Chris Piech
CS 109

October 15, 2018

Problem Set #2 Solutions

With solutions by Mehran Sahami, writeup by Will Monroe, edited by Julia Daniel

1. $P(\text{Java}) = 0.36$. $P(\text{C++} \mid \text{Java}) = 0.24$. $P(\text{C++}) = 0.33$.

a. $P(\text{Java}, \text{C++}) = P(\text{C++} \mid \text{Java})P(\text{Java}) = (0.24)(0.35) = 0.084$

b. $P(\text{Java} \mid \text{C++}) = P(\text{Java}, \text{C++})/P(\text{C++}) = 0.084/0.33 = 0.255$

2. $P(E \mid F) = \frac{P(EF)}{P(F)} = \frac{(1/52)(3/51)}{(1/52)} = \frac{3}{51}$. Event EF occurs when first the Ace of Spades is chosen (1/52) and then second one of 3 remaining Aces is chosen from the remaining 51 cards (3/51). We can even intuit this without worrying about F at all - all we are really asked to find out is the probability that the second card is an ace, which is $\frac{3}{51}$.

3. a. $1 - (1 - p)^5$. This is $1 - P(\text{all 5 servers failed})$. Alternatively, the same probability can be arrived at through the sum:

$$\binom{5}{1}p(1-p)^4 + \binom{5}{2}p^2(1-p)^3 + \binom{5}{3}p^3(1-p)^2 + \binom{5}{4}p^4(1-p) + \binom{5}{5}p^5$$

The explanation for this summation follows the same reasoning as part (c) below.

b. $\binom{5}{3}p^3(1-p)^2$. We choose 3 of the 5 servers that are working, and then multiply by p^3 (the probability that the chosen 3 servers are working) and $(1-p)^2$, which is the probability that the unchosen 2 servers are not working.

c. $\binom{5}{3}p^3(1-p)^2+\binom{5}{4}p^4(1-p)+\binom{5}{5}p^5$. This represents the sum of the following probabilities: $P(\text{exactly 3 servers working}) + P(\text{exactly 4 servers working}) + P(\text{exactly 5 servers working})$. $P(\text{exactly } i \text{ servers working})$ is determined by choosing i of the 5 servers that are working and then multiplying by p^i (the probability that the chosen i servers are working) and $(1-p)^{5-i}$, which is the probability that the unchosen $5-i$ servers are not working.

4. a. The probability that a robot succeeds at all three games is $(0.3)^3$, so the probability the robot gets flagged is $1 - (0.3)^3 = 0.973$.

b. Similarly, the probability a human gets flagged is $1 - (0.95)^3 \approx 0.143$.

c. Let F be the event that the player is flagged, and let R be the event that the player is a robot. We can use Bayes' rule to compute $P(R \mid F)$:

$$\begin{aligned} P(R \mid F) &= \frac{P(F \mid R)P(R)}{P(F)} \\ &= \frac{P(F \mid R)P(R)}{P(F \mid R)P(R) + P(F \mid R^C)P(R^C)} \\ &= \frac{(0.973)(0.1)}{(0.973)(0.1) + (0.143)(0.9)} \\ &\approx 0.431 \end{aligned}$$

5. Let W_X be the payoff of the game that has a maximum payoff of X. The expectation for the game is given by:

$$E[W_X] = \sum_{i=0}^{\lfloor \log_2 X \rfloor} \left(\frac{1}{2}\right)^{i+1} 2^i + \left(1 - \sum_{i=0}^{\lfloor \log_2 X \rfloor} \left(\frac{1}{2}\right)^{i+1}\right) X$$

The first term in the expectation represents the cases where the number of consecutive heads flipped (denoted by the index i) leads to a payoff of 2^i, and 2^i is less than or equal to the maximum payout of X, or equivalently $i \leq \lfloor \log_2 X \rfloor$. The second term represents the cases where the payoff of the game is the maximum value X. Note that the probability of the second case is simply the complement of first case: $1 - P(\text{payoff is } 2^N)$. We use the equation above for each of the computations below.

