CS109A Week 2 Notes

Ian Tullis

January 11, 2022

I. Combinatorics of Entry Codes

When I emailed out entry codes for the class, some of you got additional comments from me about numerically interesting properties of your particular codes. (CS109A is truly full-service!) The following practice problems are based on those properties...

Problem 1. The entry codes are six digits and can begin with one or more leading zeroes.[1] Examples include 379009, 012345, 999999, and 000000.

(a) How many possible entry codes are there?

(b) How many possible entry codes consist of the same two digits repeated three times back-to-back, like 121212 or 333333?

(c) How many possible entry codes consist of the digits 1 through 6, once each, in some order, like 254631 or 123456?

(d) How many possible entry codes contain exactly three of some digit in a row, like 864443, 212221, or 000777?

(e) How many possible entry codes include a 5-digit palindrome (a string that reads the same forward and backward), like 123217 or 657775?

(f) Suppose that the Stanford course admins get tired of making up and checking lists of entry codes. They decide instead that a code is valid if and only if its digits sum to **exactly 9**. For example, 009000 and 122121 are valid codes, whereas 330000, 999999, and 963749 are not.

Now suppose a student (who is unaware of this system) picks a code uniformly at random and enters it. What is the probability that their code is valid?

(g) How would you solve the problem in part (f) if the target sum were 10 instead of 9?

[1] Some of you actually got five-digit codes without a leading 0, but just for simplicity, we'll pretend there was one.

Solutions to Problem 1.

(a) Each digit of the code can be any of the 10 digits from 0 to 9, and these choices do not depend on each other at all, so the answer is $\boxed{10^6 = 1000000}$. Another way to see this is that any code from 000000 to 999999 is valid, and these are the nonnegative integers that are less than a million.

(b) Here, we get to choose any two-digit string, but then we have to repeat that string three times. So our only choice is which two-digit string to use, and there are $\boxed{10^2 = 100}$ choices for that.

(c) Each valid code here corresponds to one permutation (ordering) of the digits 1 through 6. There are 6 choices for which digit to put first, 5 choices for which of the remaining digits to put second, and so on, so the answer is $\boxed{6! = 720}$.

(d) A good way to approach this problem is to enumerate the possible patterns that are valid, and figure out the number of each type. We specifically need there to be *exactly* three of a digit in a row, so we must make sure that any adjacent digits are different.

- Three of one digit D in the first three slots, followed by some digit N that is not D, followed by two of any digit (we will use ? for these to clarify that they are not necessarily even different from each other). There are 10 choices for D, 9 choices for N, and then 10 choices for each of the two ?s, for a total of $10 \cdot 9 \cdot 10 \cdot 10 = 9000$. (Once we pick D, we use it for all three of the first slots, which is why we really only have 10 choices there, not 10^3.)[2]
- As above, but with the three identical digits D in the second through fourth slots, with the first and fifth slots N and N' not equal to D (but potentially equal to each other), and the sixth slot ?. There are 10 choices for D, 9 choices for N, 9 choices for N', and 10 choices for ?, for a total of $10 \cdot 9 \cdot 9 \cdot 10 = 8100$.
- As above, but with the three identical digits in the third through fifth slots. By symmetry, this is the same as the previous case, so there are 8100 more possibilities.
- Finally, what if the three identical digits are in the fourth through sixth slots? By symmetry, this is the same as our first case, so there are 9000 more possibilities.

We also need to check for the cardinal sin of combinatorics: double-counting. Are the four situations above mutually exclusive? If we think about it for a while, we notice that there is one overlap: a code like

[2] What if we had instead started by saying there were 10 choices for N? Then there are 9 choices for D, but – whew – we get the same answer. It's a good idea to try this kind of thought exercise occasionally... if it changes your answer, then something is wrong!

000777, with *two* runs of three identical digits, is counted in both the first and fourth cases. How many problematic codes like this (of the form $DDDddd$) are there? We have 10 choices for D and 9 choices for d, so there are 90. So we can fix our double-counting problem by subtracting 90 from the total.

