Section 9: Final Exam Review

Before you leave lab, make sure you click here so that you’re marked as having attended this week’s section. The CA leading your discussion section can enter the password needed once you’ve submitted. **Note:** This is your last section this quarter, so be sure to say goodbye to everyone in your section and wish them well on the final and over the summer.

1 Warmups

1.1 Maximum A Posteriori

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

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

a. 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.

b. 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 \( i = 1, \ldots, 6 \).

1.2 Naive Bayes Review

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

a. 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.
b. 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 $X_5 = 7$.

\[
P(Y = 1 | X = x) = \frac{P(Y = 1) P(X = x | Y = 1)}{P(Y = 1) P(X = x | Y = 1) + P(Y = 0) P(X = x | Y = 0)} \quad \text{(Bayes+LTP)}
\]

\[
P(Y = 1) \prod_{i=1}^{m} P(X_i = x_i | Y = 1) + P(Y = 0) \prod_{i=1}^{m} P(X_i = x_i | Y = 0) \quad \text{(NB Assumption)}
\]

b. $P(X_i = x_i | Y = y) = \frac{\text{count}(X_i = x_i, Y = y)}{\text{count}(Y = y)}$

2 Problems

2.1 Bayesian Carbon Dating

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

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

b. 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.

\[
X \sim \text{Bin}(n = 100, p = 0.0586)
P(X = 5) = \text{scipy.stats.binom.pmf}(5, 100, 0.05868)
\]

c. Write pseudocode for a function pr_measure_given_age(m, age) which returns $P(M = m | A = \text{age})$, the probability that exactly $m$ molecules are left out of the original 100 after exactly age number of years.
Consider each of the 100 original molecules as an independent trials of a C14 molecule. We can model :math:`\text{pr\_measure\_given\_age}(m, \text{age})` as a binomial with parameters 100 and probability of a single element left after some age.

```python
def pr_measure_given_age(m, age):
    # Find the exponential probability
    p = 1 - np.exp(-(1/8267) * age)

    # Find the amount of successes after 100 trials.
    pr_m_given_age = scipy.stats.binom.pmf(100 - m, 100, p)
    return pr_m_given_age
```

d. 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 :math:`i` in the list stores :math:`P(A = i|M = 95)`. Age is a discrete random variable which takes on whole numbers of years. :math:`A = i` is the event that the sample organism died :math:`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 :math:`A = 500` and :math:`A = 600` inclusive and you assume that every year in that range is equally likely.

First apply Bayes rule

\[
P(A = i|M = 95) = \frac{P(M = 95|A = i)P(A = i)}{P(M = 95)}
\]

We will write psudo-code using this equation. A few notes,

a. :math:`P(M = 95|A = i) = \text{pr\_measure\_given\_age}(95, i)`

b. :math:`P(A = i) = 1 / (100 + 1)`, with 101 because 500 to 600 inclusive.

c. :math:`P(M = 95)` Can be disregarded in the code since it is only a normalization term (e.g, we normalize the returned array).

With that, here’s the code:
2.2 Continuous Joint Distributions

a. Let $X$, $Y$, and $Z$ be independent Normal variables with means of $\mu_X = 4$, $\mu_Y = 5$, and $\mu_Z = 6$ and variances $\sigma^2_X = 16$, $\sigma^2_Y = 25$, and $\sigma^2_Z = 36$. Let $A = X + Y$ and $B = Y + Z$. It can be shown that the joint distribution $(A, B)$ is Bivariate Normal. What are the parameters of the joint distribution $(A, B)$?

$$(A, B) \sim N(\mu, \Sigma), \mu = \begin{bmatrix} \mu_X + \mu_Y \\ \mu_Y + \mu_Z \end{bmatrix}, \Sigma = \begin{bmatrix} Var(A) & Cov(A, B) \\ Cov(A, B) & Var(B) \end{bmatrix}$$

Now, $Var(A) = Var(X + Y)$, and because $X$ and $Y$ are independent, $Var(A) = Var(X + Y) = \sigma^2_X + \sigma^2_Y$. Similarly, $Var(B) = Var(Y + Z) = \sigma^2_Y + \sigma^2_Z$. Also, $Cov(A, B) = Cov(X + Y, Y + Z)$, but because $X$, $Y$, and $Z$ are independent, $Cov(A, B) = Cov(X + Y, Y + Z) = Cov(Y, Y) = \sigma^2_Y$. Therefore,

$$\mu = \begin{bmatrix} \mu_X + \mu_Y \\ \mu_Y + \mu_Z \end{bmatrix} = \begin{bmatrix} 9 \\ 11 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma^2_X + \sigma^2_Y & \sigma^2_Y \\ \sigma^2_Y & \sigma^2_Y + \sigma^2_Z \end{bmatrix} = \begin{bmatrix} 41 & 25 \\ 25 & 61 \end{bmatrix}$$

b. Suppose hundreds of thousands (that is, a sufficiently large number) of student scores on a 150-question exam are distributed according to the following random variable:
\[ R = \sum_{i=1}^{50} M_i + 0.5 \sum_{j=1}^{100} W_j \]  