a. $X = 5$. $E[W_X] = \sum_{i=0}^{3} \left(\frac{1}{2}\right)^{i+1} 2^i + \left(1 - \sum_{i=0}^{3} \left(\frac{1}{2}\right)^{i+1}\right) 5 = 2.125$.

b. $X = 500$. $E[W_X] = \sum_{i=0}^{8} \left(\frac{1}{2}\right)^{i+1} 2^i + \left(1 - \sum_{i=0}^{8} \left(\frac{1}{2}\right)^{i+1}\right) 500 = 5.47$.

c. $X = 4096$. $E[W_X] = \sum_{i=0}^{12} \left(\frac{1}{2}\right)^{i+1} 2^i + \left(1 - \sum_{i=0}^{12} \left(\frac{1}{2}\right)^{i+1}\right) 4096 = 7.0$.

6. $P(\text{at least one 1 in } n \text{ bits}) = 1 - P(\text{no 1's in } n \text{ bits}) = 1 - (1-p)^n$. We want to determine n, such that $1 - (1-p)^n \geq 0.7$.

$$\begin{aligned}
1 - (1-p)^n &\geq 0.7 && \\
(1-p)^n &\leq 0.3 && \text{rearranging terms} \\
\log[(1-p)^n] &\leq \log(0.3) && \text{take log of both sides} \\
n \log(1-p) &\leq \log(0.3) && \text{property of logs} \\
n &\geq \log(0.3)/\log(1-p) && \text{divide by } \log(1-p)
\end{aligned}$$

Note that since $\log(1-p) \leq 0$, we must flip the $\leq$ to a $\geq$ in the inequality when we divide both sides by this quantity.

Finally, noting that n must be an integer, we use the ceiling function to obtain:

$$n \geq \lceil \log(0.3)/\log(1-p) \rceil$$

7. a. By conditional independence, the probability of liking all three movies is $p_1 p_2 p_3$.

b. Let $L_i \mid G$ be the event that they like movie M_i given that they like the genre.

$$\begin{aligned}
P(L_1 \cup L_2 \cup L_3 \mid G) &= 1 - P((L_1 \cup L_2 \cup L_3)^C \mid G) \\
&= 1 - P({L_1}^C {L_2}^C {L_3}^C \mid G) \\
&= 1 - (1-p_1)(1-p_2)(1-p_3)
\end{aligned}$$

Another approach is to use the inclusion/exclusion principle to expand the probability of the union:

$$P(L_1 \cup L_2 \cup L_3 \mid G) = p_1 + p_2 + p_3 - p_1 p_2 - p_1 p_3 - p_2 p_3 + p_1 p_2 p_3$$

c. $$P(G \mid L_1L_2L_3) = \frac{P(L_1L_2L_3 \mid G)P(G)}{P(L_1L_2L_3 \mid G)P(G) + P(L_1L_2L_3 \mid G^C)P(G^C)}$$
$$= \frac{(p_1p_2p_3)(0.6)}{(p_1p_2p_3)(0.6) + (q_1q_2q_3)(0.4)}$$

8. a. The only way the `fairRandom` function can return a value is when `r1` = 1 and `r2` = 0 or vice versa, so we only need to consider the final values of `r1` and `r2` that led to the function terminating. Let event A = (`r2` = 1). Let event B = (`r1` = 1 and `r2` = 0). Let event C = (`r1` = 0 and `r2` = 1). We want to show that $P(A \mid B \cup C) = 1/2$.

