



UNIVERSITY OF LIÈGE

---

## Project 1

---

Information and coding theory

Maxime MEURISSE (s161278)  
François ROZET (s161024)

Master in Computer Science and Engineering  
Academic year 2019-2020

# Information measures

## Exercises by hand

*Remark.* Only theoretical formulas and final results are presented in this section. All developments are detailed in appendix of this report in order to show that the application of the stated formulas does lead to the results presented.

1. Since  $\mathcal{X}$  and  $\mathcal{Y}$  are independent random variables, their joint probability distribution is equal to the (dyadic) product of their marginal probability distributions.

$$\mathcal{X} \perp \mathcal{Y} \Leftrightarrow P(\mathcal{X}, \mathcal{Y}) = P(\mathcal{X})P(\mathcal{Y}) \quad (1)$$

This yields Table 1.

		$Y_j$			
		0	1	2	3
$X_i$	0	1/8	1/16	1/32	1/32
	1	1/8	1/16	1/32	1/32
	2	1/8	1/16	1/32	1/32
	3	1/8	1/16	1/32	1/32

Table 1 – Joint probability distribution  $P(\mathcal{X} = X_i, \mathcal{Y} = Y_j)$ .

As the sum of  $\mathcal{X}$  and  $\mathcal{Y}$ ,  $\mathcal{S}$  takes value in  $\{0, 1, 2, 3, 4, 5, 6\}$ . By definition of the marginal probability,

$$\begin{aligned} P(\mathcal{S} = S_i) &= \sum_{X_j, Y_k} P(\mathcal{S} = S_i, \mathcal{X} = X_j, \mathcal{Y} = Y_k) \\ &= \sum_{X_j, Y_k} P(\mathcal{S} = S_i | \mathcal{X} = X_j, \mathcal{Y} = Y_k) P(\mathcal{X} = X_j, \mathcal{Y} = Y_k) \end{aligned}$$

But  $\mathcal{S}$  is a (deterministic) function of  $\mathcal{X}$  and  $\mathcal{Y}$ . Therefore,  $P(\mathcal{S} = S_i | \mathcal{X} = X_j, \mathcal{Y} = Y_k)$  is 1 if  $S_i = X_j + Y_k$  and 0 otherwise.

$$P(\mathcal{S} = S_i) = \sum_{X_j, Y_k | S_i = X_j + Y_k} P(\mathcal{X} = X_j, \mathcal{Y} = Y_k) \quad (2)$$

After computation by hand, we obtain Table 2.

$S_i$	0	1	2	3	4	5	6
$P(\mathcal{S} = S_i)$	1/8	3/16	7/32	1/4	1/8	1/16	1/32

Table 2 – Marginal probability distribution of  $\mathcal{S}$ .

Using the same reasoning for  $\mathcal{Z} = 1(\mathcal{X} = \mathcal{Y})$ ,

$$P(\mathcal{Z} = Z_i) = \sum_{X_j, Y_k | Z_i = 1(X_j = Y_k)} P(\mathcal{X} = X_j, \mathcal{Y} = Y_k) \quad (3)$$

and

$Z_i$	0	1
$P(\mathcal{Z} = Z_i)$	3/4	1/4

Table 3 – Marginal probability distribution of  $\mathcal{Z}$ .

2. By definition, the entropy  $H$  of a random variable  $\mathcal{X}$  is the expectation of the information  $h$  given by the realisation of  $\mathcal{X}$ .<sup>1</sup>

$$\begin{aligned} H(\mathcal{X}) &= \mathbb{E}_{\mathcal{X}}(h(\mathcal{X})) \\ &= \mathbb{E}_{\mathcal{X}}(-\log_2 P(\mathcal{X})) \\ &= -\sum_{X_i} P(X_i) \log_2 P(X_i) \end{aligned} \quad (4)$$

Replacing  $\mathcal{X}$  by  $\mathcal{Y}$ ,  $\mathcal{S}$  and  $\mathcal{Z}$ , one obtains Table 4.