(1)

Each of the \( M_i \) are independent and identically distributed (IID) Beta random variables—yes, the questions are scored on a continuous scale from 0 to 1—and the \( W_j \) are separate IID Beta random variables, where all \( W_j \) are independent of all \( M_i \). The Beta parameters are \( \alpha_M = 10, \beta_M = 2, \alpha_W = 8, \) and \( \beta_W = 4 \). If we sample 100 student scores \( R_1, \ldots, R_n \) IID according to the distribution of \( R \) above, what is the distribution of the sample mean \( \overline{R} \)?

\[
E[M_i] = \frac{\alpha_M}{\alpha_M + \beta_M} = 0.83333 \\
E[W_i] = \frac{\alpha_W}{\alpha_W + \beta_W} = 0.66667 \\
Var(M_i) = \frac{\alpha_M \beta_M}{(\alpha_M + \beta_M)^2 (\alpha_M + \beta_M + 1)} = 0.01068 \\
Var(W_i) = \frac{\alpha_W \beta_W}{(\alpha_W + \beta_W)^2 (\alpha_W + \beta_W + 1)} = 0.01709
\]

We can compute \( R \)'s expectation using linearity of expectation. Because \( R \) is a sum of independent RVs, we can compute \( R \)'s variance by summing up the variance of the independent \( M_i \) and \( W_i \)'s as below:

\[
E[R] = 50 E[M_i] + 0.5 \cdot 100 E[W_i] = 75 \\
Var(R) = 50 \ Var(M_i) + 0.25 \cdot 100 \ Var(W_i) = 0.961
\]

As an aside, \( R \) can be approximated as \( R \sim N(75, 0.961) \), since the sums of both question types \( M_i \) and \( W_i \) respectively approach Normal distributions according to the Central Limit Theorem, and the sum of independent Normal distributions is itself a Normal distribution.

The distribution of the sample mean \( \overline{R} \) is then given by:

\[
\overline{R} = \frac{1}{100} \sum_{i=1}^{100} R_i \sim N(75, \frac{1}{100} 0.961) \sim N(75, 0.0096)
\]

### 2.3 Timing Attacks

In this problem we are going to show you how to crack a password in linear time, by measuring how long the password check takes to execute (see code below). Assume that our server takes \( T \) ms to execute any line in the code where \( T \sim N(\mu = 5, \sigma^2 = 0.5) \) ms. The amount of time taken to execute a line is always independent of other values of \( T \).

```python
# An insecure string comparison
def string_equals(guess, password):
    n_guess = len(guess)
    n_password = len(password)
```
On our site all passwords are length 5 through 10 (inclusive) and are composed of lower case letters only. A hacker is trying to crack the root password which is “gobayes” by carefully measuring how long we take to tell them that her guesses are incorrect.

a. What is the distribution of time that it takes our server to execute \( k \) lines of code? Recall that each line independently takes \( T \sim N(\mu = 5, \sigma^2 = 0.5) \) ms.

Let \( Y \) be the amount of time to execute \( k \) lines. \( Y = \sum_{i=1}^{k} X_i \) where \( X_i \) is the amount of time to execute line \( i \). \( X_i \sim N(\mu = 5, \sigma^2 = 0.5) \).

Since \( Y \) is the sum of independent normals:

\[
Y \sim N(\mu = \sum_{i=1}^{k} 5, \sigma^2 = \sum_{i=1}^{k} 0.5) \\
\sim N(\mu = 5k, \sigma^2 = 0.5k)
\]

b. First the hacker needs to find out the length of the password. What is the probability that the time taken to test a guess of correct length (server executes 6 lines) is longer than the time taken to test a guess of an incorrect length (server executes 4 lines)? Assume that the first letter of the guess does not match the first letter of the password. Hint: \( P(A > B) \) is the same as \( P(A - B > 0) \).

From last problem:
Time to run 6 lines of code \( A \sim N(\mu = 30, \sigma^2 = 3) \)
Time to run 4 lines of code \( B \sim N(\mu = 20, \sigma^2 = 2) \)

\[-B \sim N(\mu = -20, \sigma^2 = 2) \]
\[A - B \sim N(\mu = 10, \sigma^2 = 5) \]

\[
P(A > B) = P(A - B > 0) \\
= 1 - F_{A-B}(0) \\
= 1 - \Phi\left(\frac{0 - 10}{\sqrt{5}}\right) \\
\approx 1.0
\]
c. Now that our hacker knows the length of the password, to get the actual string she is going to try and figure out each letter one at a time, starting with the first letter. The hacker tries the string “aaaaaaa” and it takes 27ms. Based on this timing, how much more probable is it that first character did not match (server executes 6 lines) than the first character did match (server executes 8 lines)? Assume that all letters in the alphabet are equally likely to be the first letter.

