# CS109A Week 10 Notes (Final Practice)

Ian Tullis

May 31, 2022

One more set of problems to practice together before the exam! I realize that no one's schedule is exactly replete with free time in Weeks 10-11, but also consider going back to review problems in the 109A notes that we didn't get to (and even those we did get to!)

## Problem 1: Convoluted arguments

Suppose that $X$ and $X'$ are independent random variables, each supported on $[0, 1]$, each with the PDF $f(X = x) = 2x$. Let $Y$ be the sum of $X$ and $X'$. Suppose that we want to find $P(Y \leq 1)$.

(a) Explain why the following mostly correct argument is ultimately incorrect.

We can write the convolution

$$\int_0^y f(X = x) f(X' = y - x) dx$$

which is

$$\int_0^y (2x) \cdot 2(y - x) = 4 \int_0^y xy - x^2 = 4 \cdot [\frac{1}{2}x^2 y - \frac{1}{3}x^3]_0^y = 4 \cdot (\frac{1}{2}y^3 - \frac{1}{3}y^3) = \frac{2}{3}y^3$$

So the answer is $\frac{2}{3}(1^3) = \frac{2}{3}$.

(b) Explain why the following mostly correct argument is ultimately incorrect.

$$P(Y \leq 1) = \int_0^1 \int_0^x f(X = x) f(X' = x') dx' dx = \int_0^1 \int_0^x 4xx' dx' dx = \int_0^1 2x \cdot x^2 dx = [\frac{1}{2}x^4]_0^1 = \frac{1}{2}$$

(c) Fix both arguments, and show that the fixed versions both yield the same answer.

(d) Suppose that we now want to find $P(Y \leq 1.5)$, and so we replace 1 with 1.5. Even the fixed versions of both arguments now fail, for the same reason. What is that reason?

(e) How would you get the overall PDF of $Y$ (over the entire supported range $[0, 2]$)?

# Solutions to Problem 1

(a) The setup and calculation of the convolution are correct, and $\frac{2}{3}y^3$ is the correct PDF of the new distribution, but then the argument uses it as if it were a CDF. That is, the method accidentally finds $f(1)$ rather than the $F(1)$ that the problem asks for.

(b) Now the issue is that the bounds of integration are set up incorrectly. We want to restrict to all $x, x'$ such that $x + x' \leq 1$, but the bounds are instead forcing $x'$ to be less than $x$, which is not what we want.

(c) To fix the first argument, we need to integrate the PDF from 0 to 1. This is

$$\int_0^1 \frac{2y^3}{3} = [\frac{y^4}{6}]_0^1 = \frac{1}{6}$$

To fix the second argument, we change the inner bound[1] to require $x + x' \leq 1$, i.e., to force $x' \leq 1 - x$:

$$\int_0^1 \int_0^{1-x} (2x)(2x')dx'dx = \int_0^1 [2xx'^2]_0^{1-x} dx = \int_0^1 2x(1-x)^2 dx$$

$$= \int_0^1 (2x - 4x^2 + 2x^3)dx = \int_0^1 [x^2 - \frac{4}{3}x^3 + \frac{1}{2}x^4]_0^1 = 1 - \frac{4}{3} + \frac{1}{2} = \frac{1}{6}$$

Yay, the same answer! (I also triple-checked this by running a Python simulation.)

(d) The problem is that the original distributions are only defined on $[0, 1]$. When $y = 1.5$, the convolution $\int_0^{1.5} f(X = x)f(X' = 1.5-x)$ is implying that it is possible to have, e.g., the first random variable equal 0.3 and the second equal 1.2. We are using $2(1.5-0.3)$ there as the PDF of $X'$ evaluated at 1.2, but that value should actually be 0.

(e) Here we could fix the issue by recognizing that because each of $x$ and $x'$ can only range from 0 to 1, they each have to be between 0.5 and 1 if we want them to add up to exactly 1.5. Generalizing this for any $y$ between 1 and 2, we need each of $x$ and $x'$ to be in the range $[y - 1, 1]$:

$$\int_{y-1}^1 f(X = x)f(X' = y - x)dx = \int_{y-1}^1 (2x) \cdot 2(y - x)dx = [2x^2y - \frac{4x^3}{3}]_{y-1}^1$$
$$= 2y - \frac{4}{3} - 2(y-1)^2y + \frac{4(y-1)^3}{3} = -\frac{2}{3}(y^3 - 6y + 4)$$

where we used Wolfram Alpha for the last part. Notice that this looks like it's going to be a negative probability because of the leading minus sign, but $y^3 - 6y + 4$ is also

[1] Note that the two bounds can be written in the other order, too, as long as you properly match each integral with its $dx$ or $dx'$ term.