Therefore the answer is $2(9000) + 2(8100) - 90 = \boxed{34110}$. You may feel a little uncertain about this – what if we missed some other instance of double-counting? This is a great example of an answer that can be easily checked with Python, as we will show later.

(e) In this case the codes can be in one of two forms: $ABCBAX$, or $XABCBA$. In either case, we have total freedom to choose A, B, C, and X, and there are $10 \cdot 10 \cdot 10 \cdot 10 = 10000$ ways to do this. So the answer is 20000... but wait, are those patterns really mutually exclusive?

Curses, a single code might contain *two* five-digit palindromes. What are the constraints there? Let $ABCDEF$ represent the six digits.

- Since the first five slots form a palindrome, we know $E = A$ and $D = B$. So now the form is $ABCBAF$.
- Since the second five slots form a palindrome, we know $F = B$ and $C = A$. So now the form is $ABABAB$.

There are $10 \cdot 10$ ways to choose A and B (notice that they can be the same!), so this gives us 100 codes that we have double-counted. Therefore the answer is actually $20000 - 100 = \boxed{19900}$.

(f) Here we can take advantage of the divider method! Creating a valid code here is like adding 9 marbles to 6 buckets, and then turning the number of marbles in each bucket into a digit. For example, the following way of distributing the marbles is like the code 101340. (Here, each $|$ represents a divider between buckets, and each $\circ$ represents a marble. We leave off the leftmost and rightmost $|$s since they are implied. Notice that two of the buckets are empty: the second and the sixth.)

$\circ \,||\, \circ \,|\, \circ \circ \circ \,|\, \circ \circ \circ \circ \,|$

We have $n = 9$ things to distribute among $k = 6$ buckets, so by the divider method formula, the number of valid codes is $\binom{n+k-1}{k-1} = \binom{9+6-1}{6-1} = \binom{14}{5} = \boxed{2002}$.

Remark. It's good to be able to rederive the divider method formula! In the example above, notice that we have a string of symbols, each of which is $|$ or $\circ$. There must be exactly n $\circ$s since that is the number of things we are distributing. There must be exactly $k-1$ $|$s, since there are k buckets, which are delineated by $k-1$ inner divisions. Therefore there are $n+k-1$ symbols in total. To create such a string, we choose locations for the $k-1$ | symbols; per the usual expression for choosing, this is $\binom{n+k-1}{k-1}$. Notice that we could have instead chosen locations for the n $\circ$ symbols; then we would have $\binom{n+k-1}{n}$, which is an equivalent form of the divider method formula. Just make sure you don't accidentally use $\binom{n+k-1}{n-1}$...

(g) Unfortunately, for this scenario, we can only directly use the divider method when the target sum is 9 or less. What goes wrong when we have a target sum of, e.g., 10 instead? Then the divider method starts allowing situations like this, where 10 or more marbles end up in the same bucket:

$|| \circ \circ \circ \circ \circ \circ \circ \circ \circ \circ |||$

But then our correspondence between marbles/buckets and digits breaks down! There is no digit that represents having 10 marbles in one bucket. Digits here have to be between 0 and 9.

We can hack our way around this by explicitly excluding all cases in which all 10 marbles end up in one bucket. Because there are 6 buckets, there are 6 ways for this to happen. So the answer is the divider method answer minus the invalid scenarios: $\binom{10+6-1}{6-1} = \binom{15}{5} = 3003$, minus 6, for $\boxed{2997}$.

We could similarly hack the divider method for a target sum of 11, but then, in addition to cases in which all 11 marbles are in one bucket, we would need to consider cases in which 10 marbles are in one bucket and 1 marble is in another. So the divider method would get harder and harder to correct for larger target sums!

II. Python Interlude

By now you have seen that it is easy to get combinatorics problems slightly wrong, or to be nervous even when you have the right answer. Fortunately, we are computer scientists, i.e., we can just write some brute-force code to reassure ourselves!

