Practice Problem Set 3: Conditional Probability & Independence

Reading: Schaum's Chapters 4.5 to 4.6, LLM Chapter 18 (in the 2017 revision).

Optional Extra Practice. In the 2000 version of Schaum's: 4.35-4.40. 4.69-4.85. In the 2011 version of Schaum's: 4.26-4.39. 4.69-4.85.

Exercise 1. (Fuzzing to find exploits.) You are using fuzzing to find bug in a computer program that you will then develop into an exploit. To do this, you are feed the program a bunch of independent random inputs r_i and check if it crashes or not. Your ultimate goal is to get the program to crash; once you figure out to get the program to crash, you have a 'clue' that you will then use to develop your exploit.

Suppose this program crashes on 10^{-5} of inputs.

1. Let F_i be the event the the program crashes on the i^{th} input. Write down the following events in term of unions, intersections, and negations of the F_i.
   (a) "You feed the program n independent random inputs and it crashed on the 7^{th} only."
   (b) "You feed the program n independent random inputs and it crashed exactly once."
   (c) "You feed the program n independent random inputs and never crashed."
   (d) "You feed the program n independent random inputs and it always crashed."
   (e) "You feed the program n independent random inputs and crashed at least once."
   (f) "You feed the program n independent random inputs and crashed at least twice."
   (g) "The program crashed for the first time on the 17^{th} input."
2. Determine the probability of each of the above events.
3. How big does n need to be to ensure that you can find at least one input that crashes the program with probability 75%?

Exercise 2. (Random bit flipping) A faulty communication line for digital signals flips 0 bits with probability $\frac{1}{4}$ and 1 bits with probability 50%. If 40% of a transmitted file consists of 0's and 60% consists of one, what is the probability that a received 0 was transmitted correctly (*i.e.,* , it corresponds to a transmitted 0)? Hint: Use a tree!

Exercise 3. (Cards.) A 52 card deck is shuffled. You are given 13 cards.

- If you have one ace, what is the probability that you get another ace?
- If you have the ace of spades, what is the probability that you get another ace?

Notice that the answers are not the same! Explain in words why they are different.

Exercise 4. There are three caves in Boston. A wolf is looking for a home; he moves into cave 1 with probability $\frac{5}{9}$, cave 2 with probability $\frac{1}{9}$ and cave 3 with probability $\frac{1}{9}$, and with the remaining probability he moves to San Diego. A goat moves into one of the unoccupied caves, choosing his cave with equal probability. These caves are in Boston, so it snows with probability $\frac{9}{10}$. The goat always leaves tracks in the snow, but the wolf does not. What is the probability that the wolf lives in cave 2 given that the there are no tracks in front of cave 1? Hint: Use a tree!

Exercise 5. (Hash tables.) You are given a set of 42 items $(x_1, ..., x_{42})$ that you need to store in a memory. The items x_i are 100-bit strings, i.e. $x_i \in \{0,1\}^{100}$. The memory consists of 800 slots, and each slot can store a single 100-bit string.

You do this using a hash table as follows. You have a hash function,

$$h : \{0,1\}^{100} \to \{1, ..., 800\}$$

that independently and uniformly randomly maps every 100-bit string item to a number from 1 to 800. You store every item x_i in slot $h(x_i)$.

1. Consider the first slot in memory. What is the probability that the first item x_1 is stored in the first slot?

2. What is the probability that *at least* one item is stored in the first slot?

3. Suppose now that the first 50 slots of memory are corrupted; any item that is supposed to be stored in one of these slots is lost. What is the probability that the first item x_1 is lost?

4. To combat memory corruptions, you will use a second hash function,

   $$g : \{0,1\}^{100} \to \{1, ..., 800\}$$

   that independently and uniformly randomly maps every 100-bit string item to a number from 1 to 800, and is independent of the hash function h. Now, you store every item x_i in two different slots, namely slot $h(x_i)$ and slot $g(x_i)$. What is the probability that the first item x_1 is lost?

5. What is the probability that *some item* is lost?