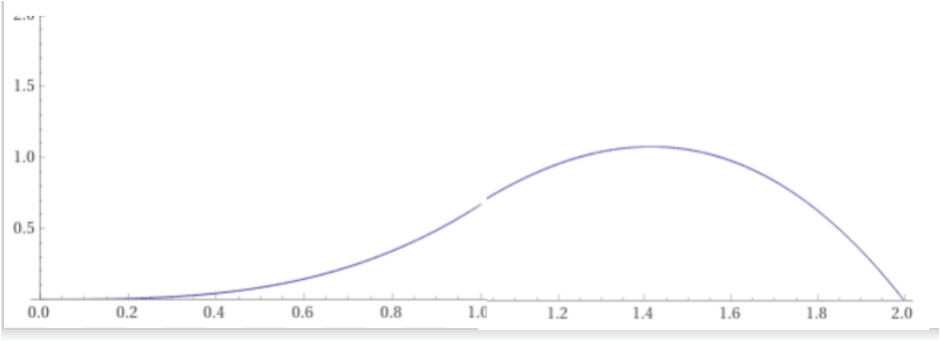
always nonpositive, so it works out.

Then an overall PDF is:

$$f(Y = y) = \begin{cases} \frac{2y^3}{3}, & \text{if } 0 \leq y \leq 1 \\ -\frac{2}{3}(y^3 - 6y + 4) & \text{if } 1 \leq y \leq 2 \end{cases} \tag{1}$$

As a quick check, does this properly integrate to 1? We know the first part integrates to $\frac{1}{6}$. For the second half, we are feeling lazy and Wolfram Alpha tells us that it indeed integrates (between 0 and 1) to $\frac{5}{6}$.

Here's what this bizarre function actually looks like. It's not actually discontinuous at $x = 1$... I just had to stitch two separate Wolfram Alpha plots together.



# Problem 2: Pretend that industry cares about grades

Suppose that a company is building a model that tries to predict success in industry (somewhat unrealistically using a binary variable $S$, where $S = 1$ means success) based on grades in $n$ Stanford CS courses $X_1, X_2, ..., X_n$ (using one multinomial variable per course, ignoring +/-, so A = 4, B = 3, C = 2, D = 1, NP = 0). The company uses complete data (grades + success) from 1000 students.

(a) Suppose that the model is like Naive Bayes but *without* the independence assumption, and estimates each individual parameter of the form $P(X_1 = x_1, ..., X_n = n|S = y)$ or $P(S = y)$ directly from the data. How many of these individual parameters would need to be estimated? Give your answer in terms of $n$.

(b) What is the major problem with the above approach, apart from having to estimate so many parameters? (Hint: Unless $n$ is very small, what will most of those estimated probabilities end up being?)

(c) Now suppose that the model *does* use the Naive Bayes assumption. How many individual parameters of the form $P(X_i = x_i|S = y)$ or $P(S = y)$ would need to be estimated? Give your answer in terms of $n$.

(d) Give one "real-world" reason why the Naive Bayes assumption might be unrealistic in this particular scenario.

(e) Show how you would estimate the parameter $P(X_1 = 3|Y = 1)$ from the data, *including Laplace smoothing.*

**Now, for the remaining parts, instead assume that the company uses a logistic regression model.** Before training, an extra "intercept" feature is added; it has the value 1 for every student.

(f) What undesirable consequences might arise if the model did not include this intercept feature? (How would its absence limit the expressiveness of the model?)

(g) Why would such an intercept feature be useless in a Naive Bayes model?

(h) Suppose that $n = 2$, and the vector of weights found by the logistic regression model is $[-2.1, 0.4, 0.5]$, with the weights corresponding to the intercept term, course 1, and course 2, respectively. Would the model predict success for a student with an A in course 1 and a B in course 2? (Justify your answer mathematically.)

