1 Warmups

1.1 Conditional Expectation
Let $X \sim Geo(p)$. Use the Law of Total Expectation to prove that $E[X] = 1/p$, by conditioning on whether the first flip is heads or tails.

$$E[X] = E[X|H]P(H) + E[X|T]P(T) = 1 \cdot p + (E[1 + X])(1 - p)$$
Solving yields $E[X] = 1/p$.

1.2 MLE
Suppose $x_1, \ldots, x_n$ are i.i.d. (independent and identically distributed) values sampled from some distribution with density function $f(x|\theta)$, where $\theta$ is unknown. Recall that the likelihood of the data is

$$L(\theta) = f(x_1, x_2, \ldots, x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta)$$

Recall we solve an optimization problem to find $\hat{\theta}$ which maximizes $L(\theta)$, i.e., $\hat{\theta} = \arg \max_{\theta} L(\theta)$.

1. Why can we optimize $LL(\theta)$ rather than $L(\theta)$?
2. Why ought we optimize $LL(\theta)$ rather than $L(\theta)$?

1. $\sum_{i=1}^{n} \log f(x_i|\theta)$
2. Logarithms are monotonic. For any monotonic function $f$ and any function $g$, the following holds: $\arg \max g(x) = \arg \max f(g(x))$
3. Finding the max of a function requires taking derivatives. The original expression consists of a product of many functions of $\theta$. This would lead to a difficult derivation because of the chain rule of calculus.

By taking the log, we convert the product into sums, and the derivative of the sum of many functions of $\theta$ is much easier to compute.

That is, $\frac{\partial}{\partial \theta} \sum_i f_i(\theta) = \sum_i \frac{\partial}{\partial \theta} f_i(\theta)$ which is easy.

However, $\frac{\partial}{\partial \theta} \prod_i f_i(\theta)$ is a complicated expression.
1.3 Maximum A Posteriori

1. Intuitively, what is MAP? What problem is it trying to solve? How does it differ from MLE?

2. Given a 6-sided die (possibly unfair), you roll the die $N$ times and observe the counts for each of the 6 outcomes as $n_1, ..., n_6$. What is the maximum a posteriori estimate of this distribution, using Laplace smoothing? Recall that the die rolls themselves follow a multinomial distribution.

1. From the course notes: The paradigm of MAP is that we should choose the value for our parameters that is the most likely given the data. At first blush this might seem the same as MLE; however, remember that MLE chooses the value of parameters that makes the data most likely. One of the disadvantages of MLE is that it best explains data we have seen and makes no attempt to generalize to unseen data. In MAP, we incorporate prior belief about our parameters, and then we update our posterior belief of the parameters based on the data we have seen.

2. Using a prior which represents one imagined observation of each outcome is called Laplace smoothing and it guarantees that none of your probabilities are 0 or 1. The Laplace estimate for a Multinomial RV is $p_i = \frac{n_i + 1}{N + 6}$ for $i = 1, ..., 6$.

1.4 Naive Bayes

Recall the classification setting: we have data vectors of the form $X = (X_1, \ldots, X_m)$ and we want to predict a label $Y \in \{0, 1\}$.

1. Recall in Naive Bayes, given a data point $x$, we compute $P(Y = 1|X = x)$ and predict $Y = 1$ provided this quantity is $\geq 0.5$, and otherwise we predict $Y = 0$. Decompose $P(Y = 1|X = x)$ into smaller terms, and state where the Naive Bayes assumption is used.

2. Suppose we are given example vectors with labels provided. Give a formula to estimate (using maximum likelihood) each quantity $P(X_i = x_i|Y = y)$ above, for $i \in \{1, \ldots, m\}$ and $y \in \{0, 1\}$. You can assume there is a function count which takes in any number of boolean conditions and returns a count over the data of the number of examples in which they are true. For example, count$(X_3 = 2, X_5 = 7)$ returns the number of examples where $X_3 = 2$ and $X_5 = 7$.