.	$\mathcal{X}$	$\mathcal{Y}$	$\mathcal{S}$	$\mathcal{Z}$
$H(\cdot)$	2	1.75	2.5887	0.8113

Table 4 – Marginal entropies of  $\mathcal{X}$ ,  $\mathcal{Y}$ ,  $\mathcal{S}$  and  $\mathcal{Z}$ .

3. The joint entropy of a random variables pair  $(\mathcal{X}, \mathcal{Y})$  is (as well) the expectation of the information given by the realisation of such pair. With the same development as before,

$$H(\mathcal{X}, \mathcal{Y}) = -\sum_{X_i, Y_j} P(X_i, Y_j) \log_2 P(X_i, Y_j) \quad (5)$$

which gives

.	$(\mathcal{X}, \mathcal{Y})$	$(\mathcal{X}, \mathcal{S})$	$(\mathcal{Y}, \mathcal{Z})$	$(\mathcal{S}, \mathcal{Z})$
$H(\cdot)$	3.75	3.75	2.5613	2.8789

Table 5 – Some joint entropies.

4. Following the same logic, the conditional entropy of  $\mathcal{X}$  given  $\mathcal{Y}$  is the expectation of the *conditional* information given by the realisation of a  $(\mathcal{X}, \mathcal{Y})$  pair.

$$\begin{aligned} H(\mathcal{X} | \mathcal{Y}) &= \mathbb{E}_{\mathcal{X}, \mathcal{Y}}(h(\mathcal{X} | \mathcal{Y})) \\ &= \mathbb{E}_{\mathcal{X}, \mathcal{Y}}(-\log_2 P(\mathcal{X} | \mathcal{Y})) \\ &= \mathbb{E}_{\mathcal{X}, \mathcal{Y}}\left(-\log_2 \frac{P(\mathcal{X}, \mathcal{Y})}{P(\mathcal{Y})}\right) \\ &= \mathbb{E}_{\mathcal{X}, \mathcal{Y}}(-\log_2 P(\mathcal{X}, \mathcal{Y}) + \log_2 P(\mathcal{Y})) \\ &= \mathbb{E}_{\mathcal{X}, \mathcal{Y}}(-\log_2 P(\mathcal{X}, \mathcal{Y})) - \mathbb{E}_{\mathcal{X}, \mathcal{Y}}(-\log_2 P(\mathcal{Y})) \\ &= H(\mathcal{X}, \mathcal{Y}) - H(\mathcal{Y}) \end{aligned} \quad (6)$$

---

<sup>1</sup>From now on, for the sake of conciseness, the probability of events (realisations) of the type  $\mathcal{X} = X_i$  will be simply denoted  $P(X_i)$ . For example  $P(\mathcal{X} = X_i)$  will now be denoted  $P(X_i)$ .

.	$(\mathcal{X}   \mathcal{Y})$	$(\mathcal{Z}   \mathcal{X})$	$(\mathcal{S}   \mathcal{X})$	$(\mathcal{S}   \mathcal{Z})$
$H(\cdot)$	2.0	0.7246	1.75	2.0676

Table 6 – Some conditional entropies.

5. Using the same development,

$$\begin{aligned}
H(\mathcal{X}, \mathcal{Y} | \mathcal{Z}) &= - \sum_{X_i, Y_j, Z_k} P(X_i, Y_j, Z_k) \log_2 P(X_i, Y_j | Z_k) \\
&= - \sum_{X_i, Y_j, Z_k} P(X_i, Y_j, Z_k) \log_2 \frac{P(X_i, Y_j, Z_k)}{P(Z_k)} \\
&= H(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) - H(\mathcal{Z})
\end{aligned} \tag{7}$$

After the computation of  $P(\mathcal{X}, \mathcal{Y}, \mathcal{S})$ , we have

.	$(\mathcal{X}, \mathcal{Y}   \mathcal{S})$	$(\mathcal{S}, \mathcal{Y}   \mathcal{X})$
$H(\cdot)$	1.1613	1.75

Table 7 – Some conditional joint entropies.