(i) Suppose that $S$ is truly independent of e.g. $X_7$. What would you expect the corresponding weight for $X_7$ to be?

(j) Suppose that e.g. $X_{10}$ is accidentally an exact copy of $X_9$. Explain why in this case, a gradient-based method might give different sets of weights depending on the learning rate.

# Solutions to Problem 2

(a) We need $P(S = 0)$ and $P(S = 1)$. Once we have one of those from the data, we know the other, but then again, it's hard to imagine a way of processing the data where we couldn't have just as easily kept track of both the 0s and 1s at the same time.

For each of the $n$ courses, there are 5 possible grade values, and then each of those comes in two flavors: conditioned on $S = 1$, and conditioned on $S = 0$. So there are $2 \cdot 5^n$ such parameters. However, notice that if we know $5^n - 1$ out of the $5^n$ possible parameters conditioned on $S = 1$, for instance, we know the other one as well, because their values must sum to $P(S = 1)$. So we technically only need to estimate $5^n - 1$ parameters from each of the two $|S = 0$ and $|S = 1$ categories, but again, this trick is only of theoretical interest since we need to look at all the data.

In summary, there are $2 \cdot 5^n + 2$ parameters total (which is the important part), but we could find the right set of only $2 \cdot 5^n - 1$ of them and then derive the other 3 (which is less important).

(b) There are only 1000 students, so there will be many grade/success combinations that are not represented by any data point, resulting in a probability estimate of 0. This also means that estimates for new students with these previously-unknown grade/success combinations would come out as 0, which kinda defeats the purpose of making the model. Laplace smoothing would get rid of the 0s but would not address the model's poor ability to generalize.

(c) We'll leave out the discussion from (a) of which probabilities can be derived from which others. As before, we need $P(S = 0)$ and $P(S = 1)$. But now we only need parameters corresponding to grades in individual courses, not combinations of grades in all courses. Conditioning on $S = 1$, for example, we need $P(X_1 = 4|S = 1), P(X_1 = 3|S = 1), P(X_1 = 2|S = 1), P(X_1 = 1|S = 1), P(X_1 = 0|S = 1)$. (Again, once we have gotten four of these from the data, we technically know the fifth for free.)

So here there are $2 \cdot 5 \cdot n + 2 = 10n + 2$ probabilities in total.

(d) Naive Bayes assumes that – conditioning on success, for example – a student's grade in one class is independent of their grades in other classes. This doesn't seem very plausible, especially since, e.g., someone who does very well in one systems course might also do very well in other systems courses.

That is, let $X_1$ and $X_2$ be two related courses. Suppose the model learns that

- for class $X_1$, $P(X_1 = 4|S = 1) = 0.5$ and $P(X_1 = 3|S = 1) = 0.3$
- for class $X_2$, $P(X_1 = 4|S = 1) = 0.6$ and $P(X_1 = 3|S = 1) = 0.2$

But it's likely that in general (and even within the $S = 1$ cohort), people who get an A in the first class tend to be much more likely to get an A in the second class as well. This would violate the independence assumption. (Remember that this doesn't mean that Naive Bayes can't be used! We know it's naive – it's right in the name...)

(e) This is just like what you did in Problem 2 of Homework 6, except now there are 5 possible values that $X_1$ can take on, and the Laplace smoothing adds one "bonus" instance to each of them. So the denominator has a +5 instead of a +1.

$$P(X_1 = 3|Y = 1) = \frac{(\text{count of data points with } X_1 = 3, Y = 1) + 1}{(\text{count of data points with } Y = 1) + \mathbf{5}}$$

(f) A logistic model with no intercept term produces a decision boundary that *has* to go through the origin. This is usually an undesired restriction.

E.g., in the case of this model, suppose there are two courses. We want to draw a decision boundary (line, in this case) that separates the $S = 1$ points from the $S = 0$ points. But we are hampered in our ability to do so if that line must go through the origin, especially since most of the points from both categories will be far from the origin.

(g) If we added an extra feature $X_0$ to Naive Bayes that was 1 for everyone, it would just create a multiplicative $P(X_0 = 1|S = 1)$ or $P(X_0 = 1|S = 0)$ term that would always be 1, and so it would essentially not be there.

