Problem Set 5

Due: Wednesday, February 7, by 11:59pm (Tasks 1-6)

Task 7 due Friday, February 9, by 11:59pm

Instructions

Solutions format and late policy. See PSet 1 for further details. The same requirements and policies still apply. Also follow the typesetting instructions from the prior PSets.

Collaboration policy. The written problems on this pset may be done with a **single partner**. In this case, only one person will submit the written part on Gradescope and add their partner as a collaborator. You must do the coding part (Task 7a) on your own.

Individuals and pairs are still encouraged to discuss problem-solving strategies with other classmates as well as the course staff, but each pair must write up their own solutions and, as stated above, submit a single joint homework. However, you should make sure you are both involved in coming up with and writing up all the solutions.

Solutions submission. You must submit your solution via Gradescope. In particular:

- For the solutions to Task 1-6 submit to "PSet 5 [Written]" a **single** PDF file containing the solution to all tasks in the homework (for you and your partner). Each numbered task should be solved on its own page (or pages). Follow the prompt on Gradescope to link tasks to your pages. Do not write your names on the individual pages – Gradescope will handle that. **Tasks 1-6 are due Wednesday, February 7th at 11:59pm**
- For the programming part (Task 7 part a)), submit your code under "PSet 5 [Coding]" as a file called bloom.py. **This needs to be done individually! Task 7 is due Friday, February 9th at 11:59pm.**
- For the written parts of Task 7 (parts b and c), submit to "PSet 5 Task 7 [Written]" (one submission per partnership) and appropriately tag questions as usual. Add your partner to the submission as you did for Tasks 1-6, if applicable. **Task 7 is due Friday, February 9th at 11:59pm.** Late days are calculated separately for this problem, but we highly encourage you not to use them since you will want to study for the midterm over the weekend!

Task 1 – A random sphere [10 pts]

Suppose that the radius R of a certain sphere is a random variable where

$$R = \begin{cases} 1 & \text{with probability } 1/3 \\ 2 & \text{with probability } 1/3 \\ 3 & \text{with probability } 1/3 \end{cases}$$

Use LOTUS to calculate the expected volume of the sphere. Recall that a sphere with radius r has volume $\frac{4}{3}\pi r^3$.

Task 2 – A simple investing strategy [16 pts]

A certain investor buys n different stocks, with each independently having probability p of going up in value each day, where $0 < p < 1$. At the end of each day, he keeps all the stocks that went up and sells all of the others. He repeats this process each day, until he has sold all of his stocks. We assume that each stock has probability p of going up on any given day and whether or not that happens is independent of what happens on other days or what happens with other stocks.

Let X_i be the random variable representing the number of stocks the investor owns at the end of i days, so $X_0 = n$. Explain each of your answers to the following questions (4 points each)

a) What is the distribution of X_1 and its parameters? In other words, what is $\mathbb{P}(X_1 = k)$?

b) What is the distribution of X_2? In other words, what is $\mathbb{P}(X_2 = k)$? What distribution in our zoo is this, and what are its parameters?
Hint: Think about the probability that a particular stock 'survives' till the end of day 2.

c) Repeat the previous part for X_t for arbitrary integer t.

d) What is the probability that he owns at least one stock after t days? Your answer should not have any summations in it.

Task 3 – X and Y [18 pts]

Let X be a Geometric RV with parameter p, let Y be a Poisson RV with parameter λ, and $Z = \max(X, Y)$. Assume that X and Y are independent. For each of the following problems, your final answers should not have summations. You may want to use the Taylor series expansion of e^x, that is, that for any x,

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

a) (3 points) What is $\mathbb{P}(X > k)$? [Think about it from first principles]

b) (6 points) Compute $\mathbb{P}(X > Y)$. Hint: Use the law of total probability to obtain that

$$\mathbb{P}(X > Y) = \sum_{k=0}^{\infty} \mathbb{P}(X > Y | Y = k) \cdot \mathbb{P}(Y = k).$$

c) (4 points) Compute $\mathbb{P}(Z \geqslant X)$.

d) (5 points) Compute $\mathbb{P}(Z \leqslant Y)$.

Task 4 – Comparing photos [16 pts]

A random collection of N photos are each compared with each other to see if there is any person that is in both pictures. Thus a total of $M = \binom{N}{2} = \frac{N(N-1)}{2}$ photo comparisons are performed.

Answer each of the following questions (4 points each). Make sure that each of your answers are **not** in the form of a summation. **In each case include the expectation and variance of N as part of your answer.**

It may be useful to recall that for any random variable $E(X^2) = Var(X) + [E(X)]^2$.

a) What is the expected value of M if N equals some fixed positive integer c with probability 1? (Your answer will be a function of c.)

b) What is the expected value of M if N has a Poisson distribution with parameter λ? (Your answer will be a function of λ.)

c) What is the expected value of M if N has a geometric distribution with parameter p? (Your answer will be a function of p.)

d) What is the expected value of M if $N = 10X + 7$, where X is a Bernoulli random variable with parameter p? (Your answer will be a function of p.)

