Will Monroe
CS 109

Problem Set #3
July 12, 2017

Problem Set #3
Due: 12:30pm on Wednesday, July 19th

With problems by Mehran Sahami and Chris Piech

For each problem, briefly explain/justify how you obtained your answer. Brief explanations of your answer are necessary to get full credit for a problem even if you have the correct numerical answer. The explanations help us determine your understanding of the problem whether or not you got the correct answer. Moreover, in the event of an incorrect answer, we can still try to give you partial credit based on the explanation you provide. It is fine for your answers to include summations, products, factorials, exponentials, or combinations; you don't need to calculate those all out to get a single numeric answer.

Unless otherwise stated, you may also use functions in a library like Python's `scipy.stats` to compute values of PMFs and CDFs; if you use these, provide your code that calls these functions and explain how you arrived at each parameter to a function or constructor.

1. An urn contains 4 white balls and 4 black balls. Two balls are drawn randomly (without replacement) from the urn. If they are the same color, you win \$2.00. If they are different colors, you lose \$1.00 (i.e., you win -\$1.00). Let X = the amount you win.
   a. What is $E[X]$?
   b. What is $\text{Var}(X)$?

2. Say there are k buckets in a hash table. Each new string added to the table is hashed to bucket i with probability p_i, where $\sum_{i=1}^{k} p_i = 1$. If n strings are hashed into the table, find the expected number of buckets that have at least one string hashed to them. (Hint: Let X_i be a binary variable that has the value 1 when there is at least one string hashed to bucket i after the n strings are added to the table (and 0 otherwise). Compute $E\left[\sum_{i=1}^{n} X_i\right]$.)

3. Recall the coin-flipping game set-up discussed in class (called the "St. Petersburg paradox"): there is a fair coin which comes up "heads" with probability $p = 0.5$. The coin is flipped repeatedly until the first "tails" appears. Let N = number of coin flips before the first "tails" appears (i.e., N = the number of consecutive "heads" that appear). Given that no one really has infinite money to offer as payoff for the game, consider a variant of the game where you win $\texttt{min}(\$2^N, X)$, where X is the maximum amount that the game provider will pay you after playing. Compute the expected payoff of the game for the following values of X. Show how you derived your answer.
   a. X = \$20.
   b. X = \$500.
   c. X = \$10,000.

4. Say we have an integer array `arr[10]` (indexed from 0 to 9), which contains the numbers 1 through 10 in sorted order. Now, say key is a randomly generated integer value between 1 and 10 (inclusive), where each value is equally likely.

   a. What is the expected number of times that the "equality test" (as noted by the comment in the code) is executed in the function linear below (assuming `linear` is passed the array `arr` and the randomly chosen value `key`). Give an exact value (not a big-Oh running time or an approximation) for the expectation, and explain how you derived your answer.

```
int linear(int arr[], int key) {
  for(int i = 0; i < 10; i++) {
    // Equality test: (arr[i] == key)
    if (arr[i] == key) return i;
  }
  return -1;  // Will never get here when key is in [1,10]
}
```

   b. Under the same conditions for array `arr` and the randomly chosen value `key`, what is the expected number of times that the "equality test" is executed in the function binary below. Give an exact value (not a big-Oh running time or an approximation) for the expectation, and explain how you derived your answer.

```
int binary(int arr[], int key) {
  int low = 0;
  int high = 9;
  while (low <= high) {
    int mid = (low + high) / 2;
    // Equality test: (arr[mid] == key)
    if (arr[mid] == key) return mid;
    else if (arr[mid] < key) low = mid + 1;
    else high = mid - 1;
  }
  return -1;  // Will never get here when key is in [1,10]
}
```

5. When a bit string is sent over a network, each bit in the string will independently be corrupted (flipped) with probability p. Say we come up with a protocol for sending strings over the network where if we have an original string s of length n bits, we create the message ss (just two copies of the original message in a row, so ss has length $2n$ bits) and send that message over the network instead. Thus, the recipient can detect an error if there are any discrepancies between the first and second halves of the string they receive. Note that it is possible for the recipient to not be able to detect an error if a bit and its corresponding duplicate in the second half of the message are both corrupted (flipped).

a. What is the expression (in terms of n and p) for the probability that the message ss is received without any corruption? Also, compute the numerical value for your expression for $n = 5$ and $p = 0.1$.
b. What is the expression (in terms of n and p) for the probability that the recipient receives a corrupted message and is not able to detect that it is corrupted? Also, compute the numerical value for your expression for $n = 5$ and $p = 0.1$.
c. What is the expression (in terms of n and p) for the probability that the recipient receives a corrupted message where the recipient can detect that some sort of corruption took place? Also, compute the numerical value for your expression for $n = 5$ and $p = 0.1$.

