Chris Piech
CS109

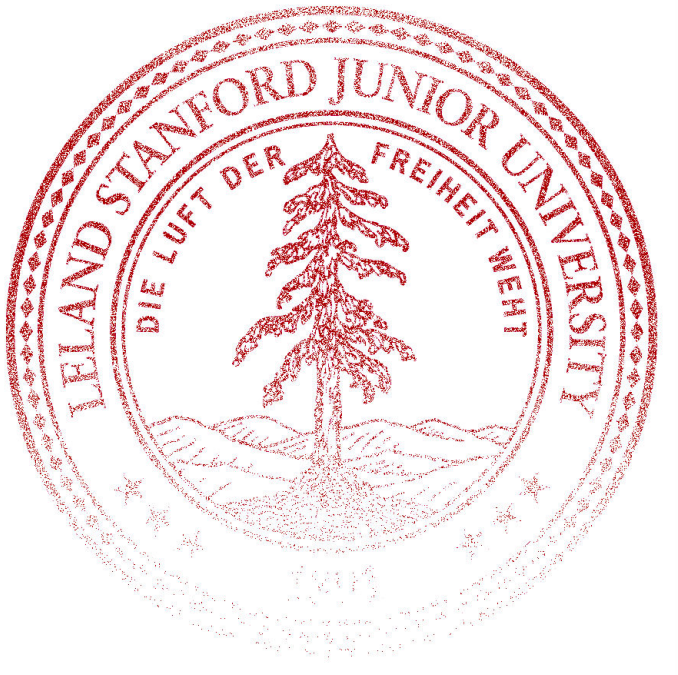
Winter 2022

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): _____

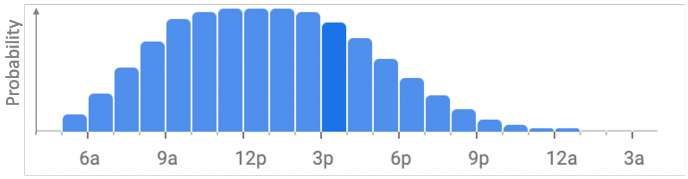
Given Name (print): _____

Email (preferably your gradescope email): _____

1 Short Answer (30 points)

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

a. (6 points) Google Maps stores a probability distribution of the time of day that people go to the San Francisco Airport (SFO):



Let p_i be the probability that a person will show up at SFO i hours past midnight (so p_6 is the probability that they will show up between 6a and 7a). Out of the visitors that will visit SFO on a given day, what fraction do you expect to show up between midnight and noon? Give an expression.

$$\sum_{i=0}^{11} p_i$$

Since $\sum_{i=0}^{23} p_i = 1$, the following is another equivalent answer:

$$\frac{\sum_{i=0}^{11} p_i}{\sum_{i=0}^{23} p_i}$$

b. (7 points) We will consider a datapoint x an "outlier" if the probability of seeing a value x or larger is ≤ 0.05. Consider a distribution $X \sim \text{Geo}(p = 0.8)$. Is 7 an outlier? Give an expression.

7 is an outlier if $1 - \Sigma_{i=0}^{6}(0.2)^{i-1}(0.8) \leq 0.05$

c. (7 points) A class uses High Resolution Course Feedback. In a given week they ask a random sample of 20 people for a numerical rating, instead of asking everyone. Assume the population expectation of rating is 4.5 and the population variance of rating is 1.0. What is the probability that the sample mean rating from 20 students will be ≥ 4.7?

True

A p-value of 0.99 would occur if the difference we observed between the modes of the two datasets is the kind of difference we would observe frequently by chance if the two sets of points were drawn from distributions with the same mode.

d. (10 points) You meet a person at a party who mentions that they have two children. They reveal that they have a daughter who was born on a Tuesday. What is the probability that they have **two daughters**? Assume a child has a ½ chance of being a daughter and a ⅐ chance of being born on a Tuesday. Assume that the genders of the children, and the weekdays they were born on, are all independent of one another.