(h) The dot product of the weights vector and the feature vector is $[-2.1, 0.4, 0.5] \cdot [1, 4, 3] = 1.0$. Plugging this into the sigmoid, the predicted probability is

$\boxed{\frac{1}{1+e^{-1}}}$. This is greater than 0.5, so the model predicts success.

(i) If $S$ and $X_7$ are truly independent within the dataset, the model can derive no useful predictive information from $X_7$ and should in theory give it a weight of 0.

Unfortunately this isn't always the case in a more complex model, because of interactions with other features. Consider these data points:

| $S$ | $X_7$ | $X_8$ |
|---|---|---|
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 0 |

By inspection of the first two columns, $S$ and $X_7$ are independent. (Specifically, $P(S = s, X_7 = x) = P(S = s)P(X_7 = x)$ for all possible $(s, x)$ pairs.) However, when I use our HW 6 code or sites to fit this, I get nonzero weights for the $X_7$ term.

(j) Suppose that our fitting function assigns weights $w_9$ and $w_{10}$ to $X_9$ and $X_{10}$. Then observe that because $X_{10}$ is the same as $X_9$, one of many other equally good sets of weights would be $w_9 + w_{10}$ and 0 for $X_9$ and $X_{10}$. Depending on how our learning rate pushes us along the landscape, we might end up at different equally good optima. (In this case, the landscape is still convex, but not *strictly* convex.)

# Problem 3: Algorithmic fairness

Suppose that a certain protected demographic $D$ makes up 10% of the population. Within demographic $D$ (i.e. $D = 1$), 5% have a certain health condition ($H = 1$). In the remaining 90% of the population, only 1% have that health condition.

(The setup, and parts (a) and (b), are the same as in the Week 3 notes. But then the problem goes on to some new places!)

(a) For a randomly selected person, what is $P(D = 1|H = 1)$?

(b) Suppose we are trying to predict $H$. Consider a stupid model that, when given any new person, invariably predicts that $H = 0$. When applied to a representative sample of the population, how often will the stupid model be correct? (The takeaway is that impressive-looking accuracy numbers may mask serious problems!)

(c) Does the stupid model in (b) achieve **parity**? What about **calibration**? Explain.

(d) Now suppose that we try to incorporate additional features, including age. Suppose that the true distribution of age is $\mathcal{N}(\mu = 50, \sigma = 10)$ both within and outside of demographic $D$. So, for example, we would expect the mean age of a sample of 18 people (regardless of their demographic status) to be 50, with a variance of $(\frac{1}{18})^2(18\sigma^2) = \frac{\sigma^2}{18}$, and therefore a standard deviation of $\frac{\sigma}{\sqrt{18}} \approx 2.35$. (For this problem, do not worry about the $n - 1$ bias correction.)

Suppose that a sample of 10 people has only 1 from demographic $D$. To try to fight bias, the researchers create 8 more copies of that individual. (They copy the data point, not the actual person!) The mean age of this sample is still 50, but now what would you expect the standard deviation of that mean to be? (This illustrates one issue with trying to correct for bias in this way!)

# Solutions to Problem 3

(a) By Bayes' Rule, $P(D=1|H=1) = \frac{P(H=1|D=1)P(D=1)}{P(H=1)}$. We are given that $P(H=1|D=1) = 0.05$ and $P(D=1) = 0.1$. Using the Law of Total Probability for the denominator, we have $P(H=1) = P(H=1|D=1)P(D=1) + P(H=1|D=0)P(D=0)$, and we are told that $P(H=1|D=0) = 0.01$ and $P(D=0) = 0.9$. Putting this all together, we get $\frac{(0.05)(0.1)}{(0.05)(0.1)+(0.01)(0.9)} = \boxed{\frac{5}{14}}$.

Here's another less formal way to solve problems like this: say there are 1000 people in the population. Then 100 of them have $D=1$, and 5 of those have $H=1$. The remaining 900 people have $D=0$, and 9 of those have $H=1$. So there are 14 people with $H=1$, and 5 of them have $D=1$.

