Juliette Woodrow
CS109

Winter 2026
March 17th

CS109 Final Exam

This is a closed calculator/computer/phone/smart-watch/smart-toothbrush exam. You are, however, allowed to use **6 pages** of 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 $\Phi(\frac{3}{4})$ is an acceptable final answer. You can only leave your answer in **terms of code** when the problem explicitly says so.

Stanford Honor Code: The Honor Code is an undertaking of the Stanford academic community, individually and collectively. Its purpose is to uphold a culture of academic honesty. Students will support this culture of academic honesty by neither giving nor accepting unpermitted academic aid on this examination.

This course is participating in the proctoring pilot overseen by the Academic Integrity Working Group (AIWG), therefore proctors will be present in the exam room. The purpose of this pilot is to determine the efficacy of proctoring and develop effective practices for proctoring in-person exams at Stanford.

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: _____

First and Last Name (print): _____

Stanford Email (@stanford.edu): _____

Exam Break Sign-out

I pledge that during my exam break:

- I will not bring any paper, electronic devices (phone, smart watch, smart glasses, etc), or aid (permitted or unpermitted) *out of or into* the exam room, nor access any aid during the break.
- I will not communicate with anyone other than the course instructional staff about the content of the exam.

Signature Confirming Honor Code	**Exit Time**	**Return Time**	**Proctor Initial**	**Length (mins)**

If you are feeling unwell and are not able to complete the exam, please speak with the proctor.

1. Let's Get This Party Started [18 Points]

You may not use code to answer any questions in this section.

a. (4 points) It is an empirical curiosity of Wikipedia that repeatedly clicking the first hyperlink of any article will almost always lead to the article **Philosophy**. We model each click as independently reaching Philosophy with probability p. You start at a random article on Wikipedia that isn't Philosophy. What is the probability of reaching the Philosophy article for the first time in exactly 6 clicks?

Let X represent the number of clicks until we hit the first link to the Philosophy article. Because we assume that each click is independent and has the same probability p of reaching the Philosophy article, we can model this problem using a Geometric distribution.

$$X \sim \text{Geo}(p)$$
$$P(X = 6) = (1-p)^5 p$$

We can similarly arrive at the same solution using principles of core probability. In order to arrive at Philosophy article in exactly 6 clicks, we need exactly 5 non-Philosopher clicks and the final click must lead to the Philosophy article. Let E represent the event that we arrive to the Philosophy article in exactly 6 clicks. p represents the probability that one click reaches the Philosophy article, and $1-p$ represents the probability of reaching an article that is not Philosophy.

$$\begin{aligned} P(E) &= (1-p) \cdot (1-p) \cdot (1-p) \cdot (1-p) \cdot (1-p) \cdot p \\ &= (1-p)^5 p \end{aligned}$$

b. (4 Points) You've observed that the length of time, in minutes, between birds flying by your window is distributed as $\text{Exp}(\lambda = 1/5)$. What is the probability that exactly 3 birds fly by your window in the next 7 minutes?

In section, we saw that we could use both a Poisson or Exponential variable to answer questions about an event, E, taking longer than some amount of time to occur (e.g. the probability of a single download taking longer than 10 minutes).
This exam problem, however, is an example of a question that can only be solved using a Poisson. If you wanted to solve this problem using a single Exponential variable, you'd need to reason about the joint behavior of multiple inter-arrival times simultaneously.
We are told that the time interval (i.e. units) of the Exponential variable is 1 minute. Thus, we can use λ to derive a rate of $\frac{1}{5}$ of bird per 1 minute (or 1 bird in 5 minutes).
Because we are interested in the number of birds observed in the next 7 times, our Poisson distribution will have a lambda rate that expresses the average number of birds seen in 7 minutes. Let Y represent the number of birds in 7 minutes. Our $\lambda = 7 \cdot \frac{1}{5} = \frac{7}{5} = 1.4$.

$$Y \sim \text{Poi}(\lambda = \frac{7}{5})$$
$$P(Y = 3) = \frac{\frac{7}{5}^3 e^{-\frac{7}{5}}}{3!}$$

c. (5 Points) A researcher is trying to test whether experiencing a headache is independent of having consumed caffeine the previous day. Let H be the event that a person has a headache on a given day, and let C be the event that the person consumed caffeine the previous day.