The answer is 13/27.

Let B be the event that both children are girls. Let T be the event that there is a girl born on a Tuesday.

$$P(B|T) = \frac{P(T|B)P(B)}{P(T)}$$

$$P(B) = 1/4$$

$$P(T|B) = 1/7 + 1/7 - 1/49$$

$$P(T) = 1/14 + 1/14 - 1/196$$

Another explanation:

This is a well known paradox known as the boy-girl-paradox. Intuition says that the answer is 0.5, but it is not!

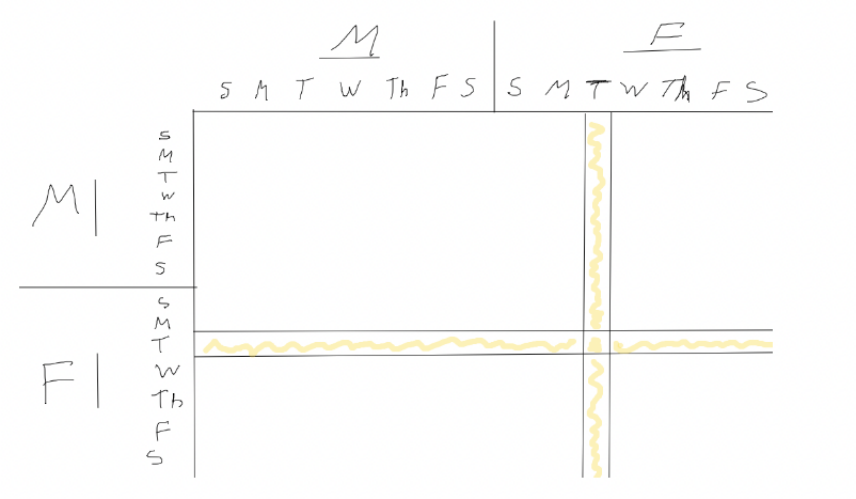


Figure 1: We are conditioning on the yellow region (i.e. think of it as the "new sample space"). We are assessing the probability of the yellow region inside the "female/female" quadrant.

Let the event "there are two daughters" be E.

Let the event "there is at least one daughter born on Tuesday" be F.

$$P(E|F) = \frac{P(EF)}{P(F)}$$

Using Inclusion-Exclusion Principle:

$$= \frac{P(\text{C1 Fem Tue, C2 Fem}) + P(\text{C1 Fem, C2 Fem Tue}) - P(\text{C1 Fem Tue, C2 Fem Tue})}{P(\text{C1 Fem Tue}) + P(\text{C2 Fem Tue}) - P(\text{C1 Fem Tue, C2 Fem Tue})}$$

This turns into

$$\frac{(7+7-1)/14^2}{(14+14-1)/14^2} = \frac{13}{27} = 0.48148148$$

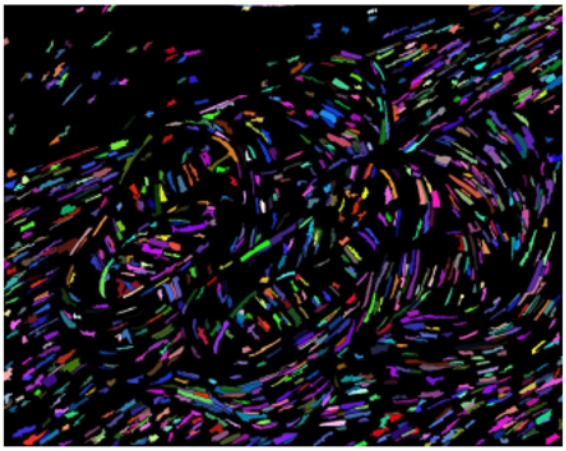
.

2 Quantitative Art History (20 points)

A Stanford alumni discovered that the “Average Number of Brushstrokes in the Neighborhood" is very predictive of whether a painting has been painted by Van Gough [1].