We know
$$P(A \mid B \cup C) = \frac{P(A \cap (B \cup C))}{P(B \cup C)}$$

Since event A is only compatible with C, we have
$$P(A \cap (B \cup C)) = P(C)$$

Moreover, B and C are mutually exclusive events, so:
$$P(B \cup C) = P(B) + P(C)$$

So we have:
$$P(A \mid B \cup C) = \frac{P(C)}{P(B) + P(C)}$$
$$P(C) = (1 - p)p$$
$$P(B) = p(1 - p)$$

Combining yields the desired result: $P(A \mid B \cup C) = \frac{(1-p)p}{p(1-p)+(1-p)p} = 1/2$.

b. No, `simpleRandom` does not generate 1's and 0's with equal probability. Consider the fact that `simpleRandom` will always return a 1 if `r1` is initially set to 0 (since the generation of any subsequent 0's will not cause the function to return and when the first 1 is generated, the function will return). Similarly, it will always return a 0 if `r1` is initially set to 1. So, $P($`simpleRandom` returns 1$)$ = $P($initial value of `r1` = 0$) = 1 - p$, which is not necessarily 1/2.

9. This can be solved using Bayes' theorem. Let O be the event that we observe a window. The problem gives us $P(O|L_1)$ and $P(O|L_2)$. Note that since L_1 and L_2 are the only locations, $L_1 = {L_2}^C$ and vice versa.

$$P(L_1 \mid O) = \frac{P(O \mid L_1)P(L_1)}{P(O \mid L_1)P(L_1) + P(O \mid L_1^C)P({L_1}^C)}$$
$$= \frac{P(O \mid L_1)(0.8)}{P(O \mid L_1)(0.8) + P(O \mid L_1^C)(0.2)}$$
$$= \frac{(0.2)(0.8)}{(0.2)(0.8) + (0.9)(0.2)}$$
$$= \frac{0.16}{0.34} \approx 0.471$$

10. Again we apply Bayes' theorem, using the general form of the law of total probability:

$$P(L_i \mid O) = \frac{P(O \mid L_i)P(L_i)}{\sum_{j=1}^{16} P(O \mid L_j)P(L_j)}$$

In code, we loop through the whole array, multiplying the prior and the conditional probabilities for each cell, and keeping a running sum of those products. We then divide each product by the running sum (to ensure all probabilities add up to one). This might look like the following in Python (using `load_csv_data` from the starter code):

```python
def main():
    prior = load_csv_data('prior.csv')
    cond = load_csv_data('conditional.csv')
    post = compute_posterior(prior, cond)
    for row in post:
        print(row)


def compute_posterior(prior, cond):
    '''
    Uses Bayes' theorem to compute a 2D array of posterior
    probabilities given a 2D array of prior probabilities
    and a 2D array of likelihoods (conditional probability
    of the evidence given each position).
    '''
    result = []
    normalizing_constant = 0.0

    for r in range(0, len(prior)):
        result_row = []
        for c in range(0, len(prior[r])):
            post = prior[r][c] * cond[r][c]
            result_row.append(post)
            normalizing_constant += post
        result.append(result_row)

    for r in range(0, len(result)):
        for c in range(0, len(result[r])):
            result[r][c] /= normalizing_constant

    return result
```

You can also use NumPy to calculate the same thing with much less code:

```python
def compute_posterior(prior, cond):
    import numpy as np
```

```
prior = np.array(prior)
cond = np.array(cond)

result = prior * cond
return result / result.sum()
```

The result:

0.074	0.188	0.074	0.005
0.005	0.149	0.094	0.074
0.001	0.005	0.149	0.094
0.001	0.001	0.010	0.074

11. Since William's parents have brown eyes, each must have at least one brown eye gene. Since they have a blue eyed daughter (William's sister, Claire), they must also each have one blue eye gene (in order to have been able to produce a daughter with two blue eye genes). As a result, William would be equally likely to receive a brown or blue eye gene from each parent. Let X denote the number of blue eye genes that William received.

a. Since William has brown eyes, we know he does not have two blue eye genes, so we compute

$$P(X = 1 \mid X < 2) = \frac{P(X = 1, X < 2)}{P(X < 2)} = \frac{P(X = 1)}{P(X < 2)} = (2/4)/(3/4) = 2/3$$