Let $P(H)$ be the overall probability that a person reports a headache, and let $P(H \mid C^C)$ be the probability that a person reports a headache given that they did *not* consume caffeine the previous day.

To investigate if headaches are independent of caffeine consumption the previous day, the researcher decides to test whether $P(H) \approx P(H \mid C^C)$. Explain whether the researcher's approach is correct using 1-2 sentences.

The researcher's approach is correct. If two events are independent, then the probability of one event occurring does not change based on whether the other event occurred (or didn't occur). In this case, the researcher is testing whether having a headache (A) is independent of caffeine consumption the previous day (C).
By definition of independence:

$$P(A) = P(A \mid C^C) = P(A|C)$$

So testing whether $P(A) \approx P(A \mid C^C)$ is a valid way to assess independence.

d. (5 Points) In a cancer genomics study, researchers are analyzing mutations in two important genes, TP53 and BRAF, across 1,000 tumor samples. Each tumor is tested for mutations in both genes.

The results are summarized in the table below:

	BRAF Mutated	**BRAF Not Mutated**
TP53 Mutated	0	300
TP53 Not Mutated	200	500

Are the events "TP53 Mutated" and "BRAF Mutated" independent? Explain in 1-2 sentences your reasoning.

No. They are mutually exclusive. The table shows that no tumor sample had mutations in both TP53 and BRAF: the top-left cell is 0. So, these events are mutually exclusive because they never occur together. Mutually exclusive events are dependent events (if one event happens, the other event cannot happen), which violates the definition of independence.

2. Powering a Town [18 points]

An energy company is deciding how much power to supply to a small town of $n = 2500$ households during peak hours (6–9pm). Each household's electricity consumption is independent and identically distributed. You are told that the average consumption per household during this time is 2.4 kW and the variance is 9 kW^2.

25% of households have an electric vehicle. Some households are classified as "surge-risks". Among EV households, 60% are surge-risk. Among non-EV households, 10% are surge-risk. You may not use code to answer any part of this problem.

a. (3 points) For a randomly selected house, what is the exact probability it is a surge-risk?

Let S be the event where a household is surge-risk, and let E be the event where a household has an EV. Then, we can calculate:

$$\begin{aligned} P(S) &= P(S|E)P(E) + P(S|E^C)P(E^C) \\ &= 0.6(0.25) + 0.1(0.75) \\ &= 0.225 \end{aligned}$$

b. (3 points) A randomly selected household is surge-risk. What is the probability that it has an EV?

We use Bayes' rule:

$$P(E \mid S) = \frac{P(S \mid E)P(E)}{P(S)}$$

From part (a),

$$P(S) = 0.225$$

Now we compute:

$$P(E \mid S) = \frac{0.6 \cdot 0.25}{0.225} = \frac{0.15}{0.225} = \frac{2}{3}$$

c. (6 points) What is the probability that total power consumed by all the houses in the small town during this time period exceeds 6300 kW?

The power consumed by the ith household (in kW) is given by $X_i \sim \mathcal{N}(2.4, 9)$. Since X_i are independent, we can sum the Normals. The total power consumption is

$$X = \sum_{i=1}^{2500} X_i \sim \mathcal{N}(2500 \cdot 2.4, 2500 \cdot 9).$$

Then, the desired probability is

$$\begin{aligned} P(X > 6300) &= 1 - \Phi\left(\frac{6300 - 2500 \cdot 2.4}{\sqrt{2500 \cdot 9}}\right) \\ &= 1 - \Phi\left(\frac{300}{150}\right) \\ &= 1 - \Phi(2). \end{aligned}$$

d. (6 points) We define demand for the town as the total power consumed by all the houses during this time. The company wants to provide enough power during this time such that the probability of demand exceeding supply is at most 1%. Find the minimum supply capacity T (in kW) such that the probability that the demand is greater than the supply is ≤ 0.01.

