COMP 585 **Spring 2024**

Probabilistic Toolkit for Learning and Computing *Instructor: Maryam Aliakbarpour*

Problem Set 1

Instruction:

- Submissions are due no later than **11:59 PM** on **Thursday, January 25, 2024**.
- Please upload your solution in PDF format to the course website on Canvas. You may typeset or upload a scanned version of your handwritten solution. Your solution should be legible and clear. Full credit will be given only to the correct solutions that are easy to read and understand.
- This problem set is designed to test your basic prerequisite knowledge and help you brush up on your previous knowledge. Please do not Google the solutions or use Large Language Models (LLMs) to solve the problems.
- You may collaborate with other class members (group of 2-3 people), but you must mention the names of your collaborators in your solution. The idea behind collaboration is to collectively work towards finding a solution in a fair manner. Here are some guidelines for collaboration:
  - Spend a few hours thinking about the problems before engaging in discussions with others.
  - Do not collaborate with someone who has already solved the problem or is not at the same level of progress as you.
  - Exercise good judgment to prevent one person from providing the solution to another.
  - Collaboration does not permit uploading the same solution file. After discussions with team members, you must independently write your solution. Your write-up should genuinely reflect your understanding of the solution. Avoid sharing your solution with others and refrain from copying solutions, even when working together.
- Please refer to the course syllabus for information regarding the late submission policy.

Problem 1. **(10 points)** Imagine that in the United States, 10% of the population suffers from diabetes. Alex went to a laboratory in Houston for a diabetes blood test and received a positive result. Concerned about the accuracy of this result, Alex researched the lab's reliability. An independent investigation revealed some past inaccuracies in their testing. Specifically, there's a 1% chance of a false positive (indicating diabetes when there is none) and a 5% chance of a false negative (failing to detect diabetes when it is present) in their tests. Given these factors, what is the probability that Alex actually has diabetes?

Problem 2. **(10 points)** Suppose we have an algorithm $\mathcal{A}$ that uses m samples from a distribution P and outputs $\hat{\mu}$ as an approximation for the true mean of the distribution, μ, with probability 2/3. More precisely, we have:

$$\mathbf{Pr}\left[\frac{\mu}{2} \leq \hat{\mu} \leq 2\,\mu\right] \geq 2/3\,.$$

Design an algorithm that uses $O(m \log(1/\delta))$ samples and outputs $\tilde{\mu}$, for which we have:

$$\mathbf{Pr}\left[\frac{\mu}{2} \leq \tilde{\mu} \leq 2\,\mu\right] \geq 1-\delta\,.$$

Include the proof of performance for your algorithm.

Problem 3. **(30 points)** Suppose we have an array A consisting of n distinct elements. Our objective is to sort this array using the randomized quick-sort algorithm, as described below. In this problem, we aim to demonstrate that the randomized quick-sort algorithm operates in $O(n \log n)$ time on average. Let e_i represent the i-th smallest element in the array (note that e_i may or may not be located at position $A[i]$).

Algorithm 1 Randomized quick sort algorithm

```
procedure QUICK-SORT($A$)
    $A_L$ and $A_R \leftarrow$ empty arrays
    $\ell \leftarrow$ a random number in $[n]$
    for $i = 1, \ldots, \text{size}(A)$ do
        if $A[i] < A[\ell]$ then
            Add $A[i]$ to $A_L$.
        else
            Add $A[i]$ to $A_R$.
    QUICK-SORT($A_L$)
    QUICK-SORT($A_R$)
    return concatenation of $A_L + A[\ell] + A_R$.
```

Let e_i denote the i-th element in the sorted version of array A. Now, consider two distinct elements of the array, e_i and e_j, where $i < j$. Let X_{ij} be an indicator variable that denotes whether e_i and e_j were compared during the course of the algorithm (in Line 5). With this definition in mind, please answer the following questions:

a. In the first round of the algorithm (right before invoking the recursions), what is the probability that e_i and e_j will be compared? Additionally, what is the probability that e_i and e_j will never be compared again after this round?

b. Could you bound the expected value of X_{ij} based on the values of i and j?

c. Can you express the running time of the algorithm in terms of X_{ij}? Using the expected value of X_{ij} that you computed earlier, what is the expected time complexity of this quick-sort algorithm?

Problem 4. (50 points) Suppose Rice University has n colleges and has admitted m students for undergraduate studies. On the first day of classes, every student wears a hat, and the hat shouts the name of the student's college (each with probability $1/n$): "GriffinJones", "HuffleRice", "sLovettin", etc.

a. How many students, in expectation, do we need to see before we encounter at least one student from each college?

b. Show that, given a sufficiently large constant c, if there are $m \geq n \ln n + c \cdot n$ students, then the probability of having seen at least one student from each college is at least $1 - e^{-c}$.

c. The school administration is concerned about the unbalanced distribution of students among the colleges. Show that if $m = \Omega(n \log n)$, then with probability at least $1 - 1/n$, the number of students per college will fall within the range:

$$\left(\frac{m}{n} - O\left(\sqrt{\frac{m}{n} \cdot \log n}\right), \frac{m}{n} + O\left(\sqrt{\frac{m}{n} \cdot \log n}\right)\right).$$

d. Show that if $m = n$, then with probability at least $1 - 1/n$, no college will have more than $O\left(\frac{\ln n}{\ln \ln n}\right)$ students, provided that n is sufficiently large.