In CS109, you will use the `numpy` and `scipy` packages, but you should also be aware of a hero among the built-in libraries: `itertools`. Specifically, the following functions are so, so useful:

`itertools.permutations(ls)` returns a generator with all permutations of a list `ls`. Example:

```
>>> print(list(itertools.permutations([1, 3, 7])))

[(1, 3, 7), (1, 7, 3), (3, 1, 7), (3, 7, 1), (7, 1, 3), (7, 3, 1)]
```

The output of `itertools.permutations` is a generator, not a list, so you can't do things like `itertools.permutations[0]` to get the first permutation in the list. If you want to see the whole list, you can do what I've done above, but usually this is not what you want. Python gets very slow when it has to store and deal with large lists, and the whole point of a generator is to save space by spitting out each result on demand, as it is needed. So usually you will want to do something like this:

```
# How many permutations of 1 through n have a 3 next to a 4?

import itertools

total = 0
N = 10

for p in itertools.permutations(range(1, N+1)):
  i3 = p.index(3)
  i4 = p.index(4)
  if abs(i4-i3) == 1:
    total += 1

print(total)
```

This returns 725760.[3]

Be advised: because we are still explicitly enumerating all possible permutations, this method will be intractably slow on problems involving, e.g., 20!. If you need just the values of factorials, you can `import math` and then use `math.factorial(20)`.

`itertools.combinations(ls, n)` returns a generator with all combinations of size `n` of a list `ls`.

```
>>> print(list(itertools.combinations([1, 3, 7], 2)))

[(1, 3), (1, 7), (3, 7)]
```

`itertools.product(ls, repeat=n)` returns a generator with all lists of length `n` that can be made from the elements of a list `ls`, "without replacement".

```
>>> print(list(itertools.product([1, 3, 7], repeat=2)))

[(1, 1), (1, 3), (1, 7), (3, 1), (3, 3), (3, 7), (7, 1), (7, 3), (7, 7)]
```

[3] As practice, can you see how to find this result directly? Consider: where in the permutation can the 3 end up? For each of those situations, where can the 4 go so it is next to the 3? How many ways are there to fill in the remaining slots? To help you check your work, 725760 is $18 \cdot 8!$...

Let's use `itertools.product` to check our answer to problem 1(d), which was annoying and error-prone. Admittedly, it can be easy to mess up the details of the code as well; it checks that the maximum-length run of a single digit is exactly 3. The program reassuringly prints the desired value 34110.

```
import itertools

total = 0

for p in itertools.product(range(10), repeat=6):
  max_run = 1
  curr_run = 1
  curr_target = p[0]
  for i in range(1, 6):
    if p[i] == curr_target:
      curr_run += 1
    else:
      max_run = max(max_run, curr_run)
      curr_target = p[i]. # reset what we're looking for
      curr_run = 1
  max_run = max(max_run, curr_run). # make sure to check our final run
  if max_run == 3:
    total += 1

print(total)
```

The palindrome problem, 1(e), is much easier to check:

```
total = 0
for p in itertools.product(range(10), repeat=6):
  if (p[0] == p[4] and p[1] == p[3]) or (p[1] == p[5] and p[2] == p[4]):
    total += 1
print(total)
```

And the divider method problem, 1(f), is easier still:

```
total = 0
for p in itertools.product(range(10), repeat=6):
  if sum(p) == 9:
    total += 1
print(total)
```

As a reminder, though, unless a homework problem tells you that you can use simulation or brute force code, you are still expected to come up with a mathematical explanation.

III. Return to the Casino

At the beginning of Week 1, we found ourselves playing a weird game in a casino:

- We have two tokens. We place each of our two tokens on a number. They can both be on the same number if we want.

2	3	4	5	6	7	8	9	10	11	12

- The dealer rolls the dice and computes their sum. If we have at least one token on that number, the dealer removes **one** token and gives us $1000. In any case, any other tokens remain where they are.

- The dealer again rolls the dice and computes their sum. If we have at least one token on that number, the dealer removes one token and gives us $1000.

Therefore, we can win up to $2000, but we want to place our tokens in a way that guarantees the largest *expected* return, i.e., does the best on average. We might still get unlucky and win $0, but we can still be smug that we played optimally![4]

We considered the probabilities for the possible sums of each roll of the dealer's two dice:

2	3	4	5	6	7	8	9	10	11	12
$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

Since 7 is the most likely outcome, we put our two tokens on 7. Then we were disappointed and confused when we learned that it is actually better to, for example, put one token on 2 and one token on 7. Today we'll back this up with math. First, let's consider the strategy of putting both of our tokens on 7.