Let $Y \sim \mathcal{N}(2500 * 2.4, 2500 * 9)$ be the total power consumed by all of the houses during this time, as in part (c).

We want to find T such that $P(Y > T) = 0.01$. This is equal to $1 - P(Y < T) = 0.01$. We set this up in the following way:

$$\begin{aligned} 1 - \Phi\left(\frac{T - (2500 \cdot 2.4)}{\sqrt{2500 \cdot 9}}\right) &= 0.01 \\ 0.99 &= \Phi\left(\frac{T - (2500 \cdot 2.4)}{\sqrt{2500 \cdot 9}}\right) \\ \Phi^{-1}(0.99) &= \frac{T - (2500 \cdot 2.4)}{\sqrt{2500 \cdot 9}} \\ \sqrt{2500 \cdot 9} \cdot \Phi^{-1}(0.99) + (2500 \cdot 2.4) &= T \\ 150 \cdot \Phi^{-1}(0.99) + 6000 &= T \end{aligned}$$

3. Let's Tree Max [15 Points]

A conservation group is tracking how trees regrow after a fire. They divide a large forest plot into $k = 400$ equal-sized **zones**. From a nearby grove, $n = 1200$ seeds are blown by the wind into this plot. Because the wind is turbulent, each seed is independently and equally likely to land in any of the k zones. You may not use code to answer any part of this problem.

a. (4 Points) What is the probability that zone i receives exactly 15 seeds?

Let X represent the number of seeds that land in zone i. Because each seed landing is independent, this problem can be modeled as a Binomial:

$$X \sim \text{Bin}(n = 1200, p = 1/400)$$

$$P(X = 15) = \binom{1200}{15}\left(\frac{1}{400}\right)^{15}\left(1 - \frac{1}{400}\right)^{1200-15}$$

Alternatively, since we have small p and large n, a Poisson approximation is acceptable:

$$X \sim \text{Poi}(\lambda = 3)$$

$$P(X = 15) = \frac{3^{15}e^{-3}}{15!}$$

b. (5 Points) A zone is considered "crowded" if it receives **5 or more** seeds. Using a Poisson approximation, compute the probability that Zone i is "Crowded."

Let X represent the number of seeds that land in Zone i, as above. We know that $X \sim Bin(n = 1200, p = 1/400)$ where n is the total number of seeds and p is the probability of landing in Zone i.

Because n is very large and p is very small, we can approximate X as a Poisson distribution. We calculate λ (avg. number of seeds per zone) as $\lambda = np = 1200 * \frac{1}{400} = 3$. Therefore, $X \sim Poisson(\lambda = 3)$.

A zone is "crowded" if it receives 5 or more seeds, so we want to find $P(X \geq 5)$. Using the complement rule and the Poisson PMF $P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$, we proceed as follows:

$$\begin{aligned} P(X \geq 5) &= 1 - P(X \leq 4) \\ &= 1 - \sum_{x=0}^{4} P(X = x) \\ &= 1 - \left(\frac{3^0 e^{-3}}{0!} + \frac{3^1 e^{-3}}{1!} + \frac{3^2 e^{-3}}{2!} + \frac{3^3 e^{-3}}{3!} + \frac{3^4 e^{-3}}{4!}\right) \\ &\approx 0.1847. \end{aligned}$$

c. (6 Points) An "empty zone" is a zone that receives **zero** seeds. What is the expected number of empty zones?

We can use the method of indicator variables. Let $E_i = 1$ if zone i is empty, and $E_i = 0$ otherwise. Then the expected number of empty zones is

$$\mathbb{E}\left[\sum_{i=1}^{400} E_i\right] = \sum_{i=1}^{400} \mathbb{E}[E_i] = \sum_{i=1}^{400} 1 \cdot P(E_i = 1) + 0 \cdot P(E_i = 0) = 400P(E_i = 1)$$

and the probability of a zone being empty can be computed as $P(X = 0)$, where X is the number of seeds in a zone.
From Part (a), we know that $X \sim \text{Bin}(n = 1200, p = 1/400)$ exactly, and from Part (b), we know that $X \sim \text{Poi}(\lambda = 3)$ by a Poisson approximation. We accepted either of these distributions for X to compute $P(X = 0)$. If using Binomial, this is $\left(1 - \frac{1}{400}\right)^{1200}$, and if using Poisson, this is $P(X = 0) = e^{-3}$, making the final answer either $400\left(1 - \frac{1}{400}\right)^{1200}$ or $400e^{-3}$.