Let $X \sim N(\mu = 20, \sigma^2 = 9)$ be the average number of brushstrokes in the neighborhood if Van Gough is the painter. Let $X \sim N(\mu = 12, \sigma^2 = 25)$ be the average number of brushstrokes in the neighborhood if Van Gough is not the painter.

Aside: A brushstroke j is a neighbor of brushstroke i if they are at most 200 pixels apart. For each brushstroke we can calculate how many other brushstrokes are neighbours. The Average Number of Brushstrokes in the Neighborhood is an average of these values over all brushstrokes in a painting. Here is a painting by Van Gough (left) and the auto-identified brush strokes (right).



a. (4 points) Why is a normal distribution appropriate for the “average number of brushstrokes in the neighborhood" of a painting?

Must mention the central limit theorem.

b. (16 points) A painting is discovered. Based on background information your prior belief that it was painted by Van Gough is 0.75. What is your updated belief after observing that the average number of brushstrokes in the neighbourhood is 19?

This is an inference question: Let Y be the event that Van gough was the artist.

$$P(Y|X = 19) = \frac{f(X = 19|Y)P(Y)}{f(X = 19|Y)P(Y) + f(X = 19|Y)P(Y^C)}$$

3 Beta ± (20 points)

You have observed 30 successes and 20 fails for an event with unknown probability. Based on this information you can model the probability of success, X, as a Beta random variable. You want to make a claim of the form: `"the probability of success is` $m \pm b$`."`

Select m to be the expectation of X. Select b to be the smallest value, ***rounded to two decimal places***, such that $P(m - b < X < m + b)$ is greater than or equal to 95%. Provide your answer as pseudocode that prints out the values m and b.

Incremental solution:

```
def main():
    X = stats.beta(A, B)
    mean = A / (A + B)
    x = 0
    while True:
        lower = mean - x
        upper = mean + x
        pr = X.cdf(upper) - X.cdf(lower)
        if pr > 0.95:
            break
        x += 0.01
    print(f'{mean:.2f} +- {x:.2f}')
```

4 Bayesian Carbon Dating (34 points)

We are able to know the age of ancient artefacts using a process called carbon dating. This process involves a lot of uncertainty! Living things have a constant proportion of a molecule called C14 in them. When living things die those molecules start to decay. The time to decay in years, T, of a C14 molecule is distributed as an exponential. $T \sim \text{Exp}(\lambda = 1/8267)$.

a. (5 points) Consider a single C14 molecule. What is the probability that it decays within 500 years?

$P(T \leq 500) = 1 - e^{-\frac{1}{8267} * 500} = 0.05868875306$

b. (8 points) C14 molecules decay independently. A particular sample started with 100 molecules. What is the probability that exactly 95 are left after 500 years? Let p be your answer to part a.

X ~ Bin(n=100, p = 0.0586)
P(X = 5) = scipy.stats.binom.pmf(5, 100, 0.05868)

c. (6 points) Write pseudocode for a function `pr_measure_given_age(m, age)` which returns $P(M = \texttt{m} | A = \texttt{age})$, the probability that exactly `m` molecules are left out of the original 100 after exactly `age` number of years.

d. (15 points) You observe a measurement of 95 C14 molecules in a sample. You assume that the sample originally had 100 C14 molecules when it died. Write pseudocode for a function `age_belief()` that returns a list of length 1000 where the value at index i in the list stores $P(A = i | M = 95)$. Age is a discrete random variable which takes on whole numbers of years. $A = i$ is the event that the sample organism died i years ago. You may use the function `pr_measure_given_age(m, age)` from part c. For your prior belief: you know that the sample ***must*** be between $A = 500$ and $A = 600$ inclusive and you assume that every year in that range is equally likely.

5 Reliability engineering (23 points)

The “reliability distribution" is a random variable parameterized by a with PDF:

$$f(X = x) = \frac{1}{a^2} x^{a-1} e^{-\frac{x^2}{a^2}}$$

We wish to model how long a particular model of phone will function before it breaks. We are going to use a reliability distribution. To this end we collect N independent measurements of how long the type of phone functions before it breaks: $x_1, x_2, \ldots, x_N$. Explain, in words, how you would choose parameter a using the maximum likelihood estimation framework, and provide any necessary derivatives.