6. Suppose it takes at least 9 votes from a 12-member jury to convict a defendant. Suppose also that the probability that a juror votes that an actually guilty person is innocent is 0.25, whereas the probability that the juror votes that an actually innocent person is guilty is 0.15. If each juror acts independently and if 70% of defendants are actually guilty, find the probability that the jury renders a correct decision. Also determine the percentage of defendants found guilty by the jury.

7. Consider a hash table with n buckets. Now, m strings are hashed into the table (with equal probability of being hashed into any bucket).

   a. Let $n = 2{,}000$ and $m = 10{,}000$. What is the (Poisson approximated) probability that the first bucket has 0 strings hashed to it?
   b. Let $n = 2{,}000$ and $m = 10{,}000$. What is the (Poisson approximated) probability that the first bucket has 8 or fewer strings hashed to it?
   c. Let $m = 10{,}000$. What is largest integer value n such the Poisson approximated probability that an arbitrary bucket in the hash table will have no strings hashed to it is less than 0.5 (= 50%)?
   d. Let X be a Poisson random variable with parameter λ, that is: $X \sim \text{Poi}(\lambda)$. What value of λ maximizes $P(X = 3)$? Show formally (mathematically) how you derived this result. (Hint: at some point in your derivation you should be differentiating with respect to λ.)

   (Questions such as this allow us to compute appropriate sizes for hash tables in order to get good performance with high probability in applications where we have a ballpark idea of the number of elements that will be hashed into the table.)

8. Consider a computer cluster (data center) of 100 web servers, where incoming requests are randomly assigned to servers with equal probability. Based on historical averages, each server in the data center receives requests at a rate of 2 per second. Some buggy server code was just deployed to all the servers in the cluster and as a result any server will crash if it receives more than 6 requests in a second. What is the approximate probability that no servers have crashed 1 second after the buggy code is deployed?

9. The number of times a person's computer crashes in a month is a Poisson random variable with $\lambda = 7$. Suppose that a new operating system patch is released that reduces the Poisson parameter to $\lambda = 2$ for 80% of computers, and for the other 20% of computers the patch has no effect on the rate of crashes. If a person installs the patch, and has his/her computer crash 4 times in the month thereafter, how likely is it that the patch has had an effect on the user's computer (i.e., it is one of the 80% of computers that the patch reduces crashes on)?

10. An election has two candidates in a very close race: recent polls predict that candidate A will win about 51% of the vote, while candidate B will win about 49%.

    a. Suppose there are $N = 5{,}000$ voters in the election, and that every voter in the election votes randomly and independently with those probabilities: 0.51 for candidate A, 0.49 for candidate B. Give an expression (involving a sum) for the probability that candidate A wins the election (gets more than $N/2 = 2{,}500$ votes).
    b. Compute the numerical value of this probability. (Remember you can use `scipy.stats` or functions from similar libraries in other languages. Provide your code if you use these.)
    c. Compute the numerical value of the same probability using a Poisson approximation. Is Poisson a good approximation here? Why or why not?

11. **[Coding]** In this problem you'll explore one (rather silly) way of trying to get a computer program to write the next great novel.

    In the data distribution provided on the companion page for this assignment is a copy of Herman Melville's *Moby Dick*, from Project Gutenberg. We've processed the file so that each line is one sentence, and all the words and punctuation are separated by exactly one space.

    a. Make a plot[1] of the distribution (PMF) of sentence lengths in *Moby Dick*, from length 1 up to at least length 10, where the "length" of a sentence is the total number of words and punctuation marks it contains. In other words, if you were to choose a random sentence from the novel (with all sentences equally likely), what is the probability the sentence has n words (including punctuation), for $n = \{1, 2, 3, \ldots, 10\}$?
    b. Consider the following approach to generating random "sentences":
       (1) Start with zero words in your sentence.
       (2) Pick a random word from the novel. Append it to the sentence.
       (3) With probability 0.04, stop and return the current sentence. Otherwise, go to (2).
       Let X be the length of a sentence generated with this method. What is the distribution of X? Give the type of distribution and the values of any parameters.
    c. Plot the PMF of the distribution from part (b), from length 1 up to length 10. What is one noticeable difference between of the distribution of sentence lengths in the real novel text and the distribution produced by this random-sentence generator?

    (Some of you have probably done an assignment in CS 106B involving generating random text using n-grams and Markov chains. You may find it interesting to see how the distribution of sentence lengths in your text from that program compares!)

[1] You are not required to make this plot with code; if you prefer, your program can simply print the numbers for the plot, which you can then create with Excel or sketch by hand. But if you are interested in creating the plot in your program, `matplotlib` is a popular Python library for making plots. See `starter.py` in the data distribution.