4. Are We Cooked? [28 Points]

A cybersecurity firm models the daily alert volume from two server locations, Location A and Location B. They have a different alert detection system for each server location.

Let λ_{attack} represent the average number of security attacks directed at the firm's servers. The firm has a discrete distribution over the possible values of λ_{attack} in which the values of λ_{attack} are discretized in increments of 0.1. Assume you have access to this distribution:

$$P(\lambda_{\text{attack}} = 0.1) = 0.200$$
$$P(\lambda_{\text{attack}} = 0.2) = 0.018$$
$$\vdots$$
$$P(\lambda_{\text{attack}} = 3.0) = 0.050$$

Let A be the number of alerts from Location A and B be the number of alerts from Location B, where:

- $A \sim \text{Bin}(n = 100, p = (1 - e^{-\lambda_{\text{Attack}}}))$
- $B \sim \text{Poi}(\lambda_{\text{Attack}})$
- Given λ_{Attack}, the number of alerts from location A is conditionally independent from the number of alerts from location B.

You may not use code to solve any part of this question. You may refer to your answers from each subpart as answer_a, answer_b, answer_c.

(a) (10 Points) For a given λ_{Attack}, write an expression for the probability of observing exactly 10 total alerts across both locations on a given day.

Since $N = A + B$ and A, B are conditionally independent given λ_{Attack}, we sum over all 11 ways to write $N = 10$ as $a + b$ with $0 \leq a \leq 10$:

$$L(\lambda_{\text{Attack}}) \equiv P(N = 10 \mid \lambda_{\text{Attack}})$$

$$= \sum_{a=0}^{10} \binom{100}{a} (1 - e^{-\lambda_{\text{Attack}}})^a (1 - (1 - e^{-\lambda_{\text{Attack}}}))^{100-a} \cdot \frac{e^{-\lambda_{\text{Attack}}} (\lambda_{\text{Attack}})^{10-a}}{(10-a)!}$$

$$= \sum_{a=0}^{10} \binom{100}{a} (1 - e^{-\lambda_{\text{Attack}}})^a (e^{-\lambda_{\text{Attack}}})^{100-a} \cdot \frac{e^{-\lambda_{\text{Attack}}} (\lambda_{\text{Attack}})^{10-a}}{(10-a)!}$$

(b) (8 Points) On a given day, the firm observes 10 total alerts. What is your updated belief in λ_{Attack} ?

Since we have a uniform prior belief in λ_{Attack} and are given evidence to update our belief, we'll want to use Bayes' Rule here:

$$P(\lambda_{\text{Attack}} = i \mid N = 10) = \frac{P(N = 10 \mid \lambda_{\text{Attack}} = i) * P(\lambda_{\text{Attack}} = i)}{P(N = 10)}$$

$P(N = 10 \mid \lambda_{\text{Attack}} = i)$ from part a.

$P(\lambda_{\text{Attack}} = i)$ given from problem statement.

$$P(N = 10) = \sum_{i=0.1 \text{ and increment by } 0.1}^{3.0} P(N = 10 \mid \lambda_{\text{Attack}} = i) * P(\lambda_{\text{Attack}} = i)$$

(c) (10 Points) The firm tracks a threat clock T representing the time (in days) until the next critical breach. They model $T \sim \text{Exp}(\lambda_{\text{Threat}} = \frac{1}{1000 \cdot \lambda_{\text{Attack}}})$. Given λ_{Attack}, assume T is conditionally independent of both A and B. Given that 10 total alerts were observed today, what is the probability that at least one critical breach occurs within the next 7 days?

