

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?

- 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?

- 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.

- 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.

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 kW².

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

- 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?
- 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 .

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

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.

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

- (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?

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.
- 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

```

6. That's Private [16 Points]

A bank has n customers with true account balances $X_1, \dots, X_n$. 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 than 100. You may not use code to answer any part of this problem.

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

(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 .

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: $[0.347, 0.811, 1.523, 0.194, 0.612, \dots]$. Let x_i be the i th 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 $\psi(a)$.

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 $(\mathbf{X_observations[i]}, \mathbf{R_observations[i]})$.

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?

- 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
<code>fit_logistic_regression(x, r)</code>	Given lists x and r , returns $(\hat{\theta}_0, \hat{\theta}_1)$
<code>sample_with_replacement(datapoints, n)</code>	Returns n elements drawn with replacement from datapoints
<code>sample_without_replacement(datapoints, n)</code>	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[bootstrap_indices]
        theta_0, theta_1 = fit_logistic_regression(bootstrap_x,
bootstrap_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.

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, $p = P(Y = 1 | x)$, 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 .
- (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^* .
- (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.