First, we start by defining our likelihood function.

$$L(a) = \prod_{i=1}^{N} f(X_i|a)$$

We want to determine $\hat{a} = \underset{a}{\text{argmax}} L(a)$. Since the argmax of $L(a)$ is equivalent to the argmax of the log of $L(a)$ and logs make the math simpler we write:

$$LL(a) = \log\left(\prod_{i=1}^{N} f(X_i|a)\right)$$

Plugging in the provided PDF and simplifying:

$$LL(a) = \log\left(\prod_{i=1}^{N} \frac{1}{a^2} x_i^{a-1} e^{-\frac{x_i^2}{a^2}}\right)$$

$$LL(a) = \sum_{i=1}^{N} \left[\log\left(\frac{1}{a^2} x_i^{a-1} e^{-\frac{x_i^2}{a^2}}\right)\right]$$

$$LL(a) = \sum_{i=1}^{N} \left[\log\left(\frac{1}{a^2}\right) + \log\left(x_i^{a-1}\right) + \log\left(e^{-\frac{x_i^2}{a^2}}\right)\right]$$

$$LL(a) = \sum_{i=1}^{N} \left[-2\log(a) + (a-1)\log(x_i) - \frac{x_i^2}{a^2}\log(e)\right]$$

$$LL(a) = \sum_{i=1}^{N} -2\log(a) + \sum_{i=1}^{N} (a-1)\log(x_i) - \sum_{i=1}^{N} \frac{x_i^2}{a^2}\log(e)$$

$$LL(a) = -2N\log(a) + \sum_{i=1}^{N} a\log(x_i) - \sum_{i=1}^{N} \log(x_i) - \frac{1}{a^2}\sum_{i=1}^{N} x_i^2$$

Now we will take the partial derivative of the log likelihood with respect to the parameter of interest a:

$$\frac{\partial LL(a)}{\partial a} = \frac{\partial}{\partial a}\left(-2N\log(a) + \sum_{i=1}^{N} a\log(x_i) - \sum_{i=1}^{N} \log(x_i) - \frac{1}{a^2}\sum_{i=1}^{N} x_i^2\right)$$

$$\frac{\partial LL(a)}{\partial a} = -\frac{2N}{a} + \sum_{i=1}^{N} \log(x_i) + \frac{2}{a^3}\sum_{i=1}^{N} x_i^2$$

From here, we would set the partial derivative equal to 0 and solve for a.

Note: You were required to provide the necessary derivatives but you did not need to actually determine the value of a as the problem states. Any math beyond the calculation of the partial derivative was not required or penalized.

6 Rank ordering of samples (25 points)

A random variable X can take on three values: 1, 2 or 3 with the following probabilities:

Value	Probability
1	0.625
2	0.250
3	0.125

Consider n IID samples from X. The samples are in "correct rank order" if count(1) > count(2) > count(3) such that "count" is the number of samples with the given value. Samples are ***not*** in correct rank order if number of 1s <= number of 2s, or if the number of 2s <= number of 3s.

a. (7 points) We draw 3 IID samples. What is the probability they are in "correct rank order"?

With 3 IID samples, there must two 1's and one 2 in order for the samples be in correct rank order. The probability of this is $0.625^2 * 0.250 * 3$.

b. (18 points) If n IID samples are collected, what is the probability that they will be in "correct rank order"? Write pseudo-code that calculates the probability. For full credit, your algorithm should provide an exact answer (not a sampling approximation) and should run in polynomial time.

```
import math

counts = []
for i in range(n + 1):
    for j in range(n - i + 1):
        if i > j and j > n - i - j:
            counts.append([i, j, n - i - j])

p = 0.0
for c in counts:
    # mutinomial formula
    coefficient = math.factorial(n) / (math.factorial(c[0]) * math.factorial(c[1]) *
        math.factorial(c[2]))
    p += coefficient * (0.625 ** c[0]) * (0.250 ** c[1]) * (0.125 ** c[2])
```