1. $P(Y = 1|X = x) = \frac{P(Y = 1)P(X = x|Y = 1)}{P(Y = 1)P(X = x|Y = 1) + P(Y = 0)P(X = x|Y = 0)}$ (Bayes+LTP)

2. $P(X_i = x_i|Y = y) = \frac{\text{count}(X_i = x_i, Y = y)}{\text{count}(Y = y)}$ (NB Assumption)
2 Problems

2.1 Random Number of Random Variables

Let \( N \) be a non-negative integer-valued random variable—that is, a random variable that takes on values in \{0, 1, 2, \ldots\}. Let \( X_1, X_2, X_3, \ldots \) be an infinite sequence of independent and identically distributed random variables (independent of \( N \)), each with mean \( \mu \), and \( X = \sum_{i=1}^{N} X_i \) be the sum of the first \( N \) of them.

Before doing any work, what do you think \( E[X] \) will turn out to be? Then show it mathematically to see if your intuition is correct.

\[
E[X] = E \left[ \sum_{i=1}^{N} X_i \right] = \sum_{n} E \left[ \sum_{i=1}^{N} X_i \mid N = n \right] p_N(n) = \sum_{n} E \left[ \sum_{i=1}^{n} X_i \mid N = n \right] p_N(n) \\
= \sum_{n} E \left[ \sum_{i=1}^{n} X_i \right] p_N(n) = \sum_{n} n \mu p_N(n) = \mu \sum_{n} n p_N(n) = \mu E[N]
\]

Alternatively,
\[
E[X] = E[E[X \mid N]] = E[\mu N] = \mu E[N]
\]

2.2 Binary Trees

Consider the following function for constructing binary trees:

```c
struct Node {
    Node *left;
    Node *right;
};

Node *randomTree(float p) {
    if (randomBool(p)) { // returns true with probability p
        Node *newNode = new Node;
        newNode->left = randomTree(p);
        newNode->right = randomTree(p);
        return newNode;
    } else {
        return nullptr;
    }
}
```

The if branch is taken with probability \( p \) (and the else branch with probability \( 1 - p \)). A tree with no nodes is represented by nullptr; so a tree node with no left child has nullptr for the left field (and the same for the right child).
Let $X$ be the number of nodes in a tree returned by `randomTree`. You can assume $0 < p < 0.5$. What is $E[X]$, in terms of $p$?

Let $X_1$ and $X_2$ be number of nodes the left and right calls to `randomTree`. 

$E[X_1] = E[X_2] = E[X]$. 

$$
E[X] = p \cdot E[X | if] + (1 - p)E[X | else] = p \cdot E[1 + X_1 + X_2] + (1 - p) \cdot 0 = p \cdot (1 + E[X] + E[X]) = p + 2pE[X] \\
(1 - 2p)E[X] = p \\
E[X] = \frac{p}{1 - 2p}
$$

### 2.3 Variance of Hemoglobin Levels

A medical researcher treats patients with dangerously low hemoglobin levels. She has formulated two slightly different drugs and is now testing them on patients. First, she administered drug A to one group of 50 patients and drug B to a separate group of 50 patients. Then, she measured all the patients’ hemoglobin levels post-treatment. For simplicity, assume that all variation in the patient outcomes is due to their different reactions to treatment.

The researcher notes that the sample mean is similar between the two groups: both have mean hemoglobin levels around 10g/dL. However, drug B’s group has a sample variance that is 3 (g/dL)$^2$ greater than drug A’s group. The researcher thinks that patients respond to drugs A and B differently. Specifically, she wants to make the scientific claim that drug A’s patients will end up with a significantly different spread of hemoglobin levels compared to drug B’s.