Task 5 – How many 6's? [12 pts]

Suppose that a fair 8-sided die is rolled repeatedly, with each roll independent of the others. Let Z be the number of rolls until (and including) the first time either a 2 or a 3 is rolled, and let W be the number of 6's rolled until the first 2 or 3 is rolled. So, for example if the sequence of die values until the first 2 or 3 is 6,5,4,8,7,6,7,1,2, then Z is 9 and W is 2.

Define

$$p(j) := \begin{cases} \mathbb{P}(W = j \mid Z = i) & j \in \{0, 1, \ldots, i-1\} \\ 0 & \text{otherwise} \end{cases}$$

Show that $p(j)$ is the probability mass function of a binomially distributed random variable and determine its parameters n and p.

Task 6 – Sample Sampling Algorithm [18 pts]

Consider the following algorithm for generating a random sample S of size n from the set of integers $\{1, 2, \ldots, N\}$, where $0 < n < N$.

```
Sample(N, n):
   S ← ∅                        // S is a set of distinct integers, initially an empty set
   while |S| < n do
      x ← RollDie(N)            // x is the outcome of rolling a fair N-sided die
      S ← S ∪ {x}               // if x is already in S it doesn't change
   return S
```

Let I be the number of die rolls until S is returned. Also, let I_i be the random variable which describes the number of rolls it takes from the time the set S has $i-1$ values to the first time a new value is added after that (i.e., the set S has i values).

Answer the following questions (6 points each)

a) What type of random variable from our zoo is I_i and what is/are the relevant parameter(s) for that random variable?

b) What is I in terms of the random variables I_i? Calculate $\mathbb{E}[I]$, expressing the result as a summation that depends on both N and n.

c) What is $\mathrm{Var}(I)$? You can leave your answer in summation form.

Task 7 – Bloom filters [Coding+Written, due Friday, Feb 9 at 11:59pm] [10+7 pts]

Note: This Task is due Friday, February 9 at 11:59pm. Late days calculated separately for this task.

Google Chrome has a huge database of malicious URLs, but it takes a long time to do a database lookup (think of this as a typical Set). They want to have a quick check in the web browser itself, so a space-efficient data structure must be used. A **Bloom filter** is a **probabilistic data structure** which only supports the following two operations:

- add(x): Add an element x to the structure.
- contains(x): Check if an element x is in the structure. It either returns "definitely not in the set" or "could be in the set".

It does **not** support the following two operations:

- delete an element from the structure.
- return an element that is in the structure.

The idea is that we can check our Bloom filter to see if a URL is in the set. The Bloom filter is always correct in saying a URL definitely isn't in the set, but may have false positives – it may say that a URL is in the set when it isn't. Only in these rare cases does Chrome have to perform an expensive database lookup to know for sure. Suppose that we have k **bit arrays** $t_1, \ldots, t_k$ each of length m (all entries are 0 or 1), so the total space required is only km bits or $km/8$ bytes (as a byte is 8 bits). Suppose that the universe of URLs is the set $\mathcal{U}$ (think of this as all strings with less than 100 characters), and we have k **independent and uniform** hash functions $h_1, \ldots, h_k : \mathcal{U} \to \{0, 1, \ldots, m-1\}$. That is, for an element x and hash function h_i, pretend $h_i(x)$ is a **discrete** $\text{Unif}[0, m-1]$ random variable. Suppose that we implement the add and contains function as follows:

Algorithm 1 Bloom Filter Operations

function INITIALIZE(k,m)
    for $i = 1, \ldots, k$: **do**
        t_i = new bit array of m 0's
function ADD(x)
    for $i = 1, \ldots, k$: **do**
        $t_i[h_i(x)] = 1$
function CONTAINS(x)
    return $t_1[h_1(x)] == 1 \land t_2[h_2(x)] == 1 \land \cdots \land t_k[h_k(x)] == 1$

Refer to Section 9.4 of the textbook and the relevant lecture for more details on Bloom filters.

a) (10 points) Implement the functions add and contains in the BloomFilter class of bloom.py.

To solve this task, we have set up a corresponding edstem lesson here. Press the Mark button above the terminal to run the unit tests we have written for you. Passing these unit tests is not enough. We have written a number of different tests for the Gradescope autograder. Your score on Gradescope will be your actual score - you have unlimited attempts to submit.

b) (3 points) Let's compare this approach to using a typical Set data structure. Google wants to store 1 million URLs, with each URL taking (on average) 23 bytes.

- How much space (in MB, 1 MB = 1 million bytes) is required if we store all the elements in a set?
- How much space (in MB) is required if we store all the elements in a Bloom filter with $k = 10$ hash functions and $m = 800{,}000$ buckets? Recall that 1 byte = 8 bits.

c) (4 points) Let's analyze the time improvement as well. Let's say an average Chrome user attempts to visit 36,500 URLs in a year, only 1,000 of which are actually malicious. Suppose it takes half a second (500ms) for Chrome to make a call to the database (the Set), and only 1 millisecond for Chrome to check containment in the Bloom filter. Suppose the false positive rate on the Bloom filter is 4%; that is, if a website is not malicious, the Bloom filter will incorrectly report it as malicious with probability 0.04. In order to check all 36,500 URLs, what is (in seconds):

- the time taken if we only used the database?
- the *expected* time taken if we used the Bloom filter + database combination described earlier?