For this problem, we can apply the law of total probability over the λ_{attack} value while conditioning on $N = 10$ alerts for the first day:

$$P(T \leq 7 \mid N = 10) = \sum_{\lambda_{\text{Attack}}=0.1}^{3.0} P(T \leq 7 \cap \lambda_{\text{Attack}} \mid N = 10)$$

$$= \sum_{\lambda_{\text{Attack}}=0.1}^{3} P(T \leq 7 \mid \lambda_{\text{Attack}}) \cdot P(\lambda = \lambda_{\text{Attack}} \mid N = 10)$$

$$= \sum_{\lambda_{\text{Attack}}=0.1}^{3} (1 - e^{-7/(1000\lambda_{\text{Attack}})}) \cdot (\text{answer from (b)})$$

5. Two LLMs Walk Into A Bar

You are deciding which of two large language models (LLMs) to deploy in production. To evaluate them, you test both models on the same set of 20 prompts. Model X hallucinates on 3 of the 20 prompts. Model Y hallucinates on 5 of the 20 prompts. Prior to the evaluation, you had a uniform belief for the probability that each model would hallucinate.

a. (5 Points) Let p_X be the probability that Model X hallucinates on any given prompt. What is the probability that p_X is between 0.1 and 0.2? You may not use code to write your answer.

We can model p_X using a Beta, where we have observed 3 successes and 17 failures:

$$X \sim \text{Beta}(a = 4, b = 18)$$

To find the probability that p_X is between 0.1 and 0.2, we can integrate over the Beta PDF:

$$\begin{aligned} P(0.1 < p_X < 0.2) &= \int_{0.1}^{0.2} \frac{1}{B(4,18)} x^3 (1-x)^{17} dx \\ &= \int_{0.1}^{0.2} \frac{\Gamma(4,18)}{\Gamma(4)\Gamma(18)} x^3 (1-x)^{17} dx \\ &= \int_{0.1}^{0.2} \frac{21!}{3!17!} x^3 (1-x)^{17} dx \\ &\approx 0.58 \end{aligned}$$

b. (10 Points) You are in a trial period where beta testers are helping you decide whether to deploy Model X or Model Y, but you still want to give users reasonably accurate responses. Suppose you are given a function **`is_hallucination(model)`** that takes as input either 'X' or 'Y' and returns **`True`** if the chosen model hallucinates on a prompt and **`False`** otherwise. Assume you have already observed the evaluation results from the initial 20 prompts. Write a function that uses Thompson Sampling to choose which model to use for each of the next **10** prompts. Your function should return a list of length **10** containing the selected model, 'X' or 'Y', for each prompt.

Solution 1:

```
def thompson_sampling():
    # Prior: uniform Beta(1,1) for both models
    # After observing evaluation results:
    # Posterior for hallucination prob p_X: Beta(4,18)
    # Posterior for hallucination prob p_Y: Beta(6,16)

    alpha_X, beta_X = 4, 18
    alpha_Y, beta_Y = 6, 16

    # store chosen model for each of 10 prompts
    chosen_models = []

    # run Thompson Sampling for 10 prompts
    for _ in range(10):
        sample_X = random.betavariate(alpha_X, beta_X)
        sample_Y = random.betavariate(alpha_Y, beta_Y)

        # pick model with lower sampled hallucination rate
```

```
        if sample_X < sample_Y:
            # choose X
            hallucinated = is_hallucination('X')
            chosen_models.append('X')

            # increase hallucination count (success for p_X)
            if hallucinated:
                alpha_X += 1
            else:
            # increase non-hallucination count
                beta_X += 1
        else:
            # choose Y
            hallucinated = is_hallucination('Y')
            chosen_models.append('Y')

            if hallucinated:
                alpha_Y += 1
            else:
                beta_Y += 1

    return chosen_models
```

Solution 2:

```
def thompson_sampling():
    # Define successes as non-hallucinations

    params = {}
    params['X'] = {'alpha': 18, 'beta': 4}
    params['Y'] = {'alpha': 16, 'beta': 6}

    chosen_models = []

    for _ in range(10):
        sample_X = beta.rvs(params['X']['alpha'], params['X']['beta'])
        sample_Y = beta.rvs(params['Y']['alpha'], params['Y']['beta'])

        chosen = 'X' if sample_X > sample_Y else 'Y'
        chosen_models.append(chosen)

        hallucinated = is_hallucination(chosen)
        if not hallucinated:
            params[chosen]['alpha'] += 1
        else:
            params[chosen]['beta'] += 1

    return chosen_models
```

B

6. That's Private [16 Points]

A bank has n customers with true account balances $X_1, \ldots, X_n$. Each X_i has mean μ and variance τ^2. For privacy, the bank reports noisy balances $Y_i = X_i + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$ is independent of X_i. Assume n is greater 100. You may not use code to answer any part of this problem.

