Chris Piech
CS109

April 24, 2017

Problem Set #3

Due: 10:30am on Wednesday, May 3rd

For each problem, explain/justify how you obtained your answer in order to obtain full credit. In fact, most of the credit for each problem will be given for the derivation/model used as opposed to the final answer. Make sure to describe the distribution and parameter values you used (e.g., Bin(10, 0.3)), where appropriate. It is fine for your answers to include summations, products, factorials, exponentials, or combinations, unless you are specifically asked for a computed numeric answer.

Warmup

1. Recall the game set-up in the "St. Petersburg's paradox" discussed in class: 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 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 = \$5.

   b. X = \$500.

   c. X = \$4096.

2. Lyft line gets 2 requests per 5 mins, on average, for a particular route. A user requests the route and Lyft commits a car to take her. All users who request the route in the next five minutes will be added to the car—as long as the car has space. The car can fit up to three users. Lyft will make \$6 for each user in the car (the revenue) minus \$7 (the operating cost). How much does Lyft expect to make from this trip?

3. 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 $n = 4$ and $p = 0.05$.

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 $n = 4$ and $p = 0.05$.

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 $n = 4$ and $p = 0.05$.

4. Given our recent analysis of Justice Breyer's probabilistic arguments regarding jury selection, let's consider a situation involving juries. 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 0.2, whereas the probability that the juror votes that an actually innocent person is guilty is 0.1. If each juror acts independently and if 75% 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.

5. The number of times a person's computer crashes in a month is a Poisson random variable with $\lambda = 5$. Suppose that a new operating system patch is released that reduces the Poisson parameter to $\lambda = 3$ for 75% of computers, and for the other 25% of computers the patch has no effect on the rate of crashes. If a person installs the patch, and has his/her computer crash 2 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 75% of computers that the patch reduces crashes on)?

6. 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 $E\left[\sum_{i=1}^{k} X_i\right]$.)

7. Say we have a cable of length n. We select a point (chosen uniformly randomly) along the cable, at which we cut the cable into two pieces. What is the probability that the shorter of the two pieces of the cable is less than 1/4th the size of the longer of the two pieces? Explain formally how you derived your answer.

8. Let X be a continuous random variable with probability density function:

$$f(x) = \begin{cases} c\,(3 - 2x^2) & -1 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

a. What is the value of c?

b. What is the cumulative distribution function (CDF) of X?

c. What is E[X]?

9. A disk has just 0 and 1 bits written over its entire surface. A voltage reading taken by the disk drive head at a random bit that is actually a 0 will give a reading that is normally distributed with $\mu = 4$ and $\sigma^2 = 4$. A voltage reading taken at a bit that is actually a 1 will give a reading that is normally distributed with $\mu = 6$ and $\sigma^2 = 9$. Say that α is the fraction of bits on the disk that are actually 1s. Now say that the voltage is measured at a randomly chosen bit on the disk surface and gives a voltage reading of 5. For what value of α would the probability of making an error regarding the bit value be the same, regardless of whether one concluded that the bit was supposed to be a 0 or a 1?

   More formally, let event A = bit measured is actually a 1, and event B = bit measured is actually a 0. Let random variable R = voltage value read at the bit. The probability of making an "error" is the same whether the chosen bit was a 1 or a 0 when:

   $P(A \mid R = 5) = P(B \mid R = 5) = 0.5$.

   So, basically, you want to determine the value of α when $P(A \mid R = 5) = P(B \mid R = 5) = 0.5$.

