Chris Piech
CS109

Win 2025

CS109 Final Exam

This is a closed calculator/computer/phone/smart-watch/smart-toothbrush exam. You are, however, allowed to use notes in the exam. You have 3 hours (180 minutes) to take the exam. The exam is 180 points, meant to roughly correspond to one point per minute of the exam. You may want to use the point allocation for each problem as an indicator for pacing yourself on 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}. For example $\Phi(3/4)$ is an acceptable final answer.

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 [23 points]

a. (5 points) Let $X \sim \text{Exp}(\lambda = 0.2)$. What is the probability that X is between 5 and 10?

b. (6 points) When it rains in the morning, your neighbor's dog has a 70% chance of being mucky. On mornings without rain, their dog is mucky only 20% of the time. Each day has a 10% chance of rain in the morning. Today, their dog was mucky. What is the probability it was raining in the morning?

c. (6 points) A standard deck of cards has 12 royals and 40 non-royal cards. If the top card is a royal you will have a 80% chance of winning. If the top card is a non-royal card, you have a 40% chance of winning. What is your probability of winning the game?

d. (6 points) For the CS109 Midterm, there are 300 students who are sent to one of three rooms based on their last name. Each student, independently, has a 30% chance of going to room 1, a 20% chance of going to room 2 and a 50% chance of going to room 3. What is the chance that exactly 90 students go to room 1, 60 students go to room 2, and 150 go to room 3?

2. In-Vitro Fertilization [13 points]

A goal of an In-Vitro Fertilization (IVF) cycle is to produce as many healthy embryos as possible. At the start of a cycle, a patient has a certain number of growing antran follicles. Each antran follicle must successfully pass through three stages to become a healthy embryo:

- Maturity: An antran follicle has a **0.9** chance of reaching full maturity.
- Retrieval: A fully mature follicle has a **0.8** chance of successfully being retrieved.
- Fertilization: A retrieved follicle has a **0.7** probability of being successfully fertilized into a healthy embryo.

Assume each follicle's journey is independent.

a. (6 points) What is the probability that a single antran follicle will become a healthy embryo?

b. (7 points) A particular patient has 14 antran follicles at the start of a cycle. Let your answer to part (a) be p_a. What is the probability that the patient will have k healthy embryos (for $k = 0, 1, \ldots, 14$)?

3. Bag of Coins [20 points]

A bag has 10 coins that *appear* identical. However, some coins are of type A, some coins are of type B.

- A coin of type A has probability **0.3** of landing Heads.
- A coin of type B has probability **0.6** of landing Heads.

All coins are independent of one another. All 10 coins are flipped, and we count the number of heads.

a. (7 points) Imagine 7 of the coins were type A, and 3 of the coins were type B. What is the probability that we get exactly two heads from the type A coins and exactly two heads from the type B coins?

b. (7 points) Under the same assumption (7 coins of type A, 3 of type B), what is the probability of observing 4 heads total?

c. (6 points) Let X be a discrete random variable for the number of type A coins in the bag (from 0 to 10). Your prior belief is $P(X = x) = \frac{1}{11}$. Let H be the total number of heads.

Show that $P(X = x|H = 4) = k \cdot P(H = 4|X = x)$ where k is a constant. You do not need to solve for k.

4. Microcontroller Precision Error [15 points]

A low-powered microcontroller can only represent numbers to 3 decimal places.

Let X_i be an i.i.d. sample from $\text{Uniform}(0, 1)$ with infinite precision. Let Y_i be the representation of X_i on the microcontroller. Y_i is equal to X_i truncated to 3 decimal places. Here are three examples:

X_i	Y_i
0.12340809...	0.123
0.28374110...	0.283
0.55555555...	0.555

a. (5 points) What is the distribution of $(X_i - Y_i)$? In other words, what is the distribution for the error when a single uniformly sampled value is represented in the microcontroller?

b. (10 points) We sample 1000 values, where X_i is the precise value and Y_i is the truncated value of X_i:

$$X = \sum_{i=1}^{1000} X_i \qquad Y = \sum_{i=1}^{1000} Y_i$$

What is the probability that the difference between X and Y is more than 0.51?

5. Exponential Backoff [18 points]

Two computers simultaneously attempt to send messages over a shared wire channel at second zero, causing a first collision. The messages were not delivered and they will have to re-attempt.