(b) The total fraction of people with $H=1$ is $(0.1)(0.05)+(0.9)(0.01) = 0.014$. Therefore the total fraction of people with $H=0$ is $1-0.014 = 0.986$. So the model will be correct 98.6% of the time, even though it is not even trying to do the thing it is supposed to do, and even though it is producing worse results for the demographic that presumably needs this disease detection the most.

(c) The stupid model achieves parity because the probability of a positive prediction ($H=1$) is the same even when conditioned on each group: $P(H=1|D=1) = P(H=1|D=0) = 0.986$. However, it does not achieve calibration, because its probability of a correct response ($T$) is different when conditioned on each group: $P(H=T|D=1) = 0.95$, and $P(H=T|D=0) = 0.99$.

(d) Let $X_1$ be a random variable corresponding to the copied person, and $X_2$ through $X_{10}$ be random variables corresponding to the other people. Each of these random variables has mean $\mu = 50$, $\sigma = 10$, and $\sigma^2 = 100$. Then the mean of the new artificial sample is distributed as $\frac{1}{18}(9X_1 + X_2 + ... + X_{10})$. (The $9X_1$ is *not* the same as $X_1 + ... + X_1$ nine times... make sure you understand why!) The parenthetical part has variance $81\sigma^2 + 9\sigma^2 = 90\sigma^2$, and then the variance of $\frac{1}{18}$ of that is $(\frac{1}{18})^2 90\sigma^2 = \frac{5}{18}\sigma^2 = \frac{500}{18}$. Accordingly, $\sigma = \sqrt{\frac{500}{18}} = \boxed{\frac{5\sqrt{10}}{3}} \approx 5.27$, which is much larger!

This is another one I didn't fully believe until I wrote the code:

```
from scipy.stats import norm
import numpy as np

vals = []
TRIALS = 1000000
for i in range(TRIALS):
    r = list(norm.rvs(loc=50, scale=10, size=18))
    vals.append(sum(r) / 18)
print("Original:", np.std(vals))
```

```
vals = []
for i in range(TRIALS):
    r = list(norm.rvs(loc=50, scale=10, size=10))
    r.extend([r[0]]*8)
    vals.append(sum(r) / 18)
print("With copied person:", np.std(vals))
```

# Miscellaneous Shorties

(a) Suppose that we define a random variable $X$ as follows: roll a fair 4-sided die, note the result $k$, then roll $k$ more fair 4-sided dice and add the results together. (Notice that the original roll is *not* counted in this total.) What is $P(X = 4)$?

(b) Suppose we are doing Naive Bayes (without Laplace smoothing) on a dataset with two (binary) features and a binary output. Even if the Naive Bayes assumption is correct, why is the following statement incorrect in general?

$$P(Y = 0|X_1 = 1, X_2 = 0) = 1 - P(X_1 = 1|Y = 1) \cdot P(X_2 = 0|Y = 1) \cdot P(Y = 1)$$

(c) As in the problem from Week 4, suppose that in Dungeons and Dragons, I determine my fighter character's hit points ($H_F$) by rolling ten 10-sided dice and adding them together.

(i) Suppose that my Dungeon Master allows me to reroll any 1s that come up when I roll my ten 10-sided dice, but only once each (i.e. even if a rerolled 1 comes up 1 again, I have to keep that 1). What is $\mathbb{E}(H_F)$ in this case?

(ii) Suppose that my Dungeon Master allows me to reroll any 1s that come up when I roll my ten 10-sided dice, and then keep rerolling any 1s, and so on, until there are no more 1s. What is $\mathbb{E}(H_F)$ in this case?

(d) You are writing a game app that, when run, can randomly output one of eight strings: `000`, `001`, `010`, `011`, `100`, `101`, `110`, `111`. You can decide on the probability of each of the eight strings being produced (and it is OK for one or more of these probabilities to be 0), as long as the probabilities satisfy the rules for a PMF over these eight distinct outcomes.

Let $A$ be the event that the first character of the outputted string is 1, $B$ be the event that the second character of the outputted string is 1, and $C$ be the event that the third character of the outputted string is 1. Provide any valid PMF for the eight strings such that $A$ is **not** independent of $B$, but $A$ and $B$ **are** independent when conditioning on $C$. (Note: you just need to provide a list of 8 values; you do not need to write a program!)