You are skeptical. It is possible that the two drugs have practically identical effects and that the observed different in variance was a result of chance and a small sample size, i.e. the null hypothesis. Calculate the probability of the null hypothesis using bootstrapping. Here is the data. Each number is the level of an independently sampled patient:

**Hemoglobin Levels of Drug A’s Group** ($S^2 = 6.0$): 13, 12, 7, 16, 9, 11, 7, 10, 9, 8, 9, 7, 16, 7, 9, 8, 13, 10, 11, 9, 13, 10, 10, 9, 7, 7, 6, 7, 8, 12, 13, 9, 6, 9, 11, 10, 8, 12, 10, 9, 10, 8, 14, 13, 13, 10, 11, 12, 9

**Hemoglobin Levels of Drug B’s Group** ($S^2 = 9.1$): 8, 8, 16, 16, 9, 13, 14, 13, 10, 12, 10, 6, 14, 8, 13, 14, 7, 13, 7, 8, 4, 11, 7, 12, 8, 9, 12, 8, 11, 10, 12, 6, 10, 15, 11, 12, 3, 8, 11, 10, 10, 8, 12, 8, 11, 6, 7, 10, 8, 5

Complete the prompts in the following Colab notebook to investigate this question using bootstrapping.

https://colab.research.google.com/drive/156KhKnXp71fLQmYQ0S7MUp0XFq_iEzI?usp=sharing
2.4 Bootstrapping and Null Hypotheses (courtesy of Oishi Banerjee)

While testing the efficacy of a new drug, Skylar Pharmaceuticals has collected 1000 data samples. Most of the samples came from patients who were treated with the drug, but the rest came from patients who received a placebo. Skylar observed that the sample mean blood pressure in the treated group was 80, while the sample mean blood pressure in the placebo group was 86. To demonstrate the difference is statistically significant, Skylar implemented the following to produce a p-value.

Data scientists at Skylar wrote the following bootstrapping code to arrive at a p-value to suggest the difference isn’t statistically significant. Unfortunately (or fortunately, depending on your point of view), their code is not right.

```python
import numpy as np
def resample(whole, num_samples):
    return np.random.choice(whole, num_samples, replace=True)

# list_treat is an ordinary 1-d numpy array
# it contains all the diastolic blood pressures of
# each patient who was treated

# list_placebo is an ordinary 1-d numpy array
# it contains the diastolic blood pressures of each
# patient who received a placebo
def pvalue(list_treat, list_placebo):
    # np.concatenate will make a 1000-element array containing the elements of
    # both list_treat and list_placebo
    whole = np.concatenate([list_treat, list_placebo])
    threshold = np.mean(list_treat) - np.mean(list_placebo)
    counter, num_trials = 0, 100000
    for trial in range(num_trials):
        sample_treat = resample(list_treat, 500)
        sample_placebo = resample(list_placebo, 500)
        mean_treat = np.mean(sample_treat)
        mean_placebo = np.mean(sample_placebo)
        new_diff = np.abs(mean_treat - mean_placebo)
        if new_diff == threshold: counter += 1
    return counter/num_trials
```

Point out the algorithmic errors. Be clear what Skylar should do instead, explaining why each change you would make is necessary for correct bootstrapping.

- As written, the threshold will be -6, so new_diff can never be smaller than threshold! Because were really only concerned with magnitudes, Skylar should replace threshold with np.abs(threshold).
• To simulate the null hypothesis, we should sample from our new combined distribution. As a result both calls to resample should pass in whole, not list_treat or list_placebo.

• Though were now sampling from our new combined distribution, we want to stay true to the design of the original experiment in every other way. Therefore we should make sure sample_treat has as many elements as list_treat and sample_placebo has as many elements as list_placebo. The 500s should be replaced with len(list_treat) and len(list_placebo) respectively.

• When bootstrapping, we count up how many times we see a result as dramatic or more dramatic than ours under the null hypothesis. As a result, we should check if new_diff is greater than or equal to threshold.

2.5 The Honor Code

We have decided that automated tools should be used to identify if two submissions are suspiciously similar. (N.B. these tools are never used as a basis for community standards cases — those always require professional human opinion.) However, automated tools are never perfect. As active CS109 students, we would like to explore the chance of a false positive in automated tools. For this task, we will consider Breakout, a CS106A assignment where students implement Breakout.

2.5.1 Single Match

Say there are 1000 decision points when writing Breakout. Assume: Each decision point is binary. Each student makes all 1000 decisions. For each decision there is a probability $p$ that a student takes the more popular choice. What is the probability distribution for the number of matching decisions (we are going to refer to this as the “score”)? If possible, could you approximate this probability?