Problem 2: The 7, 7 strategy.

(a) What is the probability that we win $2000, i.e., that the dealer rolls 7 twice?

(b) What is the probability that we win nothing, i.e., that the dealer does not roll 7 in either round?

(c) What is the probability that we win $1000, i.e., that the dealer rolls 7 in exactly one of the two rounds?

(d) Using the above information, what are our expected (i.e. average) winnings?

[4] But is it necessarily true that winning the most on average is optimal? Would our strategy be different if we wanted to maximize our chances of winning at least once? e.g. we owe someone at the casino $1000, and if we don't pay up, we might be invited into a back room by a nice man with a hammer, as in the movie *Casino*.

Solutions to Problem 2.

(a) We already found that the probability of the dealer rolling 7 in any given round is $\frac{6}{36} = \frac{1}{6}$. Then the probability of this happening in both rounds is $\frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$, because these events are independent. (Informally, the outcome of the second roll is not at all influenced by the outcome of the first roll.)

(b) In a given round, the probability of the dealer failing to roll a 7 is $1 - \frac{1}{6} = \frac{5}{6}$. Therefore the probability of this happening twice is $\frac{5}{6} \cdot \frac{5}{6} = \frac{25}{36}$.

(c) Notice that the following three events are mutually exclusive and exhaustive:

- We win neither of the two rounds.
- We win one of the two rounds (either the first or the second).
- We win both of the two rounds.

Therefore, using our answers to (a) and (b), the probability of winning one of the two rounds is $1 - \frac{25}{36} - \frac{1}{36} = \frac{10}{36} = \frac{5}{18}$.

What if we wanted to compute this answer more directly? Well, one of two mutually exclusive things can happen:

- We win the first round and lose the second round. Since these events are independent, this happens with probability $\frac{1}{6} \cdot \frac{5}{6} = \frac{5}{36}$.
- We lose the first round and win the second round. Here the probability is also $\frac{5}{6} \cdot \frac{1}{6} = \frac{5}{36}$.

Therefore the answer is $\frac{5}{36} + \frac{5}{36} = \frac{10}{36}$, as before.

(d) Using the results of (a)-(c), our expected winnings are $(0)(\frac{25}{36}) + (1000)(\frac{10}{36}) + (2000)(\frac{1}{36}) = \frac{12000}{36} = \$333.\overline{33}$.

Now let's repeat the exercise for the strategy of putting one token on 2 and one token on 7. Beware: there are some differences in the calculations here!

Problem 3: The 2, 7 strategy.

(a) What is the probability that we win \$2000?

(b) What is the probability that we win nothing?

(c) What is the probability that we win \$1000?

(d) Using the above information, what are our expected winnings?

(e) WTF, how in the world could this be better than the 7, 7 strategy?

(f) What is the best possible token placement strategy? (You don't have to prove it.)

(g) How would your strategy change if you had 36 tokens and the game ran for 36 rounds? (You don't have to prove it.)

Solutions to Problem 3.

(a) There are two ways to win twice: either the dealer rolls 2 and then 7, or 7 and then 2. Consulting our table on page 7, the probability of a roll of 2 is a mere $\frac{1}{36}$. So the probabilities of these two mutually exclusive outcomes are $\frac{1}{36} \cdot \frac{1}{6}$ and $\frac{1}{6} \cdot \frac{1}{36}$, for a total of $\frac{2}{216} = \frac{1}{108}$.

(b) To lose both rounds, the dealer has to roll neither 2 nor 7 on both rolls. The probability of this happening on one roll is $1 - \frac{1}{6} - \frac{1}{36} = \frac{29}{36}$. So the probability of this happening on both rolls is $\frac{29}{36} \cdot \frac{29}{36} = \frac{841}{1296}$.

(c) The probability of winning once is $1 - \frac{1}{108} - \frac{841}{1296} = \frac{443}{1296}$.