(a) (6 points) What are $E[Y_i]$ and $\text{Var}(Y_i)$?

$E[Y_i] = \mu$
$\text{Var}(Y_i) = \sigma^2 + \tau^2$

(b) (10 points) A regulator computes the sample mean of the noisy balances, $\bar{Y}$. Find the probability that $\bar{Y}$ is within 0.01 of μ.

$\bar{Y} \sim N(\mu, \frac{\sigma^2+\tau^2}{n})$

$P(|\bar{Y} - \mu| <= 0.01) = \Phi(\frac{0.01}{\sqrt{\frac{\sigma^2+\tau^2}{n}}}) - \Phi(\frac{-0.01}{\sqrt{\frac{\sigma^2+\tau^2}{n}}}) = 2\Phi(\frac{0.01}{\sqrt{\frac{\sigma^2+\tau^2}{n}}}) - 1$

7. Playing it Safe [16 Points]

When a language model generates text, it predicts the next word X by assigning a probability to each word in its vocabulary. We represent this probability distribution as a Python dictionary, where each key is a word and each value is its probability.

We may apply a Safety Filter before sampling the next word. When the filter triggers, it replaces the original distribution with a new distribution called **safe_dist**. This **safe_dist** is still a valid probability mass function over the same set of words, but has more probability mass on safer responses.

In this problem, we will measure how the Safety Filter affects the model's uncertainty using entropy. You may use any functions you defined in previous parts.

(a) (8 points) We define **Collapse** as the reduction in entropy after a filter is applied. Write a function that takes as input two dictionaries: orig (the distribution of X) and safe_dist (the distribution after the filter is applied) and computes and returns the Collapse.

```
def compute_collapse(orig, safe_dist):
    def compute_entropy(pmf):
        total = 0
        for val in pmf.values():
            total -= math.log2(val) * val
        return total

    return compute_entropy(orig) - compute_entropy(safe_dist)
```

(b) (8 Points) The safety filter triggers with some probability p. If the filter does not trigger, the distribution for X remains unchanged. If the filter triggers, the distribution is replaced with **safe_dist**.

Write a Python function that takes as input orig, safe_dist, and p, and computes and returns the expected Collapse caused by the filter.

```
def expected_collapse(orig, safe_dist, p):
    return p * compute_collapse(orig, safe_dist)
```

8. It's Gamma Time [18 Points]

The Gamma distribution is a continuous distribution that generalizes the Exponential. If $X \sim \text{Gamma}(\alpha, \beta)$ then it has the following PDF:

$$f(x \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad \text{for } x > 0$$

You have n i.i.d. samples from a Gamma: $[0.347, 0.811, 1.523, 0.194, 0.612, \ldots]$. Let x_i be the ith value.

The digamma function, $\psi(x)$, gives the derivative of $\log \Gamma(x)$ with respect to x:

$$\psi(x) = \frac{d}{dx} \log \Gamma(x)$$

The digamma function does not have an inverse function, so it is not possible to do $\psi^-1(\alpha)$.

Explain how you would use MLE to estimate the parameters of this distribution and provide any necessary derivatives. You may assume you have access to the function **`digamma(a)`** which returns $\psi(a)$.

MLE Procedure: Likelihood → Log-Likelihood → Gradient → Optimization

Note that we cannot set the derivative to zero and solve in closed form because $\psi(\alpha)$ (the digamma function) does not have a simple inverse. Therefore, we must use gradient-based optimization (e.g., gradient ascent).