Let \( M \) be the event that the first letter matched.

\[
\frac{P(M^C|T = 27)}{P(M|T = 27)} = \frac{f(T = 27|M^C)P(M^C)}{f(T = 27|M)P(M)} = \frac{\frac{f(T = 27|M^C)}{\frac{25}{26}}}{\frac{f(T = 27|M)}{\frac{1}{26}}} = 25 \cdot \frac{f(T = 27|M^C)}{f(T = 27|M)} = 25 \cdot \frac{\frac{1}{\sqrt{6\pi}}e^{-\frac{(27-30)^2}{6}}}{\frac{1}{\sqrt{8\pi}}e^{-\frac{(27-40)^2}{8}}} = 25 \cdot \frac{\sqrt{8}}{\sqrt{6}} \cdot \frac{e^{-\frac{9}{8}}}{e^{-\frac{169}{8}}} = \approx 9.6 \text{ million}
\]

d. If it takes the hacker 6 guesses to find the length of the password, and 26 guesses per letter to crack the password string, how many attempts does she need to crack our password, “gobayes”? Yikes!

\[7 \cdot 26 + 6 = 188\]

2.4 Naïve Bayes

Suppose we observe two discrete input variables \( X_1 \) and \( X_2 \) and want to predict a single binary output variable \( Y \) (which can have values 0 or 1). We know that the functional forms for the input variables are \((X_1|Y = 0) \sim \text{Poi}(\lambda_0), (X_1|Y = 1) \sim \text{Poi}(\lambda_1), (X_2|Y = 0) \sim \text{Ber}(p_0), \text{and } (X_2|Y = 1) \sim \text{Ber}(p_1)\), but we don’t know the optimal values of the parameters. We are, however, given a dataset of 9 training instances (shown at right.)

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y )</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
a. Use Maximum Likelihood Estimation to estimate the parameters $\lambda_0$, $p_0$, $\lambda_1$, and $p_1$.

\[
\begin{align*}
\lambda_0 &= \frac{1}{4} (1 + 3 + 7 + 9) = \frac{20}{4} = 5 \\
\lambda_1 &= \frac{1}{5} (3 + 5 + 5 + 5 + 7) = \frac{25}{5} = 5 \\
p_0 &= \frac{1}{4} (1 + 0 + 1 + 0) = \frac{1}{2} \\
p_1 &= \frac{1}{5} (1 + 0 + 1 + 1 + 1) = \frac{4}{5}
\end{align*}
\]

b. Use Maximum Likelihood Estimation to estimate the parameter $p_y$ for $Y \sim Ber(p_y)$.

\[P(Y = 1) = \frac{5}{9}.\]

c. You observe the following testing instance: $(X_1, X_2) = (2, 0)$. Using the Naive Bayes assumption, predict the output $Y$ for the testing instance. For this problem, showing how you computed your prediction is worth more points than the final answer.

We predict $Y = 0$ if the following Naive Bayes inequality holds:

\[
P(Y = 1)P(X_1 = 2|Y = 1)P(X_2 = 0|Y = 1) < P(Y = 0)P(X_1 = 2|Y = 0)P(X_2 = 0|Y = 0)
\]

\[
\frac{5}{9} \left( \frac{\lambda_1^2}{2!} e^{-\lambda_1} \right) \left( 1 - \frac{4}{5} \right) < \frac{4}{9} \left( \frac{\lambda_0^2}{2!} e^{-\lambda_0} \right) \left( 1 - \frac{1}{2} \right)
\]

\[
\frac{5}{9} \left( \frac{5^2}{2!} e^{-5} \right) \left( \frac{1}{5} \right) < \frac{4}{9} \left( \frac{5^2}{2!} e^{-5} \right) \frac{1}{2}
\]

\[
\frac{5}{9} \cdot \frac{1}{5} < \frac{4}{9} \cdot \frac{1}{2}
\]

\[
\frac{1}{9} < \frac{2}{9}
\]

Since the last inequality is true, that means the first inequality was true, so we predict $Y = 0$. 

2.5 **Logistic regression**

Suppose you have trained a logistic regression classifier that accepts as input a data point \((x_1, x_2)\) and predicts a class label \(\hat{Y}\). The parameters of the model are \((\theta_0, \theta_1, \theta_2) = (2, 2, -1)\). On the axes, draw the decision boundary \(\theta^T x = 0\) and clearly mark which side of the boundary predicts \(\hat{Y} = 0\) and which side predicts \(\hat{Y} = 1\).

\[\theta^T x = 2 + 2x_1 - x_2 = 0\] because \(x_0 = 1\) by definition. The prediction is 1 when \(\theta^T x > 0\). For example, the origin \((x_1, x_2) = (0, 0)\) yields \(\theta^T x = 2\), which gives us the prediction \(\hat{Y} = 1\).