Let $A_i$ be the event that decision point $i$ is matched. We note that a match occurs when both students make the more popular choice or when both students make the less popular choice. $P(A_i) = P(\text{Both more popular}) + P(\text{Both less popular}) = p^2 + (1 - p)^2$.

Let $M$ be a random variable for the number of matches. It is easy to see that each of the 1000 decisions is an independent Bernoulli experiment with probability of success $p' = p^2 + (1 - p)^2$. Therefore $M \sim Bin(1000, p')$.

We can use a Normal distribution to approximate a binomial. We approximate $M \sim Bin(1000, p')$ with Normal random variable $Y \sim N(1000p', 1000p'(1 - p'))$. 

2.5.2 Maximum Match

When we look at two assignments, the probability of a false match is exceedingly small. What would the max similarity score look like when we compare one student to 5000 historical breakout submissions? Let $X_i$ be the similarity score between a student who worked on their own and student $i$. Let $Y$ be the highest match score between the student and all other submissions:

$$Y = \max_i X_i$$

The Central Limit Theorem tells us about the distribution of the sum of IID random variables. A more obscure theorem, the Fisher-Tippett-Gnedenko theorem, tells us about the max of IID random variables. It says that the max of IID exponential or normal random variables will be a “Gumbel” random variable.

$$Y \sim \text{Gumbel}(\mu, \beta)$$

The max of IID vars

$$f(Y = k) = \frac{1}{\beta} e^{-(z + e^{-z})}$$

where $z = \frac{k - \mu}{\beta}$

The Gumbel PDF

You are given data of 1300 students’ max scores from three quarters (we believe they all worked independently): $y^{(1)} \ldots y^{(1300)}$. Set up (but do not solve) simultaneous equations we could solve to find the values of $\mu$ and $\beta$.

For this problem, we use Maximum Likelihood Estimator (MLE) to estimate the parameters $\theta = (\mu, \beta)$.

$$L(\theta) = \prod_{i=1}^{n} f(Y^{(i)} = y^{(i)} \mid \theta)$$

$$LL(\theta) = \log \prod_{i=1}^{n} f(Y^{(i)} = y^{(i)} \mid \theta)$$

$$= \sum_{i=1}^{n} \log f(Y^{(i)} = y^{(i)} \mid \theta)$$

$$= \sum_{i=1}^{n} \log \frac{1}{\beta} e^{-(z_i + e^{-z_i})}$$

where $z_i = \frac{y^{(i)} - \mu}{\beta}$

$$= \sum_{i=1}^{n} \log \frac{1}{\beta} + \sum_{i=1}^{n} -(z_i + e^{-z_i})$$

$$= -n \log(\beta) + \sum_{i=1}^{n} -(z_i + e^{-z_i})$$

Now we must choose the values of $\theta = (\mu, \beta)$ that maximize our log-likelihood function. First, we need to find the first derivative of the log-likelihood function with respect to our parameters.
\[
\frac{\partial LL(\theta)}{\partial \mu} = \frac{\partial}{\partial \mu} \left[ -n \log(\beta) + \sum_{i=1}^{n} (z_i + e^{-z_i}) \right] \\
= \sum_{i=1}^{n} \frac{\partial}{\partial \mu} \left[ (z_i + e^{-z_i}) \right] \\
= \sum_{i=1}^{n} \frac{\partial}{\partial z_i} \left[ (z_i + e^{-z_i}) \right] \frac{\partial z_i}{\partial \mu} \\
= \sum_{i=1}^{n} \left[ -1 + e^{-z_i} \right] \left[ - \frac{1}{\beta} \right] \\
= \frac{1}{\beta} \sum_{i=1}^{n} 1 - e^{-z_i}
\]