Each computer is using an algorithm called Exponential Backoff. If they have had n collisions, they independently wait for a random integer number of seconds uniformly chosen from the set $\{1, \ldots 2^n\}$ before attempting to retransmit over the shared wire:

Number of collisions, n	Possible wait times (seconds)
1	$\{1, 2\}$
2	$\{1, 2, 3, 4\}$
3	$\{1, 2, 3, 4, 5, 6, 7, 8\}$
4	$\{1, 2, 3, 4, 5, 6, 7, 8, \ldots 16\}$
...	...

If the computers choose the same number of seconds to wait, they will have another collision. This process continues until there is no collision.

For example, after their initial collision, $n = 1$, so they will both randomly choose to wait 1 or 2 seconds. If they both choose 1, or if they both choose 2, there will be a second collision. Otherwise the messages will successfully send.

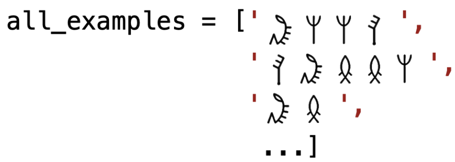
a. (8 points) What is the probability of having **exactly 3 collisions total**? (Equivalently, two more collisions after the initial one at second 0, followed by a successful send.)

b. (10 points) The computers have exactly 3 collisions and then successfully send on the next attempt. What is the expected amount of time it took the two computers to successfully send both their messages? *Note: The only time you have to consider is the time that the computers are waiting during exponential backoff, not the time it takes a message to pass over the wire, nor the time it takes to recognize a collision.*

6. Indus Valley Script [19 points]

The Indus civilization - one of the world's earliest urban societies - emerged 5,300 years ago. The script they used is one of the last remaining undeciphered alphabets.

We are using entropy analysis to show that the script represents a true linguistic system. Let **all_examples** be the list of all recorded examples of the script, where each item in the list is one example string:



a. (5 points) An AI chatbot answered a question on how to compute the entropy of the first character in a string from the Indus Valley Script. Find any mistakes.

> **Chatbot response of the entropy of the first character:**
>
> Let $P(X = x)$ be the probability that the first character in a script string is x. Let n be the total number of strings in **all_examples**, and let n_x be the number of times that character x is the first character in the example.
>
> $$\text{Entropy}(X) = \sum_x \log_2 \frac{1}{P(X = x)}$$
>
> Next, we apply a log transformation rule to simplify the expression:
>
> $$\text{Entropy}(X) = -\sum_x \log_2 P(X = x) \quad \text{(Log rule)}$$
>
> We then substitute $P(X = x) = \frac{n_x}{n}$ to reflect the frequency across all examples:
>
> $$\text{Entropy}(X) = -\sum_x \log_2 \frac{n_x}{n} \quad (P(X = x) = n_x/n)$$
>
> Finally, we refactor out the log:
>
> $$\text{Entropy}(X) = -\log_2 \sum_x \frac{n_x}{n} \quad \text{(Sum of logs)}$$

b. (7 points) Write a function **pr_y_given_x(all_examples, x, y)** which returns the conditional probability that the second character in the string is **y**, given that the first character in the string is **x**.

```python
def pr_y_given_x(all_examples, x, y):
    count_x = 0
```

```
    count_x_and_y = 0
    for example in all_examples:
        if example[0] == x:
            count_x += 1
            if example[1] == y:
                count_x_and_y += 1

    return count_x_and_y / count_x
```

c. (7 points) On average over all examples of the script, how much does the entropy of the second character decrease once you have observed the first character? Let $P(Y = y|X = x)$ be your answer to part b. Let $P(X = x)$ be the probability that the first character is x. Your answer can be a math expression or code.

Solution (code):

```
def entropy(pmf):
    H = 0
    for x in pmf:
        H -= pmf[x] * log2(pmf[x])
    return H

def solution():
    p_y = {}
    for x in p_x:
        for y in p_y_given_x[x]:
            p_y[y] += p_y_given_x[x][y] * p_x[x]

    H_y = entropy(p_y)
    E_H_y_given_x = 0
    for x in p_x:
        H_y_given_x = entropy(p_y_given_x[x])
        E_H_y_given_x += H_y_given_x * p_x[x]

    return H_y - E_H_y_given_x

# If you iterate over the set of examples, we get P(x) and P(x,y)
# for free, so we sum the surprise instead of entropy inside the loops.
def alternate_solution():
    p_y = {}
    for x in p_x:
        for y in p_y_given_x[x]:
            p_y[y] += p_y_given_x[x][y] * p_x[x]

    H_y = 0
    E_H_y_given_x = 0
    for example in all_examples:
        x = example[0]
        y = example[1]
```

```
        H_y -= log2(p_y[y])
        E_H_y_given_x -= log2(p_y_given_x[x][y])

    H_y /= len(all_examples)
    E_H_y_given_x /= len(all_examples)

    return H_y - E_H_y_given_x
```

7. Estimating Course Size [19 points]

In order to hire the correct number of TAs, Stanford needs to estimate final enrollment in CS109 two weeks before the start of the quarter. Two weeks before the start of the quarter 300 students are enrolled.