7 Machine Teaching (28 points)

In this problem you will write a machine teaching algorithm that can select three datapoints from a dataset (eg the Netflix dataset), such that a logistic regression algorithm trained on just those three datapoints would have the best possible parameters. You are given two functions and a dataset:

`theta = train(dataset)`
an implementation of logistic regression training. It returns `theta`, a list of $m + 1$ weights, where m is the number of features in the dataset. `theta[0]` is the intercept and `theta[i]` is the weight for feature i.

`all_combinations = comb(items_list, k)`
which returns a list of all unique sets of size k from the input list of items. For example `comb([1,2,3], 2)` would return `[[1,2],[1,3],[2,3]]`

`dataset`
is a list of dictionaries (one for each datapoint). Each datapoint has both a list of **x** values as well as a single label y. All the values are binary. Here is an example with four datapoints and $m = 3$:

```
dataset = [
   {"x": [0,0,1], "y": 1},
   {"x": [1,0,1], "y": 1},
   {"x": [1,1,1], "y": 0},
   {"x": [0,0,0], "y": 0},
]
```

a. (5 points) What is the length of the list returned by **`comb`** in terms of n, the length of `items_list`, when $k = 3$? For full credit, give both the exact number and the big-O notation. Assume that each item is distinct and that the same item can't show up more than once in any set of three.

There are $\binom{n}{3}$ possible combinations of 3, so this is the length of comb. If we expand, we get $\binom{n}{3} = \frac{n!}{3!(n-3)!} = \frac{n(n-1)(n-2)}{3!} = \frac{n(n-1)(n-2)}{6}$. From this, we can estimate the Big-O: $O(n^3)$.

b. (5 points) What is the logistic regression likelihood for three datapoints $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and parameters θ? Recall that $|x_i| = m$ and $|\theta| = m + 1$.

First we append a 1 to the beginning of x_1, x_2, and x_3. This allows us to create the "bias" term and make $|x_i| = m + 1$. For example $(\theta^T x_1) = \theta_0 + \theta_1 x_1^{(1)} + \theta_2 x_1^{(2)} + \theta_3 x_1^{(3)}$

$$L(\theta) = \prod_{i=1}^{n} P(Y = y^{(i)} | X = x^{(i)})$$

$$L(\theta) = \prod_{i=1}^{3} \sigma(\theta^T x_i)^{y_i} \cdot [1 - \sigma(\theta^T x_i)]^{1-y^{(i)}}$$

c. (14 points) Write pseudo-code that returns a list of **three** datapoints from a given dataset. Select the three datapoint, such that a logistic regression algorithm trained on just those three datapoints will have the best possible parameters. The best parameters are ones which maximise the likelihood of the full dataset. You can (and should) define helper functions:

```
def machine_teach(dataset):
```

d. (4 points) Consider a randomly chosen set of three datapoints. A logistic regression classifier is trained on just those three datapoints. For the learned θ do you expect the likelihood of those three datapoints to be larger or smaller than the likelihood of the whole dataset? Explain.

Larger. The classifier will overfit those three datapoints during training, i.e. learn θ that maximizes the likelihood of those 3 datapoints. However, since these three datapoints probably don't cover the diversity in the dataset, we expect the likelihood of those three datapoints to be larger than that of the whole dataset.

That's all folks. Thank you all for the wonderful quarter and we hope you have a fantastic spring break. Machine teaching is a real field which has implications for education. Carbon Dating had very minor simplifications from the true probabilistic algorithm! Many things in the natural world are analyzed in terms of their rank ordering. Reliability engineering introduced a version of the Weibull distribution. This was a great class and the teaching team really appreciated how positive, curious and intelligent you all were. All the best.

[1] http://infolab.stanford.edu/~wangz/project/imsearch/ART/PAMI11/li.pdf