By the Chain Rule

\[
\frac{\partial LL(\theta)}{\partial \beta} = \frac{\partial}{\partial \beta} \left[ -n \log(\beta) + \sum_{i=1}^{n} (z_i + e^{-z_i}) \right] \\
= -\frac{n}{\beta} + \sum_{i=1}^{n} \frac{\partial}{\partial \beta} \left[ (z_i + e^{-z_i}) \right] \\
= -\frac{n}{\beta} + \sum_{i=1}^{n} \frac{\partial}{\partial z_i} \left[ (z_i + e^{-z_i}) \right] \frac{\partial z_i}{\partial \beta} \\
= -\frac{n}{\beta} + \sum_{i=1}^{n} \left[ -1 + e^{-z_i} \right] \left[ \frac{1}{\beta} \right] \\
= -\frac{n}{\beta} \sum_{i=1}^{n} \left[ 1 - \frac{\mu - y^{(i)}}{\beta^2} \right] \\
= \frac{1}{\beta} \sum_{i=1}^{n} \left[ 1 - \frac{\mu - y^{(i)}}{\beta^2} \right]
\]

Where the last term equals \( \frac{\partial z_i}{\partial \beta} \)

We want to find a simultaneous solution for both, but this is algebraically not possible. We would instead use an approximation method called gradient ascent/descent (to be taught later this week) to solve for these.

### 2.6 Multiclass Bayes

In this problem we are going to explore how to write Naive Bayes for multiple output classes. We want to predict a single output variable \( Y \) which represents how a user feels about a book. Unlike in your homework, the output variable \( Y \) can take on one of the four values in the set \{Like, Love, Haha, Sad\}. We will base our predictions off of three binary feature variables \( X_1, X_2, \) and \( X_3 \) which are indicators of the user’s taste. All values \( X_i \in \{0, 1\} \).

We have access to a dataset with 10,000 users. Each user in the dataset has a value for \( X_1, X_2, X_3 \) and \( Y \). You can use a special query method \texttt{count} that returns the number of users in the dataset with the given equality constraints (and only equality constraints). Here are some example usages of \texttt{count}:
\text{count}(X_1 = 1, Y = \text{Haha}) \quad \text{returns the number of users where } X_1 = 1 \text{ and } Y = \text{Haha.}
\text{count}(Y = \text{Love}) \quad \text{returns the number of users where } Y = \text{Love.}
\text{count}(X_1 = 0, X_3 = 0) \quad \text{returns the number of users where } X_1 = 0, \text{ and } X_3 = 0.

You are given a new user with \(X_1 = 1, X_2 = 1, X_3 = 0\). What is the best prediction for how the user will feel about the book \((Y)\)? You may leave your answer in terms of an \text{argmax} function. You should explain how you would calculate all probabilities used in your expression. Use \textbf{Laplace estimation} when calculating probabilities.

We can make the Naive Bayes assumption of independence and simplify \text{argmax} of \(P(Y|X)\) to get an expression for \(\hat{Y}\), the predicted output value, and evaluate it using the provided \text{count} function.

\[
\hat{Y} = \arg \max_y \frac{P(X_1 = 1, X_2 = 1, X_3 = 0|Y = y)P(Y = y)}{P(X_1 = 1, X_2 = 1, X_3 = 0)}
\]

\[
= \arg \max_y P(X_1 = 1, X_2 = 1, X_3 = 0|Y = y)P(Y = y)
\]

\[
= \arg \max_y P(X_1 = 1|Y = y)P(X_2 = 1|Y = y)P(X_3 = 0|Y = y)P(Y = y), \text{ where:}
\]

\[
P(X_1 = 1|Y = y) = \frac{\text{count}(X_1 = 1, Y = y) + 1}{\text{count}(Y = y) + 2}
\]

\[
P(X_2 = 1|Y = y) = \frac{\text{count}(X_2 = 1, Y = y) + 1}{\text{count}(Y = y) + 2}
\]

\[
P(X_3 = 0|Y = y) = \frac{\text{count}(X_3 = 0, Y = y) + 1}{\text{count}(Y = y) + 2}
\]

where we evaluate for \(y \in \{0, 1, 2, 3\}\)

You don’t need to use smoothing to estimate \(P(Y)\): \(P(Y = y) = \text{count}(Y = y)/10,000\)