Exercises # 3: Conditioning

Léo Neufcourt

September 27, 2019

Please do all problems. Due at the beginning of class on September 27. Python codes must be sent by email before September 27, 12:40pm.

Exercise 1. *We consider a square polygon ABCD and its center O and denote* $\Gamma := \{A, B, C, D, O\}$. *A tick is moving randomly in* Γ *by jumping from one point to another, with the only constraint that if a jump joins two vertices of the square ABCD then they most be adjacent. For instance, a tick in A can jump in B, D or O; a tick in O can jump to A, B, C or D. At every step all allowed moves have the same probability. The tick cannot stay at the same location between two steps. At the beginning, i.e. before its first jump, the tick is in O. For every integer n we denote* O_n *the event "the tick is at O after its* n^{th} *jump". We denote* $p_n := \mathbb{P}(O_n)$ *the probability, with* $p_0 = 1$. *We define similarly the events* A_n, B_n, C_n *and* D_n.

(i) *Compute* p_1 *and* p_2

(ii) *For every integer* $n \geq 1$ *show (e.g. by recurrence over n) that*

$$P(A_n) = P(B_n) = P(C_n) = P(D_n).$$

(iii) *Show that for every integer n we have*

$$p_{n+1} = \frac{1}{3}(1 - p_n)$$

and deduce the value of p_n *for* $n \in \mathbb{N}$

(iv) *Based on the previous questions – what proportion of time does the tick spend on each of the points of* Γ*?*

Exercise 2. *Let $n \geq 2$ and $X_1, X_2, ..., X_n$ be independent random variables with the same distribution and suppose that they have finite expectation $m := \mathbb{E}[X_1]$. Let $S_n := \sum_{i=1}^n X_i$.*

1. *Compute $\mathbb{E}[S_n|X_i]$ for $1 \geq i \geq n$.*
2. *Compute $\mathbb{E}[X_i|S_n]$ for $1 \geq i \geq n$.*
3. *(Bonus) Suppose that $n = 2$ and that X_i have a common density f. What is the conditional density of X_1 given S_2?*

Exercise 3. *Show that if X and Y are two independent random variables with the same distribution, we have*

$$\mathbb{E}[X - Y|X + Y] = 0$$

.

Exercise 4 (Python)**.** *Plot the expectation of $Y_n := \max_{i=1,...,n} X_i$, when X_i are i.i.d. standard normal variables, as a function of $1 \leq n \leq 20$. You are encouraged to use the law of large numbers to approximate the expectations (thus avoiding formal calculations), but make sure you keep a relative approximation errors below 1%.*

Hint: *Once you have computed the expectations $\mathbb{E}[Y_n]$ for all n and store them in a* `numpy` *array* `y` *of size 20, you can use the library* `matplotlib` *for plotting. A minimal template:*

```python
import numpy as np
import matplotlib.pyplot as plt

# compute y ...

x = np.arange(1, 21)
fig, ax = plt.subplots(figsize = (10, 10))
ax.plot(x, y)
```