What if we want to calculate *this* directly? This is a lot trickier. One of these three mutually exclusive things must happen:

- The dealer rolls 2 on the first round and something other than 7 (potentially including 2) on the second round. The chances of this are $\frac{1}{36} \cdot \frac{5}{6} = \frac{5}{216}$.
- The dealer rolls 7 on the first round and something other than 2 (potentially including 7) on the second round. The chances of this are $\frac{1}{6} \cdot \frac{35}{36} = \frac{35}{216}$.
- The dealer rolls something other than 2 or 7 on the first round, then rolls either 2 or 7 on the second round. The former has a chance of $\frac{29}{36}$, as before and the latter has a chance of $1 - \frac{29}{36} = \frac{7}{36}$. The chances of this are $\frac{29}{36} \cdot \frac{7}{36} = \frac{203}{1296}$.

The sum of all of these is indeed the lovely $\frac{443}{1296}$.

(d) Here our expected winnings are $(0)(\frac{841}{1296}) + (1000)(\frac{443}{1296}) + (2000)(\frac{1}{108}) = \frac{467000}{1296} \approx \$360.34$, beating our earlier $333.\overline{33}$.

(e) To see the shortcoming of the two-sevens strategy, imagine that we instead had, say, 100 tokens, and the game ran for 100 rounds. Suppose we put all 100 tokens on 7. Now at some point the dealer rolls a 6. "Oh no," we think. Now we feel foolish. Surely we should have put at least *one* token on 6, right? since at least one was bound to come up...

Even in the regular game, picking 7 twice is not a diversified enough strategy. Even though there is a very small chance of 2 coming up, it is actually substantially more likely that we will see (2, 7) or (7, 2) (probability $\frac{443}{1296}$) than (7, 7) (probability $\frac{10}{36} = \frac{360}{1296}$). This is counterintuitive, but again, this is why WHEN we are IN DOUBT, we MATH IT OUT.

(f) Clearly,[5] picking, e.g., 3 and 7 is better than picking 2 and 7. We can show this with the same math. The best we can do is to pick 6 and 7

[5] Tip for reading math proofs critically: when someone says "clearly", that's the part of the proof you should *really* be suspicious about, since it means the author thought something was obvious but couldn't find a way to actually demonstrate it.

(or, equivalently, 7 and 8). Our expected winnings from that strategy are **much** better: using the same reasoning as before, the expected winnings are $(0)\frac{625}{1296} + (1000)\frac{611}{1296} + (2000)\frac{5}{108} = \frac{91375}{162} \approx \$564.04$ – so much better than our paltry $333.33 from the two-sevens strategy!

(g) If the game goes on for 36 rolls, how many tokens should we place on 7, for instance? We might consider how many 7s will probably come up: the chance of a 7 is $\frac{1}{6}$, so we would expect 6 of them. Adapting the table from page 7, we would expect the following frequency distribution:

2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	5	4	3	2	1

So placing our tokens that way should be the best we can do! Ian suspects this is true but has not formally proven it yet. But the overall idea is: **the placement of our tokens should be as close an approximation as possible to the probability distribution of the die rolls**. In the 36-round version, it was clear how to do this. In the two-round version of the game, a 6 and a 7 was a better approximation than the two 7s.

IV. Conditional Probability

This is one of the most crucial topics in CS109 (and in AI/ML!) Let's start by examining the differences between some similar-looking expressions.

Problem 4. Suppose we roll an 8-sided die. Consider the following two events:

Event A: The result is even. *Event B*: The result is ≥ 6.

(a) What is $P(A)$?

(b) What is $P(B)$?

(c) What is $P(A \cap B)$? ($\cap$ means "and" / the intersection of two sets.)

(d) What is $P(A \cup B)$? ($\cup$ means "or" / the union of two sets)

(e) What is $P(A|B)$? ($|$ means A, given that B is true.)

(f) Is it true in this case that $P(A \cap B) = P(A)P(B)$? Would you expect it to be always true, always false, or possibly either?