Algorithmic Analysis

10. 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++) {
      if (arr[i] == key) return i;  // Equality test: (arr[i] == key)
   }
   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;
      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]
}
```

Bloom Filter

11. A bloom filter is a space-efficient, probabilistic set. In this problem we are going to look at it theoretically. Our bloom filter uses 3 different independent hash functions H_1, H_2, H_3 that each take any string as input and each return an index into a bit-array of length n. Each index is equally likely for each hash function.

    To add a string into the set, feed it to each of the 3 hash functions to get 3 array positions. Set the bits at all these positions to 1. For example, consider this bit-array of length $n = 10$. All values in the bit-array are initially zero.

Index:	0	1	2	3	4	5	6	7	8	9
Value:	0	0	0	0	0	0	0	0	0	0

    After adding a string "pie" where H_1("pie") = 4, H_2("pie") = 7 and H_3("pie") = 8:

Index:	0	1	2	3	4	5	6	7	8	9
Value:	0	0	0	0	**1**	0	0	**1**	**1**	0

    Bits are never switched back to 0. Now, $m = 24{,}000$ strings are added to the bloom filter.

    a. Let $n = 8{,}000$. What is the (Poisson approximated) probability that the first bucket has 0 strings hashed to it?

    b. Let $n = 8{,}000$. What is the (Poisson approximated) probability that the first bucket has 10 or fewer strings hashed to it?

    To *check* whether a string is in the set, feed it to each of the 3 hash functions to get 3 array positions. If any of the bits at these positions is 0, the element is not in the set. If all bits at these positions are 1 the string is reported as in the set (though it might never have been added).

    c. Let $n = 100{,}000$. After $m = 25{,}000$ strings have been added to the bloom filter, what is the probability that a string, that has not been added to the set, will (incorrectly) be reported as in the set? Use approximations where appropriate.

    d. Our bloom filter uses three hash functions. Was that necessary? Repeat your calculation in (c) assuming that we only used a single hash function (not 3).

    e. Chrome uses a Bloom Filter to keep track of malicious URLs. 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. A bit-array of length 100,000 takes about as much memory as 3,000 characters.

Errata: The numbers in part (c) were changed, and part (d) was modified to make the problem more compelling. Noon, April 25th.

Predicting Elections

12. [Coding] On May 7th France will be choosing between two candidates (candidate A and candidate B) to be their next president. There have been 10 polls which each asked voters if they intend to vote for candidate A or B. For each of the 10 polls we report their sample size (N samples), how many people said they would vote for candidate A (A votes), how many people said they would vote for candidate B (B votes). Not all polls are created equal—many have a bias towards one candidate or the other. For each poll we also report a value "weight" which represents how accurate we believe the poll was (see polls.csv):

Poll	N samples	A votes	B votes	Weight
1	862	548	314	0.93
2	813	542	271	0.85
3	984	682	302	0.82
4	443	236	207	0.87
5	863	497	366	0.89
6	648	331	317	0.81
7	891	552	339	0.98
8	661	479	182	0.79
9	765	609	156	0.63
10	523	405	118	0.68

a. First, assume that each sample in each poll is an independent experiment of whether or not a random person in France would vote for candidate A. In other words, there is no difference between a vote for candidate A in poll 1 vs a vote for candidate A in vote 7. Calculate the probability that a random person in France votes for candidate A.

b. The population of France is 64,888,792. Assume each person votes for candidate A with the probability calculated in the part (a) and otherwise votes for candidate B. What is the probability that candidate A gets more than half of the votes? Report your answer to two decimal places and explain how you computed it.

13. Nate Silvers at fivethirtyeight pioneered an approach called the "poll of polls" for predicting elections. For both candidates we are going to have a random variable which represents their strength on election night: variables S_A and S_B for candidates A and B respectively (this is the same ideas as ELO scores). The probability that A wins is $P(S_A > S_B)$.

a. Calculate the parameters for the random variables S_A and S_B. Both S_A and S_B are defined to be normal with the following parameters:

$$S_A \sim \mathcal{N}\left(\mu = \sum_i p_{A_i} \cdot \text{weight}_i,\ \sigma^2\right)$$

$$S_B \sim \mathcal{N}\left(\mu = \sum_i p_{B_i} \cdot \text{weight}_i,\ \sigma^2\right)$$

where p_{A_i} is the ratio of A votes to N samples in poll *i*, p_{B_i} is the ratio of B votes to N samples in poll *i*, weight_i is the weight of poll *i*, m_i is the N samples in poll *i* and:

$$\sigma = \frac{K}{\sqrt{\sum_i m_i}} \text{ s.t. } K = 350$$

b. Calculate $P(S_A > S_B)$ by simulating 100,000 fake elections. In each fake election draw a random sample for the strength of A from S_A and a random sample for the strength of B from S_B. If S_A is greater than S_B, candidate A wins. Else candidate B wins. Report your answer to two decimal places.

c. Write one reason to use the model in question 13(b) and one reason why you might want to use the model in 14(b).

d. Since the polls were simulated I know that the true value for the probability that candidate A wins is 0.661. On election night candidate B wins. Was your answer to part (b) "wrong"? Explain, briefly.

e. For extra credit, propose another way of calculating the probability that candidate A wins. The data in extraPolls1.csv and extraPolls2.csv are two other sets of 10 polls for the exact same election. Show that your model is able to consistently predict that the probability that candidate A wins is close to 0.661 in all three polling outcomes (the original polls.csv and the two extras).