

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 must 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 ?*

Solution 1.

(i) Starting in O the tick cannot be in O after its first jump, so $p_1 = 0$. After the first jump, the tick is in A, B, C or D with the same probability. In every case, it will come back to O after the second jump with probability $\frac{1}{3}$, thus $p_2 = \frac{1}{3}$.

(Formally:

$$p_2 = \mathbb{P}(O_2) = \mathbb{P}(O_2|A_1)\mathbb{P}(A_1) + \mathbb{P}(O_2|B_1)\mathbb{P}(B_1) + \mathbb{P}(O_2|C_1)\mathbb{P}(C_1) + \mathbb{P}(O_2|D_1)\mathbb{P}(D_1)$$

from the total probability formula, with $\mathbb{P}(A_1) = \mathbb{P}(B_1) = \mathbb{P}(C_1) = \mathbb{P}(D_1) = \frac{1}{4}$ and $\mathbb{P}(O_2|A_1) = \mathbb{P}(O_2|B_1) = \mathbb{P}(O_2|C_1) = \mathbb{P}(O_2|D_1) = \frac{1}{3}$.)

(ii) Let us show the property

$$\mathcal{H}_n : P(A_n) = P(B_n) = P(C_n) = P(D_n)$$

by recurrence over n .

We have seen that \mathcal{H}_1 is true. Let us suppose that \mathcal{H}_n is true and prove \mathcal{H}_{n+1} . The events A_n, B_n, C_n, D_n and O_n are disjoint and their union covers all the possible outcomes, so we can use the formula of total probability

$$\begin{aligned} p_{n+1} &= \mathbb{P}(O_{n+1}) \\ &= \mathbb{P}(O_{n+1}|A_n)\mathbb{P}(A_n) + \mathbb{P}(O_{n+1}|B_n)\mathbb{P}(B_n) + \mathbb{P}(O_{n+1}|C_n)\mathbb{P}(C_n) + \mathbb{P}(O_{n+1}|D_n)\mathbb{P}(D_n). \end{aligned}$$

Now,

$$\begin{aligned} P(A_{n+1}|A_n) &= P(A_{n+1}|C_n) = 0, \\ P(A_{n+1}|B_n) &= P(A_{n+1}|D_n) = \frac{1}{3}, \\ P(A_{n+1}|O_n) &= \frac{1}{4}. \end{aligned}$$

whence

$$\begin{aligned}
P(A_{n+1}) &= \frac{1}{3}P(B_n) + \frac{1}{3}P(D_n) + \frac{1}{4}P(O_n) \\
&= \frac{2}{3}P(A_n) + \frac{1}{4}P(O_n) \\
&= \frac{2}{3}P(B_n) + \frac{1}{4}P(O_n) = P(B_{n+1}) \\
&= \frac{2}{3}P(C_n) + \frac{1}{4}P(O_n) = P(C_{n+1}) \\
&= \frac{2}{3}P(D_n) + \frac{1}{4}P(O_n) = P(D_{n+1}),
\end{aligned}$$

using \mathcal{H}_n . This shows \mathcal{H}_{n+1} , and we conclude that \mathcal{H}_n is true for every $n \geq 1$.

(iii) We know from (ii) that $p_{n+1} = 1 - 4P(A_{n+1})$ and $p_n = 1 - 4P(A_n)$. Follow $P(A_{n+1}) = \frac{2}{3}P(A_n) + \frac{1}{4}p_n = \frac{2}{3} \times \frac{1}{4}(1 - p_n) + \frac{1}{4}p_n$ and

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

This a classical arithmetico-geometric sequence. We find possible candidates for the limit by solving the equation $\ell = \frac{1}{3}(1 - \ell)$ which has a unique solution $\frac{1}{4}$. We consider next the sequence $v_n := p_n - \ell$, which satisfies the recurrence relation $v_{n+1} = -\frac{1}{3}v_n$ for every $n \geq 0$. The general term of this geometric sequence is $v_n = v_0(-\frac{1}{3})^n = \frac{3}{4}(-\frac{1}{3})^n, n \geq 0$, from which $p_n = \frac{1}{4} + \frac{3}{4}(-\frac{1}{3})^n$.