For the last 10 offerings of CS109 Stanford has recorded the ratio:

$$r_i = \frac{\text{(final enrollment for offering } i\text{)}}{\text{(enrollment two weeks before start for offering } i\text{)}}$$

which you can access as a list of values $[r_1, r_2, \ldots, r_{10}]$. Assume that each r_i is an i.i.d. sample from the true ratio random variable R. The number of students who end up taking the class this quarter, T, will be $T = 300 \cdot R$. From historical analysis we know that both R and T are normally distributed.

a. (5 points) Estimate $E[T]$, the expected class size. Provide your answer as a math expression.

b. (6 points) Estimate $\text{Var}(T)$. Provide your answer as a math expression.

c. (8 points) Use *bootstrapping* to estimate the *standard deviation* of your estimate of $\text{Var}(T)$ from part (b) (i.e., the sampling variability due to having only 10 historical observations of R).

8. Bayesian RNA Sequencing [18 points]

Let N_A and N_B be the count of type-A RNA and type-B RNA in a person's body. Both are Poisson distributed:

$$N_A \sim \text{Poi}(12 \cdot X) \qquad N_B \sim \text{Poi}(12 \cdot (1 - X))$$

N_A and N_B have rates that come from an unknown continuous variable X. X takes on values between 0 and 1 and is an indicator of a person's overall health. N_A and N_B are **independent**, conditioned on knowing X.

a. (5 points) For part (a) only, assume $X = 1/4$. What is $P(N_A > 10)$?

b. (10 points) Our prior belief is that $X \sim \text{Beta}(a = 2, b = 2)$. An experiment observes $N_A = 12$ and $N_B = 16$. Based on these observations, what is your updated distribution for X? In other words, what is $f(X = x | N_A = 12, N_B = 16)$? You can leave your answer in terms of a normalization constant.

c. (3 points) What is the variance of your updated belief distribution for X, found in part (b)?

9. MLE of a Truncated Normal Distribution [18 points]

A normal distribution can allow negative values. A truncated normal is a version of the normal distribution that only supports positive values of x. If $X \sim \text{TruncatedNormal}(\mu)$ then it has the following PDF:

$$f(X = x) = \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2}}}{\Phi(\mu)} \quad \text{for } x > 0.$$

Where Φ is the Cumulative Distribution Function for the Standard Normal.

You have n i.i.d. samples from a TruncatedNormal: [7.923, 12.388, 9.984, 12.019, 8.725, . . .]. Let x_i be the ith value. Explain how you would use MLE to estimate parameter μ and provide any necessary derivatives.

10. Logistic Regression Conceptual [17 points]

a. (4 points) In training a logistic regression model, why do we typically optimize the *log-likelihood* directly, rather than the likelihood?

b. (4 points) Why is logistic regression a more principled model than linear regression for binary classification?

c. (4 points) Consider a logistic regression that predicts Y from features $X_1, \ldots, X_m$. If Y and a particular feature X_i are independent, what would you expect for the learned coefficient θ_i, and why?

d. (5 points) Below is pseudocode for training for a logistic regression model:

```
1   thetas[j] = 0 for all j
2   repeat many times:
        gradients[j] = 0 for all j
3       for each training example (x, y):
            for each parameter j:
4               gradients[j] += x[j]*(y - sigmoid(dot_prod(theta, x))
5       thetas[j] += step_size * gradients[j] for all j
```

For each line of pseudocode 1-5, explain briefly, as if teaching, what the line of code does, and why it is necessary.

That's all folks! Thank you for the lovely quarter. You were a wonderful class. IVF numbers are mostly real, but I removed a few steps. Exponential Backoff is the true algorithm used to handle congestion for everything from radio transmissions to the internet. In 2025 the chief minister of southern India's Tamil Nadu state announced a $1m prize for anyone who can crack the Indus Valley Script. Estimating Course Sizes is (very close to) the real algorithm the CS department uses. Bayesian RNA quantification is a twist on a widely used algorithm called cufflinks for estimating RNA isoform proportions. Truncated Normals are super useful! They have a variance parameter, a min value and a max value which I removed for exam simplicity.