(g) Is it true in this case that $P(A \cap B) = P(A|B)P(B)$? Would you expect it to be always true, always false, or possibly either?

(h) What is $P(B|A)$?

(i) Now let C be the event that the result is a cube (1^3 or 2^3, i.e., 1 or 8). What is $P(C|A \cap B)$?

Solutions to Problem 4. In all of parts (a) through (d), the sample space S is the set of all possible outcomes: $\{1, 2, 3, 4, 5, 6, 7, 8\}$. The size of S, namely $|S|$, is 8.

(a) The event space E is $\{2, 4, 6, 8\}$, so $P(A) = \frac{|E|}{|S|} = \frac{4}{8} = \boxed{\frac{1}{2}}$.

(b) Now $E = \{6, 7, 8\}$, so $P(B) = \boxed{\frac{3}{8}}$.

(c) Here our new event of interest is the intersection of events A and B. We see that the intersection of $\{2, 4, 6, 8\}$ and $\{6, 7, 8\}$ is $\{6, 8\}$, so $P(A \cap B) = \frac{2}{8} = \boxed{\frac{1}{4}}$.

(d) The union of $\{2, 4, 6, 8\}$ and $\{6, 7, 8\}$ is $\{2, 4, 6, 7, 8\}$, so $P(A \cup B) = \frac{2}{8} = \boxed{\frac{5}{8}}$.

(e) Once we condition on B, we change the sample space. We are focusing only on the world in which event B is true, namely, the set $\{6, 7, 8\}$. So $|S| = 3$. Now, in this world, A is true for only 6 and 8, so $|E| = 2$ and $P(A|B) = \frac{|E|}{|S|} = \boxed{\frac{2}{3}}$.

   Alternatively, by the definition of conditioning, $P(A|B) = \frac{P(A \cap B)}{P(B)}$. We already have these values, and we get $\frac{\frac{1}{4}}{\frac{3}{8}} = \frac{2}{3}$.

(f) In this case, $P(A \cap B) = \frac{1}{4}$, and $P(A)P(B) = \frac{1}{2} \cdot \frac{3}{8} = \frac{3}{16}$. So the two sides are not equal. That is, the probability that two events are both true is not necessarily the product of their separate probabilities. This only holds (by definition) if the two events are *independent*. Here, informally, notice that event B being true (i.e. restricting ourselves to the world $\{6, 7, 8\}$) makes event A (an even result) more likely $(\frac{2}{3})$ than it would have been in the overall sample space $(\frac{1}{2})$.

(g) In this case, $P(A \cap B) = \frac{1}{4}$, and $P(A|B)P(B) = \frac{2}{3} \cdot \frac{3}{8} = \frac{1}{4}$. So yes, the two sides are equal in this case. Moreover, this holds in general! It follows directly from the definition of $P(A|B)$ as $\frac{P(A \cap B)}{P(B)}$.

(h) Now, by conditioning on A, we are focusing only on the world in which event A is true, namely, the set $\{2, 4, 6, 8\}$. In this world, B is true for only 6 and 8, so $|E| = 2$ and $P(B|A) = \frac{|E|}{|S|} = \frac{1}{2} = \boxed{\frac{1}{2}}$.

   We could also use the definition of conditioning as above. Alternatively, by Bayes' Rule, $P(B|A) = \frac{P(A|B)P(B)}{P(A)}$. We have these values, and we get $\frac{\frac{2}{3}\cdot\frac{3}{8}}{\frac{1}{2}} = \frac{1}{2}$.

(i) We treat $A \cap B$ as a single event here and use the definition of conditioning: $P(C|A\cap B) = \frac{P(C\cap(A\cap B))}{P(A\cap B)} = \frac{P(A\cap B\cap C)}{P(A\cap B)}$. Now, $A \cap B \cap C$ is the set of outcomes that are even, greater than 6, and a cube. The only outcome that works is 8. Therefore $P(A \cap B \cap C) = \frac{1}{8}$, and the overall answer is $\frac{\frac{1}{8}}{\frac{1}{4}} = \boxed{\frac{1}{2}}$.