(iv) The proportion of time spent in O after n steps is

$$\frac{p_0 + p_1 + \dots + p_n}{n+1} \xrightarrow{n \rightarrow \infty} \ell = \frac{1}{4}$$

as the Cesaro sum of the converging sequence $p_n \xrightarrow{n \rightarrow \infty} \ell$.

Hence the proportion of time (at the limit) spent in O is $\frac{1}{4}$, and by symmetry the proportion of time spent on each of A , B , C and D is $\frac{3}{16}$.

Exercise 2. Let $n \geq 2$ and X_1, X_2, \dots, 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 $1 \leq i \leq n$.

2. Compute $\mathbb{E}[X_i|S_n]$ for $1 \leq i \leq n$.
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 ?

Solution 2. 1.

$$\mathbb{E}[S_n|X_i] = \mathbb{E}\left[\sum_{j=1}^n X_j|X_i\right] \quad (1)$$

$$= \sum_{j=1}^n \mathbb{E}[X_j|X_i] \quad (2)$$

$$= \mathbb{E}[X_i|X_i] + \sum_{j \neq i} \mathbb{E}[X_j|X_i] \quad (3)$$

$$= X_i + \sum_{j \neq i} \mathbb{E}[X_j] \quad (4)$$

$$= X_i + (n-1)m, \quad (5)$$

where we used that the X_j are independent (4) and have the same distribution (5).

2. By symmetry (see also next problem) the terms $\mathbb{E}[X_i|S_n]$ are the same for all i , thus

$$\begin{aligned} \mathbb{E}[X_i|S_n] &= \frac{1}{n} \sum_{j=1}^n \mathbb{E}[X_j|S_n] \\ &= \frac{1}{n} \mathbb{E}\left[\sum_{j=1}^n X_j|S_n\right] \\ &= \frac{1}{n} \mathbb{E}[S_n|S_n] \\ &= \frac{S_n}{n} \end{aligned}$$

3. Let us write $X := X_1$ and $Y := X_2$. From Solution 2 we have the joint distribution

$$F_{X, X+Y}(x, z) = \mathbb{E}[F_X(\min(x, z - Y))].$$

In the case of continuous random variables this is

$$\begin{aligned}
 & \int_{-\infty}^{+\infty} F_X(\min(x, z-y))f_X(y)dy \\
 = & F_X(x) \int_{-\infty}^{z-x} f_X(y)dy + \int_{z-x}^{+\infty} F_X(z-y)f_X(y)dy \\
 = & F_X(x)F_X(z-x) + \int_{z-x}^{+\infty} F_X(z-y)f_X(y)dy
 \end{aligned}$$

and the joint density of X and $X+Y$ is thus

$$\begin{aligned}
 f_{X,X+Y}(x, z) &= \frac{\partial^2}{\partial x \partial z} F_{X,X+Y}(x, z) \\
 &= \frac{\partial^2}{\partial x \partial z} (F_X(x)F_X(z-x)) + \frac{\partial^2}{\partial z \partial x} \int_{z-x}^{+\infty} F_X(z-y)f_X(y)dy \\
 &= \frac{\partial}{\partial x} (F_X(x)f_X(z-x)) + F_X(x) \frac{\partial}{\partial z} f_X(z-x) \\
 &= f_X(x)f_X(z-x).
 \end{aligned}$$

Finally, the conditional density

$$f_{X|X+Y}(x|z) = \frac{f_{X,X+Y}(x, z)}{f_{X+Y}(z)} = \frac{f_X(x)f_Y(z-x)}{\int_{-\infty}^{+\infty} f_X(x)f_Y(z-x)}.$$

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

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

Solution 3. The devil is in the detail. The question is to show $\mathbb{E}[X|X+Y] = \mathbb{E}[Y|X+Y]$. This is true because X and Y have the same **conditional** distribution given $X+Y$, which follows from a symmetry argument: for instance, denoting $Z := X+Y$ and $F_X, F_Y, F_{X,Z}, F_{Y,Z}$ the respective (joint)

