CS1550/CS2540 Probabilistic Methods in CS Upfal

Homework 0

Due: N/A

This problem set is optional and will not be graded. Feel free to ask questions about these problems at hours! Solutions will be posted on January 30.

Please follow these guidelines when writing up your CS 155 homeworks.

A major goal of this class is that you learn how to mathematically analyze probabilistic processes and events. Hence, make sure your proofs are rigorous, i.e., that every step is adequately justified. Try to keep your proofs as simple and as concise as possible, keeping in mind that you will potentially have to examine multiple proof strategies in order to achieve this. Make sure to CLEARLY STATE YOUR ASSUMPTIONS at the beginning of any solution, when necessary. As we are doing probabilistic analysis, we shall require that you clearly define the probability space you are working with and identify the events within it that you will analyze.

You are allowed to discuss these problems with other students, but the writeups must be done by yourself without help from others. If you have any questions regarding how these guidelines apply to a particular problem or what they mean in general, please email the TAs.

Problem 1

The following approach is often called *reservoir sampling*. Suppose we have a sequence of items passing by one at a time. We want to maintain a sample of one item with the property that at each step it is uniformly distributed over all the items that we have seen up to this step. Moreover, we want to accomplish this without knowing the total number of items in advance or storing all of the items that we see.

Consider the following algorithm, which stores just one item in memory at all times. When the first item appears, it is stored in the memory. When the kth item appears, it replaces the item in memory with probability $1/k$. Explain why this algorithm solves the problem.

Problem 2

If you reload the home page of the course website a few times, you might notice that the picture of the mathematician changes. Ever wonder how many different pictures there are, and how you can find out?

Consider the following scenario. There are n different pictures, where n is unknown to you. Each time you load the website, it chooses one of the n pictures uniformly at random and displays it.

Suppose we want to design an algorithm that outputs n. The algorithm can load the website and see the displayed picture (it *cannot* view the page source, etc.). For example, the following is a valid algorithm that loads the website 100 times and outputs the number of different pictures displayed:

Algorithm.

```
1. Initialize $S$ to be the empty set. ($S$ will store the different pictures
   seen so far.)
2. Repeat 100 times:
     i. Load the website.
    ii. If the displayed picture is not in $S$, then add it to $S$.
3. Output the size of $S$.
```

This algorithm would probably find the right answer for our course website because we don't have too many different pictures. But if n were larger, say 50, then it's not so clear how well this algorithm would work. And if n were more than 100, then this algorithm cannot possibly succeed.

We want to design an algorithm that succeeds (outputs the correct value of n) with probability at least $\frac{1}{2}$, *regardless of the value of* n. Feel free to stop and think about how you might design such an algorithm. In the rest of this problem, we'll describe one solution and ask you to analyze it.

The following algorithm keeps loading the website, keeping track of the different pictures that have been seen so far. If s is the number of different pictures that have been seen, the algorithm stops if no new pictures have been seen in the last $4s$ website visits.

Algorithm.

```
1. Initialize $S$ to be the empty set. ($S$ will store the different pictures seen so far.)
2. $s \leftarrow 0$. ($s$ represents the size of $S$.)
3. $Count \leftarrow -1$.
4. Repeat until $Count > 4s$:
    i. Load the website.
    ii. If the displayed picture is not in $S$: add it to $S$, $s \leftarrow s + 1$, and $Count \leftarrow 0$. Else, $Count \leftarrow Count + 1$.
5. Output the size of $S$.
```

a. Assume $s < n$. Given that at some point the algorithm has observed s different pictures, prove that the probability that it terminates before observing $s + 1$ pictures is bounded by

$$\left(1 - \frac{n-s}{n}\right)^{4s} \leq e^{-4s(n-s)/n}$$

Hint: Here is an inequalities that might help in your analysis and will be useful throughout the course: for any real number x,

$$1 - x \leq e^{-x}.$$

b. Prove that the probability that the algorithm terminates before observing all the n pictures is bounded by $1/2$.

Hints: (1) recall that for $0 < q < 1$

$$\sum_{i=1}^{k} q^i \leq \frac{q}{1-q}.$$

(2) Partition the sum to two parts, $s \leq n/2$ and $s > n/2$.