being liked than video B, based on historical observations. Each video has two values: `likes` and `not_likes`. Model the probability that a viewer likes a movie as a random variable and use a Laplace prior for the random variable. You may use sampling if it is helpful (use on the order of 1 million samples).

```
def better(a_likes, a_not_likes, b_likes, b_not_likes):
    num_samples = 1000000

    count_where_aprob_gt_bprob = 0
    for samp in (range(num_samples)):
        # params for beta are (num_successes + 2) and (num_fails +2), with LaPlace prior
        a_sample_prob = scipy.stats.beta.rvs(a_likes + 2, a_not_likes + 2)
        b_sample_prob = scipy.stats.beta.rvs(b_likes + 2, b_not_likes + 2)
        if a_sample_prob > b_sample_prob:
            count_where_aprob_gt_bprob += 1

    return count_where_aprob_gt_bprob / num_samples
```

6 B-Reel (35 points)

A social media application "B-Reel" promises to send users a notification exactly once each day "randomly" in a 10 hour period. You want to test if the time that notifications come in are truly uniform. You have recorded 100 IID historical values: $[x_1, x_2, \ldots, x_{100}]$ where $x_i \in [0, 10]$ is the time the notification came in for the ith day, measured in hours from the start of the time period.

a. (5 points) Calculate the likelihood of the dataset given each value is IID from a uniform(0, 10). In otherwords, the density of each value.

b. (10 points) You have an alternative hypothesis: there is a 0.4 probability a notification comes in the first half of the day, and a 0.6 probability that a notification comes in the second half of the day (and that the time is uniform within the halves). The historical data has 45 notifications in the first half of the day and 55 in the second half. What is the probability density of these 100 samples, given this alternative hypothesis?

c. (8 points) Let p_a and p_b be your answers to part a and b respectively. What is the probability of your alternative hypothesis? Assume that the samples must come from either the uniform or the alternative hypothesis. Your prior belief that B-Reel is using a uniform(0,10) is 0.6.

d. (12 points) Perhaps we don't have enough data. Use bootstrapping to estimate the variance of the probability calculated, if you were to repeat this experiment 10,000 times. Let `data` be the list of historical values. You can use the function `var(list)` to estimate the sample variance from a list of values.

```
import numpy as np

def solution(data):
    n = len(data)
    probs = []
    for i in range(10000):
        resampled = np.random.choice(data, size=n, replace=True)

        # calculate likelihood of samples using method in part b.
        count_morning = sum([1 if num < 5 for num in resampled])
        count_night = n - count_morning
        prob_a = 0.1**100
        prob_b = ((0.4 * 0.2) ** count_morning) * ((0.6 * 0.2) ** count_night)
        p_alternative = (prob_b * 0.4) / (prob_b * 0.4 + prob_a * 0.6)
        probs.append(p_alternative)
    return var(probs)
```

7 Code survival (20 points)

The Gopertz distribution can be used to model how long a piece of code will remain in production. It is defined by parameter a and has probability density function:

$$f(X = x) = 2a \cdot e^{3a-2a\cdot e^{2x}}$$

We wish to model how long a particular code will last at a given company. To this end we collect N independent measurements of how long code lasts in production: $x_1, x_2, \ldots, x_N$. Explain, in words, how you would choose parameter a using the maximum likelihood estimation framework, and provide any necessary derivatives.

8 Approximate Counting Algorithm (22 points)

What if you wanted a counter that could count up to the number of atoms in the universe, but you wanted to store the counter in 8 bits? You could use the algorithm below. Show that the expected return value of `stochastic_counter(4)`, where `count` is called four times, is in fact equal to four.

```
def stochastic_counter(true_count):
   n = -1
   for i in range(true_count):
       n += count(n)
   return 2 ** n  # 2ˆn, aka 2 to the power of n

def count(n):
   # To return 1 you need n heads. Always returns 1 if n is <= 0
   for i in range(n):
       if not coin_flip():
           return 0
   return 1

def coin_flip():
   # returns true 50% of the time
   return random.random() < 0.5
```

9 Ranking (1 point)

Bonus point: rank the questions (1 - 8) from this exam in order of how confident you feel in the correctness of your answer. For example, writing $3 \leq 1 \leq 5$ would mean you are most confident in your answer for question 5 and least confident in your answer for question 3.

Please rank whole questions (1 - 8) and not subparts.

$$\underline{\qquad} \leq \underline{\qquad} \leq \underline{\qquad} \leq \underline{\qquad} \leq \underline{\qquad} \leq \underline{\qquad} \leq \underline{\qquad} \leq \underline{\qquad}$$

That's all folks. Thank you all for the wonderful quarter and we hope you have a fantastic winter break. Night Sight is a real algorithm invented by Stanford CS folks and is now in production in Google pixel (and presumably other places too). Approximate counting is a real randomized algorithm which can count in log log space. Better is a more sophisticated way to rank videos than sorting by average likes.