c.d.f. of X , Y , (X, Z) , (Y, Z) , we have

$$F_{X,Z}(x, z) = \mathbb{P}(X \leq x, Z \leq z) \quad (6)$$

$$= \mathbb{E}[\mathbb{P}(X \leq x, Z \leq z|Y)] \quad (7)$$

$$= \mathbb{E}[\mathbb{P}(X \leq x, X + Y \leq z|Y)] \quad (8)$$

$$= \mathbb{E}[\mathbb{P}(X \leq x, X \leq z - Y|Y)] \quad (9)$$

$$= \mathbb{E}[F_X(\min(x, z - Y))] \quad (10)$$

$$= \mathbb{E}[F_Y(\min(x, z - Y))] \quad (11)$$

$$= \mathbb{E}[F_Y(\min(x, z - X))] \quad (12)$$

$$= F_{Y,Z}(x, z) \quad (13)$$

where (7) and (8) follow from X and Y having the same distribution, and (9) from a symmetry argument. Thus, we also have $F_{X|X+Y} = F_{Y|X+Y}$.

Exercise 4 (Python). Plot the expectation of $Y_n := \max_{i=1,\dots,n} X_i$, 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:

```
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)
```

Solution 4.

```
import numpy as np
import matplotlib
matplotlib.rcParams['mathtext.fontset'] = 'cm'
from matplotlib import pyplot as plt
```

```

n_MC = 1000
n_min = 1
n_max = 21
n_test = 5

def simulations(n_min = 1, n_max = 21, n_MC = 1000, n_test = 5)
:
    x = range(n_min, n_max)
    y = np.zeros((n_max - n_min, n_test))
    for k, n in enumerate(x):
        z = np.zeros((n_MC, n_test))
        for i in range(n_MC):
            for j in range(n_test):
                z[i, j] = np.random.normal(size = n).max()
        y[k] = z.mean(axis = 0)
    return x, y

for n_MC in [1000, 5000, 10000, 50000]:
    x, y = simulations(n_MC = n_MC)
    fig, ax = plt.subplots(figsize = (20, 20))
    ax.set_xscale('log')
    ax.plot(x, y.mean(axis = 1), color = 'dodgerblue', label =
            'estimate')
    ax.fill_between(x, y[:, 0] * .99, y[:, 0] * 1.01, color = '
            dodgerblue', alpha = .3, label = r'$1\% \_band$')
    ax.plot(x, y, marker = '+', label = 'trials')
    ax.set_xlabel(r'$n$', fontsize = 40)
    ax.text(.5, .95, r'$y(n)$', fontsize = 40, transform = ax.
            transAxes, ha = 'center', va = 'center')
    ax.text(.5, .9, '(expectation_of_the_maximum_of_$n$
            independent_standard_Gaussians)', fontsize = 20,
            transform = ax.transAxes, ha = 'center', va = 'center')
    ax.tick_params(labelsize = 30)
    h, l = ax.get_legend_handles_labels()
    ax.legend(h, ['estimate'] + [f"trial_{1+k}" for k in
            range(n_test)] + [r'$1\% \_band$'], loc = 'lower_right',
            fontsize = 40)
    plt.savefig('max_n_gaussians_values.pdf')
    plt.show()

fig, ax = plt.subplots(figsize = (20, 10))
ax.set_yscale('log')
ax.plot(x, np.abs(y.std(axis = 1) / y.mean(axis = 1)), 'o',
        label = 'estimate')
ax.fill_between(x, -1, .01, alpha = .3, color = 'dodgerblue',

```

```

    label = r'$1\%\_band$')
ax.set_xlabel(r'$n$', fontsize = 40)
ax.text(.33, .85, r'$\frac{\delta y(n)}{y(n)}$', fontsize = 60,
        transform = ax.transAxes)
ax.tick_params(labelsize = 30)
ax.legend(fontsize = 40, loc = 'upper_right')
plt.savefig('max_n_normals_precisions.pdf')
### of course, the value at x = 1 is y = 0...