V. Bayes' Rule Intuition

Recall that for arbitrary events A and B, Bayes' Rule is $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$.

If you don't like memorizing, you can regenerate this from the following possibly-easier-to-remember expressions:

$P(A \cap B) = P(A|B)P(B)$, from the definition $P(A|B) = \frac{P(A\cap B)}{P(B)}$

$P(A \cap B) = P(B|A)P(A)$, similarly

and therefore $P(A|B)P(B) = P(B|A)P(A)$. Dividing both sides of this by either $P(B)$ or $P(A)$ yields one of the forms of Bayes' Rule.

One key thing to keep in mind about Bayes' Rule is that it is **always true, regardless of whether events A and B are independent**. This is because, as we see above, it is the result of just wrangling some definitions; no assumptions about independence of A and B were ever made (or needed).

Problem 5: 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.

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

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

(c) 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!)

(d) Notice that $P(D = 1|H = 1)$ and $P(H = 1|D = 1)$ ended up being very different. Can you imagine an alternate scenario in which they *would* be the same?

Solutions to Problem 5.

(a) The problem tells us that within demographic D, 5% have $H = 1$. So $P(H = 1|D = 1) = \boxed{0.05}$.

(b) Here we can use Bayes' Rule: $P(D = 1|H = 1) = \frac{P(H=1|D=1)P(D=1)}{P(H=1)}$. We have $P(H = 1|D = 1) = 0.05$, and the problem tells us that $P(D = 1) = 0.1$, but we are not directly given $P(H = 1)$. We must find it using the Law of Total Probability: this probability is essentially a weighted average over the subgroups $D = 0$ and $D = 1$. Specifically, $P(H = 1) = P(H = 1|D = 1)P(D = 1) + P(H = 1|D = 0)P(D = 0)$. Notice that the first term on the right looks exactly like the numerator of our original Bayes' Rule expression, which is not a coincidence!

To find $P(H = 1|D = 0)P(D = 0)$, we use $P(H = 1|D = 0) = 0.01$ from the problem statement, and $P(D = 0) = 1 - P(D = 1) = 0.9$. Therefore overall we have

$$P(D = 1|H = 1) = \frac{P(H=1|D=1)P(D=1)}{P(H=1)} = \frac{P(H=1|D=1)P(D=1)}{P(H=1|D=1)P(D=1)+P(H=1|D=0)P(D=0)}$$

$$= \frac{0.05\cdot 0.1}{0.05\cdot 0.1+0.01\cdot 0.9} = \frac{0.005}{0.005+0.009} = \boxed{\frac{5}{14}}$$

Another (occasionally useful) strategy for this type of problem is just to invent a population of some arbitrary number of people, say, 10000. Then, per the problem statement: 1000 people are in demographic D, and the other 9000 are not. 5% of the people in demographic D – i.e. 50 – have $H = 1$. 1% of the people not in demographic D – i.e. 90 – have $H = 1$. Therefore the set of people with $H = 1$ consists of the 50 with $D = 1$ and the 90 with $D = 0$. So $P(D = 1|H = 1) = \frac{50}{50+90} = \frac{5}{14}$.

(c) Our $P(H = 1)$ value of 0.014 from the previous part implies that $P(H = 0) = 0.986$. The model will always be right on the part of the population with $H = 0$, and it will always be wrong on the part of the population with $H = 1$, so we get a seemingly impressive accuracy rate of $(1)(0.986)+(0)(0.014) = \boxed{0.986}$. But the model is not doing **anything** to help the people with the health condition – the very people who need the help the most! So accuracy alone does not a good model make.

(d) Bayes' Rule tells us that if $P(D = 1|H = 1) = P(H = 1|D = 1)$, then we must also have $\frac{P(D=1)}{P(H=1)} = 1$, i.e., $P(D = 1) = P(H = 1)$. One sad way for this to be true would be for everyone in D to have $H = 1$, and everyone not in D to have $H = 0$. Alternatively, using the $P(D) = 0.1$ from the original problem, $P(D = 1|H = 1) = P(H = 1|D = 1)$ would also hold if everyone had a 10% chance of having $H = 1$, regardless of demographic.