Likelihood

$$L(\alpha, \beta) = \prod_{i=1}^{n} f(x_i \mid \alpha, \beta)$$

$$\ell(\alpha, \beta) = \sum_{i=1}^{n} \log\big(f(x_i \mid \alpha, \beta)\big)$$

Gamma Distribution PDF

$$f(x_i \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\beta x_i}$$

Log-Likelihood Expansion

$$\ell(\alpha, \beta) = \sum_{i=1}^{n} \log\left(\frac{\beta^{\alpha}}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\beta x_i}\right)$$

$$= \sum_{i=1}^{n} \left[\alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) \log x_i - \beta x_i\right]$$

Derivative w.r.t. α

$$\frac{\partial \ell(\alpha, \beta)}{\partial \alpha} = \sum_{i=1}^{n} \left[\log \beta - \psi(\alpha) + \log x_i\right]$$

$$= n \log \beta - n\psi(\alpha) + \sum_{i=1}^{n} \log x_i$$

Derivative w.r.t. β

$$\frac{\partial \ell(\alpha, \beta)}{\partial \beta} = \sum_{i=1}^{n} \left[\frac{\alpha}{\beta} - x_i\right]$$

$$= \frac{\alpha n}{\beta} - \sum_{i=1}^{n} x_i$$

9. Genomics and Sampling [18 Points]

You are part of a research team studying the role of the *BRCA1* gene in DNA repair in breast cancer cells. *BRCA1* is a tumor suppressor gene that plays a critical role in the repair of double stranded breaks in DNA.

Your team induces a double stranded DNA break into 50 breast cancer cell samples. Then, after a period of time, your team examines each cell sample and measures two things: the level of *BRCA1* expression and whether the induced double stranded break was successfully repaired.

Let X represent the levels of *BRCA1* expression in a cell and let R be a binary variable where $r = 1$ indicates that the double stranded break was successfully repaired and $r = 0$ indicates repair failure.

Your dataset has 50 observations where each observation i has both the *BRCA1* expression level, x_i, and the repair status, r_i. You are given the dataset as two separate lists

```
X_observations = [2.1, 0.4, 3.7, 1.2, ..., 0.8]   # 50 values
R_observations = [1, 0, 1, 0, ..., 0]          # 50 values
```

where one data point is (`X_observations[i]`, `R_observations[i]`).

You train a logistic regression model on your data and estimate $\theta_1 = 0.8$.

$$P(r = 1) = \sigma(\theta_0 + \theta_1 \cdot x)$$

a. (4 Points) In one sentence, explain how you could interpret the meaning of $\theta_1 = 0.8$. Hint: What does it suggest about the relationship between *BRCA1* expression levels and a cell's ability to repair itself?

> Because θ_1 has a positive value, there is a positive relationship between *BRCA1* expression levels and a cell's ability to repair itself. If there is an increase in *BRCA1* expression levels, there is an increase in the probability of predicting $R = 1$.

b. (10 Points) Because of the small sample size, your colleague is wary about making interpretations based on point estimates of the model weights. Having taken CS109, you remember that distributions can be more informative than point estimates. Use bootstrapping to estimate a sampling distribution for the estimate $\hat{\theta}_1$. You may use the following helper functions:

Function	**Description**
`fit_logistic_regression(x, r)`	Given lists x and r, returns $(\hat{\theta}_0, \hat{\theta}_1)$
`sample_with_replacement(datapoints, n)`	Returns n elements drawn with replacement from datapoints
`sample_without_replacement(datapoints, n)`	Returns n elements drawn without replacement from datapoints

Provide your code below:

```
def estimate_sampling_distribution(X_observations, R_observations):
    # list of indices into both observation lists
    indices = [i for i in range(len(X_observations))] # [0,1,...,49]

    # your code here
```

```
def estimate_sampling_distribution(X_observations, R_observations):
    # list of indices into both observation lists
    indices = [i for i in range(len(X_observations))] # [0,1,...,49]

    # Solution
    sample_dist = []

    # Repeat many times
    for i in range(10,000):
        bootstrap_indices = sample_with_replacement(indices, 50)
        bootstrap_x = X_observations[bootstrap_indices]
        bootstrap_r = R_observations[bootsrap_indices]
        theta_0, theta_1 = fit_logistic_regression(bootstrap_x,
   bootsrap_r)
        sample_dist.append(theta_1)

    return sample_dist
```