See the graph above, to the right of the original.

2.6 **The Most Important Features**

Let’s explore saliency, a measure of how important a feature is for classification. We define the saliency of the \(i\)th input feature for a given example \((x, y)\) to be the absolute value of the partial derivative of the log likelihood of the sample prediction, with respect to that input feature \(|\frac{\partial LL}{\partial x_i}|\). In the images below, we show both input images and the corresponding saliency of the input features (in this case, input features are pixels):
First consider a trained logistic regression classifier with weights $\theta$. Like the logistic regression classifier that you wrote in your homework it predicts binary class labels. In this question we allow the values of $x$ to be real numbers, which doesn’t change the algorithm (neither training nor testing).

a. What is the Log Likelihood of a single training example $(x, y)$ for a logistic regression classifier?

$$LL(\theta) = y \cdot \log \sigma(\theta^T \cdot x) + (1 - y) \log [1 - \sigma(\theta^T \cdot x)]$$

b. Calculate is the saliency of a single feature ($x_i$) in a training example $(x, y)$.

We can calculate the saliency for a single feature as follows.

$$LL(\theta) = y \log z + (1 - y) \log (1 - z) \quad \text{where } z = \sigma(\theta^T \cdot x)$$

$$\frac{\partial LL}{\partial x_i} = \frac{\partial LL}{\partial z} \cdot \frac{\partial z}{\partial x_i} \quad \text{chain rule}$$

$$= \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right) \cdot \left(z(1 - z)\theta_i\right) \quad \text{partial derivatives}$$

saliency $= \left| \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right)z(1 - z)\theta_i\right|$

Show that the ratio of saliency for features $i$ and $j$ is the ratio of the absolute value of their weights $\frac{|\theta_i|}{|\theta_j|}$.

We can take the ratio as follows using our expression above.

saliency for feature $i$, $S_i = \left| \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right)z(1 - z)\theta_i\right|$, and same for $S_j$

$$\frac{S_i}{S_j} = \left| \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right)z(1 - z)\theta_i\right| \cdot \left| \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right)z(1 - z)\theta_j\right|^{-1} = \frac{S_i}{S_j} = \left| \frac{\theta_i}{\theta_j}\right| \quad \text{by elimination}$$

3 Ethics and Beta Distribution

While there won’t be any ethics material on the final exam, we’re including a problem that will not only exercise some probability, but hopefully provoke you to begin thinking about the impact that probability- and data-driven decisions have on society.
The Economist used a beta distribution to forecast results for the 2020 U.S. presidential election.*

Figure 1: Updated prediction of Democratic vote share is "Posterior" prediction.

1. Why is the beta distribution appropriate for modeling a presidential election?
2. Read the polling report published by The Economist. What should be considered when using this model and releasing its election predictions?

1. Several features of the beta distribution map onto election modeling:
   - The beta requires number of successes and number of failures as parameters. These are easy to acquire for elections since they are the counts of the target candidate polling positively vs. negatively.
   - The beta distribution can be used to model quantities representing fractions or percentages, since it is a continuous random variable with a support of \([0, 1]\). Election outcomes are typically reported in terms of percentage vote share, which naturally lends itself to a beta.
   - Election results are highly variable and betas allow us to incorporate uncertainty!
   - Election predictions require priors and new data, especially as the election day approaches. In the Economist’s model, the expected distribution of potential vote shares in each state was used as the prior, and the state polls that trickled in during the course of the campaign were the "new data".

2. Possible answers:

   **Limitations**
   - The beta approach is bad at modeling multiparty systems because success/failure can’t split up vote shares per candidate unless it is a two-party system.

• Random drift causes uncertainty around the current polling average.

• In states that are heavily polled late in the race, the model will pay little attention to its prior forecast; conversely, it will emphasise the prior early in the race or in thinly-polled states.

• Different methods of turnout projection can produce a bias.

• *Partisan non-response bias:* The probability that a poll respondent will agree to participate in a survey varies in response to media coverage. When there is unusually bad news about a candidate, their supporters are not in the mood to tell pollsters what they think – even though their ultimate voting intention has not changed. This causes the other candidate’s vote share to be over-represented in polls.

**Ethical Considerations**

• Poll results influence how people will vote. When the public believes a candidate is extremely likely to win, some people are less likely to vote. This is why some analysts think election polls should be covered using margins of error rather than speculative win probabilities.

• Poll results act as a feedback mechanism that affect parties’ policy choices.