# Solutions to Miscellaneous Shorties

(a) This is best broken into cases according to the first roll, i.e., whether we roll 1, 2, 3, or 4 dice.

- 1 die: $P(X = 4) = \frac{1}{4}$.
- 2 dice: Call the two dice $A$ and $B$. Then $P(X = 4) = P(A = 1, B = 3) + P(A = 2, B = 2) + P(A = 3, B = 1) = \frac{1}{4} \cdot \frac{1}{4} + \frac{1}{4} \cdot \frac{1}{4} + \frac{1}{4} \cdot \frac{1}{4} = \frac{3}{16}$
- 3 dice: Here we need one of the dice to be 2 and the other two to be 1. One way to do this is to observe that there are $4^3 = 64$ outcomes total (sample space), and only 3 desired ones (event space), namely 211, 121, 112. So this probability is $\frac{3}{64}$.
- 4 dice: In this case all four have to come up 1, and the probability of this is $(\frac{1}{4})^4 = \frac{1}{256}$.

Since each outcome of the first roll is equally likely, the overall probability is

$$\frac{1}{4} \cdot \frac{1}{4} + \frac{1}{4} \cdot \frac{3}{16} + \frac{1}{4} \cdot \frac{3}{64} + \frac{1}{4} \cdot \frac{1}{256} = \boxed{\frac{125}{1024}}$$

Hmm, the fact that this is $\frac{5^3}{2^{10}}$ makes me think there was maybe some other clever way to get this answer! Combinatorics is full of tantalizing paths...

(b) The statement is incorrect because we are trying to apply the Law of Total Probability here, but these are likelihoods, not probabilities. Likelihoods come from products of PDFs (and a denominator that we often ignore), and they are not restricted to being between 0 and 1. So you can't use a 1 minus trick – you have to calculate the likelihoods for the two options ($Y = 1$ and $Y = 0$) separately.

(c) (i) Let's start by thinking about just one die roll. $\frac{9}{10}$ of the time, we keep the original value on the die, and the other $\frac{1}{10}$ of the time, it is like a single new standard roll. Therefore the expected value of the single die is $\frac{1}{10}(2) + ... + \frac{1}{10}(10) + \frac{1}{10}(\mathbb{E}(R)) = \frac{54}{10} + \frac{1}{10}(\frac{11}{2}) = \frac{119}{20}$. Then $\mathbb{E}(H_F) = \boxed{\frac{119}{2}} = 59.5$. This is a noticeable improvement over the mean of 55 without the DM's generosity.

(ii) Now when we get a 1, it is like we are beginning the entire rolling process over for that die, so we can write a recursive expression for a single die: $\mathbb{E}(R) = \frac{54}{10} + \frac{1}{10}\mathbb{E}(R)$. Solving for $\mathbb{E}(R)$, we find that it equals 6, so $\mathbb{E}(H_F) = \boxed{60}$. Allowing those infinite rerolls doesn't get us much more than the single reroll in part (d); intuitively, this is because multiple rerolls are so rare.

(d) One of many possible acceptable PMFs is as follows:

- $P(\texttt{000}) = 0.0$, $P(\texttt{010}) = 0.0$, $P(\texttt{100}) = 0.0$, $P(\texttt{110}) = 0.2$

- $P(\texttt{001}) = 0.2$, $P(\texttt{011}) = 0.2$, $P(\texttt{101}) = 0.2$, $P(\texttt{111}) = 0.2$

We see that $P(A) = 0.6, P(B) = 0.6, P(A, B = 0.4)$, and $0.6 \cdot 0.6 \neq 0.4$. So $A$ and $B$ are not independent.

However, we also see (from the bottom set of four probabilities) that $P(A|C) = \frac{P(A,C)}{P(C)} = \frac{0.4}{0.8} = 0.5$, $P(B|C) = 0.5, P(A, B|C) = 0.25$. Since $0.5 \cdot 0.5 = 0.25$, $A$ and $B$ are conditionally independent given $C$.

Notice that assigning probability mass to $P(A, B|C^c)$ but not to $P(A, B^c|C^c), P(A^c, B|C^c), P(A^c, B^c|C$ kind of "links" $A$ and $B$ together, and is one way to get them to be non-independent.