```

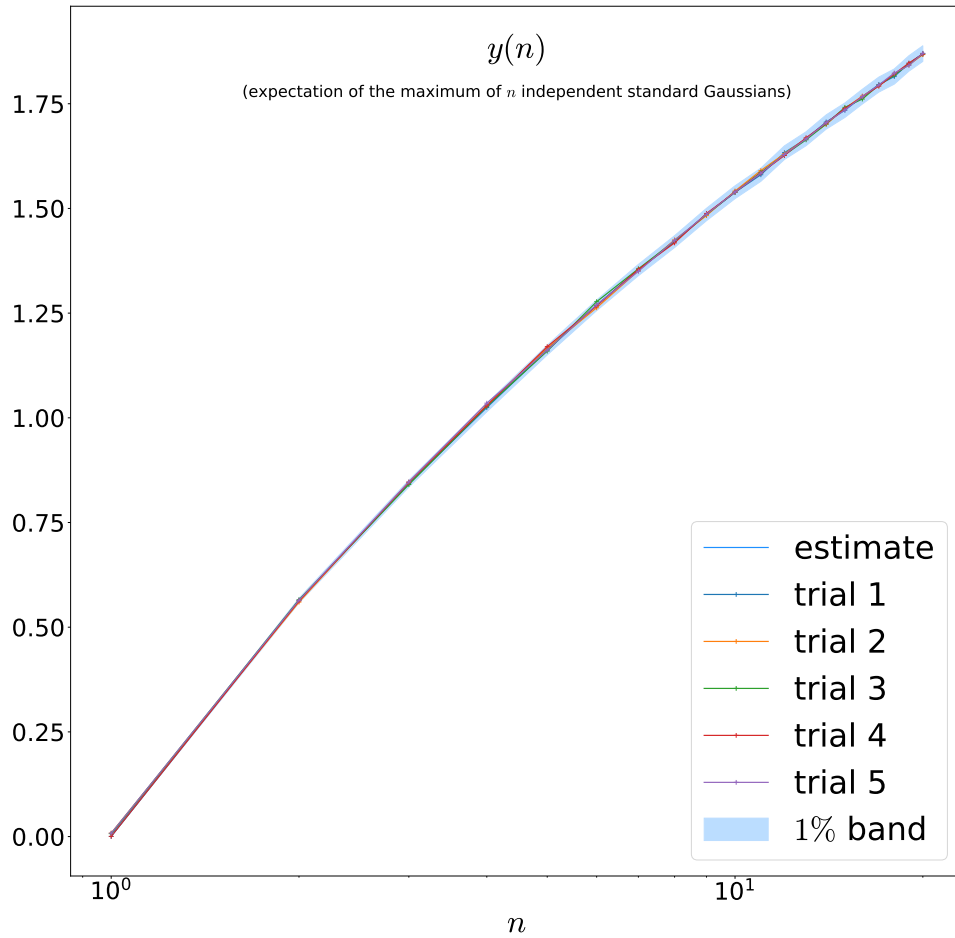



Figure 1: Estimates of the expectation of n independent standard Gaussians for $n = 1, \dots, 20$, obtained as the average of 5 trials using 50,000 samples each.

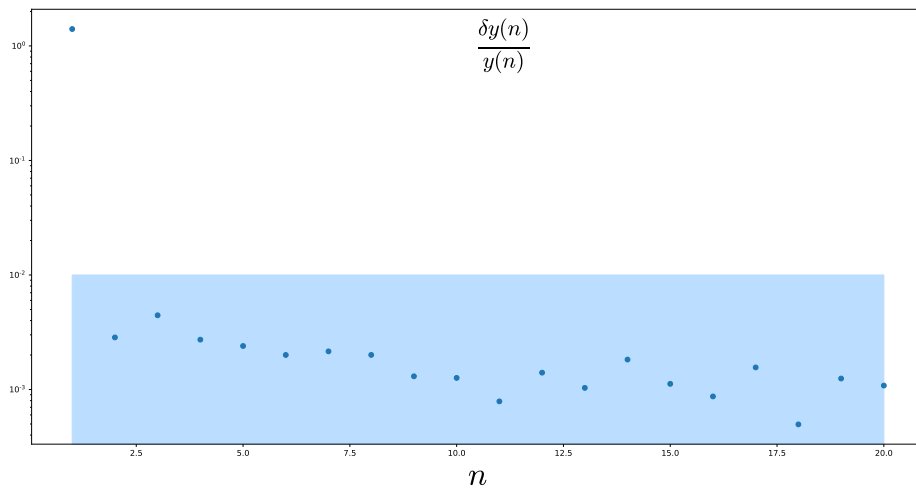


Figure 2: Relative errors on the estimates of the expectation of n independent standard Gaussians for $n = 1, \dots, 20$.