Logica e Metodi Probabilistici per L'Informatica
Second session of Exercises

Academic year 2021/2022

Instructor: Prof. Stefano Leonardi Teaching Assistant: Federico Fusco

March 15, 2022

Exercise 1. Suppose that we roll ten standard six-sided dice. What is the probability that their sum will be divisible by 6, assuming that the rolls are independent? (*Hint: Consider the situation after rolling all but one of the dice.*)

Solution. The sum of the 10 results is equally likely to be congruent to i modulo 6, for $i = 1, 2, \ldots, 6$, so the answer is $1/6$. To see this, consider for instance the case where the sum of the first 9 tosses is congruent by 2 modulo 6. Then, we have that, after the last toss, the sum is congruent to

$$\begin{cases} 0 & \text{if the last toss was 4} \\ 1 & \text{if the last toss was 5} \\ 2 & \text{if the last toss was 6} \\ 3 & \text{if the last toss was 1} \\ 4 & \text{if the last toss was 2} \\ 5 & \text{if the last toss was 3} \end{cases}$$

One can generalize this argument by induction as follows (the base case is immediate): irregardless of the sum of the first j tosses, the sum of the first $j+1$ tosses is equally likely to belong to any one of the 6 congruence classes.

Exercise 2. Consider the following variation of the min-cut algorithm presented in class. We start with a graph G with n vertices, and we first use the randomized min-cut algorithm to contract the graph down to a graph G_k with $k = \sqrt{n}$ vertices. Next, we make $\ell = \sqrt{n}$ copies of the graph G_k, and run the randomized algorithm independently on each copy of the reduced graph. We output the smallest min-cut set found in all executions.

Hint: before tackling this exercise, make sure to be familiar with the analysis of the randomized algorithm for min-cut.

- What is the probability that the reduced graph G_k has the same cut-set value and the original graph G?

  Solution G_k has the same min-cut size if we don't remove any edge in the cut-set C for the first $\hat{n} = n - \sqrt{n}$ iterations. Let E_j be the event *"the edge contracted in iteration i is not in C"*. Moreover, let $F_i = \cap_{j=1}^{i} E_j$, corresponding to the event *"no edge of C was contracted in the first i iterations"*. Then G_k maintains the same min cut-set of G with probability

$$\begin{aligned}\mathbb{P}(F_{\hat{n}}) &= \mathbb{P}(F_{\hat{n}-1} \cap E_{\hat{n}}) = \mathbb{P}(E_{\hat{n}} \mid F_{\hat{n}-1})\,\mathbb{P}(F_{\hat{n}-1}) \\ &= \mathbb{P}(E_{\hat{n}} \mid F_{\hat{n}-1})\,\mathbb{P}(E_{\hat{n}-1} \mid F_{\hat{n}-2}) \dots \mathbb{P}(E_2 \mid F_1)\,\mathbb{P}(E_1) \\ &\geq \prod_{i=1}^{\hat{n}} \left(1 - \frac{2}{n-i-1}\right) = \frac{(\sqrt{n}-2)(\sqrt{n}-3)}{(n-2)(n-3)} \geq \frac{1}{2n}.\end{aligned}$$

Note that the last inequality holds for $n \geq 10$.

- What is the probability that the algorithm outputs a correct min-cut set? [Hint: For $a << b$ you can use $(1-1/b)^a \leq 1 - \frac{a}{b}$.]

  Solution. We output a correct min-cut set if at least one of the copies is correct. By independence and the hint, all of the copies will fail (i.e. will not recover the minimum cut of G_k) with probability at most

  $$\left(1 - \frac{2}{\sqrt{n}(\sqrt{n}-1)}\right)^{\sqrt{n}} \leq 1 - \frac{2}{\sqrt{n}-1} \leq 1 - \frac{1}{\sqrt{n}}.$$

  Then we will succeed with probability at least $n^{-1/2}$. The algorithm succeeds if the reduced graph G_k still contains the min-cut and if at least the execution on one of the copies is correct, that is with probability at least $0.5 \cdot n^{-1} \cdot n^{-1/2} = 0.5 \cdot n^{-3/2}$.

- Compare the number of contractions and the error probability to running the original algorithm twice and taking the minimum cut-set value.

  Solution. Same order of number of contractions, but running the algorithm twice finds a min-cut set with probability

  $$1 - \left(1 - \frac{2}{n(n-1)}\right)^2$$

  which is asymptotically $O(n^{-2})$. There is a clear improvement in the probability of success, and hence in the runtime if one aims at constant probability via parallel amplification.

Exercise 3 (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 it is uniformly distributed over all the items that we have seen at each time step. Moreover, we want to accomplish this without knowing the total number of items in advance or storing all the items that we see.

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

Solution. For any k, let S_k be the set of the first k elements, and let $E_{j,k}$ the event that at time k the selected element is j, for $j \leq k$. Finally, let F_i be the event that the i^{th} element is selected immediately when it appears. We want to show that $\mathbb{P}(E_{j,k}) = 1/k$ for all k and all $j \leq k$.

$$\begin{aligned}\mathbb{P}(E_{j,k}) &= \mathbb{P}(F_j)\,\mathbb{P}(F_{j+1}^C)\,\mathbb{P}(F_{j+2}^C) \dots \mathbb{P}(F_k^C) \\ &= \frac{1}{j}\left(1 - \frac{1}{j+1}\right)\left(1 - \frac{1}{j+2}\right) \dots \left(1 - \frac{1}{k}\right) \\ &= \frac{1}{j} \cdot \frac{j}{j+1} \cdot \frac{j+1}{j+2} \cdots \frac{k+1}{k} = \frac{1}{k}\end{aligned}$$

Bonus. How would you generalize it to samples of more than one element?