Note: Some students attempted to fit a normal distribution to the list of θ_1 samples obtained from bootstrapping. There is no guarantee that a normal distribution accurately describes θ_1, so this is technically incorrect. However, since the point of the problem was understanding bootstrapping, and since the wording of the problem didn't specify what counted as a distribution, we chose to focus only on the bootstrapping portion when scoring.

c. (4 Points) In 1-2 sentences, explain why the manner in which you sample (either with replacement or without replacement) matters for the pseudocode you provided in part b.

> Bootstrapping assumes that we sample **with replacement**. The bootstrapping algorithm assumes that each resample is an independent sample, so we must sample with with replacement. Sampling without replacement leads to dependent samples.

10. The Doctor Will See You Now [18 Points]

A researcher is developing a machine learning model to detect a rare medical condition from X-ray images using Logistic Regression. The model outputs the probability that a patient is sick, $p = P(Y = 1 \mid x)$, using the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{where } z = \theta^T x$$

Assume the model is perfectly calibrated. The cost of different errors is not equal. A **False Negative** (predicting healthy when sick) costs $C_{FN} = 9$. A **False Positive** (predicting sick when healthy) costs $C_{FP} = 1$. A correct prediction costs 0.

(a) (6 Points) What is the expected cost of predicting $\hat{y} = 1$? Express your answer in terms of p.

If we predict $\hat{y} = 1$:

- If $Y = 1$, the prediction is correct and the cost is 0
- If $Y = 0$, this is a false positive and the cost is $C_{FP} = 1$

Since the model is calibrated,

$$P(Y = 1 \mid \hat{y} = 1) = p, \quad P(Y = 0 \mid \hat{y} = 1) = 1 - p$$

The expected cost is

$$\mathbb{E}[\text{cost} \mid \hat{y} = 1] = p \cdot 0 + (1 - p) \cdot 1 = 1 - p$$

(b) (6 Points) At what value of p is the expected cost of predicting $\hat{y} = 1$ equal to the expected cost of predicting $\hat{y} = 0$? We call this value the threshold point, p^*.

Following our logic from part (a), we know:

$$P(Y = 1 \mid \hat{y} = 0) = p, \quad P(Y = 0 \mid \hat{y} = 0) = 1 - p$$

We can calculate the expected cost of predicting $\hat{y} = 0$:

$$\mathbb{E}[\text{cost} \mid \hat{y} = 0] = (1 - p) \cdot 0 + p \cdot 9 = 9p$$

Setting the expected costs equal, we solve for the threshold p^*:

$$9p^* = 1 - p^*$$
$$p^* = 0.1.$$

(c) (6 Points) In standard Logistic Regression, $z = \theta^T x$ and we predict $\hat{y} = 1$ when $z \geq 0$. This is because $\sigma(0) = 0.5$. Find the value of z that defines the decision boundary in this new setting.

Using the sigmoid function

$$p = \frac{1}{1 + e^{-z}}$$

From part b, set $p = 0.1$ to find the decision boundary:

$$\frac{1}{1 + e^{-z}} = 0.1$$

$$1 + e^{-z} = 10$$

$$e^{-z} = 9$$

$$-z = \ln 9$$

$$z = -\ln 9.$$

The decision boundary is

$$z = -\ln 9.$$

That's all! Thank you for the amazing quarter. I feel so lucky to have had the opportunity to teach and learn alongside you all. You were a wonderful class. The Road to Philosophy is a real phenomenon. You can go try it on Wikipedia right now (well after the exam), it is oddly satisfying. Differential privacy is an active area of research and the noise you add to protect individual balances is a real technique used by Apple and Google today. Model collapse is a real and open research problem: when AI models are trained on AI-generated data, they degrade over time. And when AI models are post trained for safety purposes, their entropy is decreased. Nobody fully knows how to fix it yet. Cost-sensitive learning, the framework behind the last problem, is widely used in medical AI where false negatives can be life or death.