$P(X = 1) = 2/4$, since out of the 4 possible gene pair possibilities, there are 2 that have only 1 blue eye gene. $P(X < 2) = 3/4$ since out of the 4 possible gene pair possibilities, there are 3 that have 0 or 1 blue eye gene.

b. Here we must consider the two cases where William either has a blue eye gene or does not. Note that William's wife will always pass a blue eye gene to the child, so the gene William passes with determine the child's eye color. Let event B = William's first child has blue eyes. Let event G = William has a blue eye gene. Note $P(G) = 2/3$ from part (a).

$$P(B) = P(B \mid G)P(G) + P(B \mid G^C)P(G^C) = (1/2)(2/3) + 0(1/3) = 1/3$$

c. Let event E = William's first child has brown eyes. Let event F = William's second child has brown eyes. Let event G = William has a blue eye gene. We first compute $P(G \mid E)$.

$$\begin{aligned} P(G \mid E) &= \frac{P(E \mid G)P(G)}{P(E \mid G)P(G) + P(E \mid G^C)P(G^C)} \\ &= \frac{(1/2)(2/3)}{(1/2)(2/3) + 1(1/3)} = \frac{(1/3)}{(2/3)} = 1/2 \end{aligned}$$

Expand the probability we want to compute by considering the cases where William has (or does not have) a blue eye gene, given the first child has brown eyes:

$$\begin{aligned} P(F|E) &= P(F \mid GE)P(G \mid E) + P(F \mid G^C E)P(G^C \mid E) \\ &= (1/2)(1/2) + 1(1 - 1/2) = 1/4 + 1/2 = 3/4 \end{aligned}$$

12. a. Yes: $P(T_1 \mid G)P(T_2 \mid G) = 0.72 = P(T_1, T_2 \mid G)$.

b. Yes: $P(T_1 \mid G^C)$, $P(T_2 \mid G^C)$, and $P(T_1, T_2 \mid G^C)$ are all 0.

c. $P(T_1) = P(T_1 \mid G)P(G) + P(T_1 \mid G^C)P(G^C) = 0.8 \cdot 0.6 + 0 = 0.48$.

d. $P(T_2) = P(T_2 \mid G)P(G) + P(T_2 \mid G^C)P(G^C) = 0.9 \cdot 0.6 + 0 = 0.54$.

e. No: $P(T_1, T_2) = P(T_1, T_2 \mid G)P(G) + 0 = 0.72 \cdot 0.6 = 0.432$ does not equal $0.48 \cdot 0.54$.

13. In the network G_5 is expressed based on G_2 and G_3. The trait is expressed based on G_5.

a. $P(T) = 0.301$

b. $P(G_1) = 0.702$
$P(G_2) = 0.301$
$P(G_3) = 0.501$
$P(G_4) = 0.802$
$P(G_5) = 0.327$

c. $P(T, G_1) = 0.212$ $\quad P(T)P(G_1) = 0.211$
$P(T, G_2) = 0.091$ $\quad P(T)P(G_2) = 0.090$
$P(T, G_3) = 0.292$ $\quad P(T)P(G_3) = 0.151$
$P(T, G_4) = 0.297$ $\quad P(T)P(G_4) = 0.241$
$P(T, G_5) = 0.294$ $\quad P(T)P(G_5) = 0.098$
It seems reasonable to assume T is independent of G_1 and G_2.

d. $P(T \mid G_1) = 0.302$
$P(T \mid G_2) = 0.302$
$P(T \mid G_3) = 0.583$
$P(T \mid G_4) = 0.371$
$P(T \mid G_5) = 0.900$

e. You should figure out that G_1 and G_2 do not seem to have an effect on T. Of the three genes that do modulate the probability of T, G_5 seems to have the largest impact.