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

Academic year 2021/2022

Instructor: Prof. Stefano Leonardi Teaching Assistant: Federico Fusco

March 8, 2022

Exercise 1. Compute the expected value of a geometric random variable X with parameter $p \in (0,1)$, i.e. such that $\mathbb{P}(X = i) = p(1-p)^{i-1}$ for all $i = 1, 2, \ldots$. What about the number of failures before the first success?

Solution. We solve this problem in two ways. First, we can apply the definition of expected value:

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=0}^{\infty} i \cdot \mathbb{P}(X = i) = \sum_{i=0}^{\infty} i \cdot p(1-p)^{i-1} \\
&= -p \sum_{i=0}^{\infty} \frac{d}{dp}(1-p)^i = -p \cdot \frac{d}{dp}\left[\sum_{i=0}^{\infty}(1-p)^i\right] \\
&= -p \cdot \frac{d}{dp}\left[\frac{1}{p}\right] = \frac{1}{p}.
\end{aligned}$$

Math alert! We can invert the order of summation and derivation because for any $p \in (0,1)$ the series $\sum_{i=0}^{\infty} \frac{d}{dp}(1-p)^i$ converges totally and therefore uniformly in an interval $(p-\delta, p+\delta) \subseteq (0,1)$ for δ small enough and because $\sum_{i=0}^{\infty}(1-p)^i$ converges for all $p \in (0,1)$.

There is a simple way around this calculations, using the following well known property of non-negative random variables: $\mathbb{E}[X] = \sum_{i=0}^{\infty} \mathbb{P}(X > i)$. For the geometric random variable it gives:

$$\mathbb{E}[X] = \sum_{i=0}^{\infty} \mathbb{P}(X > i) = \sum_{i=0}^{\infty} (1-p)^i = \frac{1}{p}.$$

For the last question, it is enough to use the linearity of expectation: the failures are always one less than the total trials, so the expected number of failures is $\frac{1}{p} - 1$.

Exercise 2. Let $n \geq 2$, and let π be a permutation of $\{1, 2, \ldots, n\}$ drawn uniformly at random between all the $n!$ permutations on that set.

- Compute the probability that π does not have fixed points, i.e. $\pi(i) \neq i$, for all $i \in \{1, 2, \ldots, n\}$
- Let Z be the number of fixed points of the permutation, compute $\mathbb{E}[Z]$
- Compute $\mathbb{E}\left[(Z - \mathbb{E}[Z])^2\right]$

Hint: exploit the linearity of $\mathbb{E}$.

Solution. Using the inclusion-exclusion principle, the number of permutations without fixed points can be computed as follows: the total number of permutations, minus the permutations fixing point 1, minus those fixing point 2 ... , plus the permutations that fix points 1 and 2, plus those fixing 1 and 3, minus ... and so no. Notice that there are $\binom{n}{i}$ ways of choosing i elements of the base set, and $(n-i)!$ permutations which keep them fixed (just count the permutations on the remaining elements). All in all, calling A the event that $\pi(i) \neq i$, for all $i \in \{1, 2, \dots, n\}$

$$\mathbb{P}(A) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} \frac{(n-i)!}{n!} = \sum_{i=0}^{n} \frac{(-1)^i}{i!}.$$

The result consists of the first $n+1$ terms of the exponential sum with parameter -1, i.e. for $n \to \infty$, the probability tends to e^{-1}. For the other points, introduce the random variables X_i, which values 1 if $\pi(i) = i$ and 0 otherwise. Clearly $Z = \sum_{i=1}^{n} X_i$, and $\mathbb{E}[X_i] = \mathbb{P}(\pi(i) = i) = \frac{1}{n}$, so $\mathbb{E}[Z] = 1$, simply by linearity of expectation. Finally, consider the last point:

$$\mathbb{E}\left[(Z - \mathbb{E}[Z])^2\right] = \mathbb{E}\left[Z^2\right] - \mathbb{E}[Z]^2 = \mathbb{E}\left[\left(\sum_{i=1}^{n} X_i\right)^2\right] - 1 = \mathbb{E}\left[\sum_{i=1}^{n} X_i^2 + 2\sum_{i<j} X_i X_j\right] - 1$$

$$= \sum_{i=1}^{n} \mathbb{E}\left[X_i^2\right] + 2\sum_{i<j} \mathbb{E}[X_i X_j] - 1 = 2\sum_{i<j} \mathbb{E}[X_i X_j] = 2\sum_{i<j} \frac{1}{n(n-1)} = 1.$$

Note that $\mathbb{E}[X_i X_j]$, cannot be computed by simply multiplying $\mathbb{E}[X_i]$ and $\mathbb{E}[X_j]$, since the two random variables are not independent (consider $n = 2$ as an easy scenario to visualize it).

$$\mathbb{E}[X_i X_j] = \mathbb{P}(X_i = 1, X_j = 1) = \frac{(n-2)!}{n!} = \frac{1}{n(n-1)} \neq \frac{1}{n^2}.$$

Exercise 3. You are tasked with finding the best COVID-19 cure. So far, n candidates have been developed, with codenames $c_1, c_2, \dots, c_n$. To test a single cure, you can add a small amount of it to a vial containing a viral solution. All of them are effective, but might require a different dose to kill the virus: for example, it could take $a_4 = 42$ units of c_4 but only $a_{31} = 6$ units of c_{31}. After putting a certain amount of a cure in a test vial, to decide if the virus has been eliminated, you need to perform a complicated test which requires rare and expensive reagents. Since the world will need a very large amount of the cure, you want to find the most effective one with high precision: i.e., you want the cure $c_i \in \{c_1, \dots, c_n\}$ such that a_i is minimal. Specifically, you already know that d units of any cure will kill the virus and want to find the best one. Describe an algorithm to solve this task, using as few tests as possible.

- Find a deterministic algorithm that uses $O(n \log d)$.
- Find a randomized algorithm that uses $O(n + \log n \cdot \log d)$ tests.

Solution. For every single cure, you can do a binary search with $O(\log d)$ tests. However, doing this results in $O(n \log d)$ tests to find the best cure, so that's too much. Instead, order the cures randomly and do the search for the first one. Then, find out for the second if its needed dose is smaller than that of the first: if yes, do the binary search also there. If not, this cure is clearly worse. Move on through all cures like this, always only investigating the current one if it is better than the best one so far.

The number of binary searches to conduct is the number of times the minimum so-far amount changes in a random ordering of the cures. For each step i, the probability that the i-th number is smaller than all the ones before is exactly $1/i$, given all the doses are actually different. (To see this, try picking the very last one in the sequence first. The probability that it is the overall smallest is $1/n$. For the second to last, which is again picked from the leftover set uniformly at random, it is $1/(n-1)$, and so on.) Therefore, the expected number K of times that we have to conduct a binary search is (harmonic series)

$$\mathbb{E}[K] = \sum_{i=1}^{n} \frac{1}{i} \leq \log n + 1$$

Which makes the overall number of tests T

$$\mathbb{E}[T] = \mathbb{E}[K] \cdot O(\log d) = O(\log n \log d).$$

Note that the extra $O(n)$ additive term in the running time is given by the fact that at each iteration i the algorithm verifies if the i^{th} cure is better than the best cure observed so far.