6. The mutual information  $I$  of two random variables  $\mathcal{X}$  and  $\mathcal{Y}$  is the expectation of the *loss* of information given by  $\mathcal{X}$  when knowing  $\mathcal{Y}$ .

$$\begin{aligned}
I(\mathcal{X}; \mathcal{Y}) &= \mathbb{E}_{\mathcal{X}, \mathcal{Y}} (h(\mathcal{X}) - h(\mathcal{X} | \mathcal{Y})) \\
&= \mathbb{E}_{\mathcal{X}, \mathcal{Y}} (h(\mathcal{X})) - \mathbb{E}_{\mathcal{X}, \mathcal{Y}} (h(\mathcal{X} | \mathcal{Y})) \\
&= H(\mathcal{X}) - H(\mathcal{X} | \mathcal{Y})
\end{aligned} \tag{8}$$

giving

.	$(\mathcal{X}; \mathcal{Y})$	$(\mathcal{X}; \mathcal{S})$	$(\mathcal{Y}; \mathcal{Z})$	$(\mathcal{S}; \mathcal{Z})$
$I(\cdot)$	0	0.8387	0	0.5211

Table 8 – Some mutual information(s).

7. The conditional mutual information  $I$  of two random variables  $\mathcal{X}$  and  $\mathcal{Y}$  given a third one  $\mathcal{Z}$  is the expectation of the *loss* of information given by  $\mathcal{X}$  knowing  $\mathcal{Z}$  when also knowing  $\mathcal{Y}$ .

$$\begin{aligned}
I(\mathcal{X}; \mathcal{Y} | \mathcal{Z}) &= \mathbb{E}_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}} (h(\mathcal{X} | \mathcal{Z}) - h(\mathcal{X} | \mathcal{Y}, \mathcal{Z})) \\
&= \mathbb{E}_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}} (h(\mathcal{X} | \mathcal{Z})) - \mathbb{E}_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}} (h(\mathcal{X} | \mathcal{Y}, \mathcal{Z})) \\
&= H(\mathcal{X} | \mathcal{Z}) - H(\mathcal{X} | \mathcal{Y}, \mathcal{Z}) \\
&= H(\mathcal{X} | \mathcal{Z}) - [H(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) - H(\mathcal{Y}, \mathcal{Z})]
\end{aligned} \tag{9}$$

yielding

.	$(\mathcal{X}; \mathcal{Y}   \mathcal{S})$	$(\mathcal{S}; \mathcal{Y}   \mathcal{X})$
$I(\cdot)$	1.1613	1.75

Table 9 – Some conditional mutual information(s).

## Computer-aided exercises

The following functions have been implemented using the Python language and, more specifically, the library `numpy`. One can find them in the file `entro.py`.

It should be noted that, since it is always possible to compute marginal and conditional probability distributions from the joint one, implemented functions *only* requires the latter as argument.

8. As one can see in equation (4), the entropy can be computed solely based on the marginal probability distribution. Furthermore, one can compute such sum as the inner (dot) product of the information vector  $-\log_2 P(\mathcal{X})$  and the marginal probability vector  $P(\mathcal{X})$  which we implemented as the `entropy` function.

More specifically, the information vector is computed as the opposite  $(-)$  of the *element-wise*  $\log_2$  of the marginal probability vector. This operation has been implemented as a function named `information`. However, sometimes the probability  $p$  of an event is null. Analytically, the information of such event is not defined which causes problems numerically ( $\log_2 0$  isn't defined). But, we know that

$$\lim_{p \rightarrow 0} p \log p = 0$$

Therefore, the function `entropy` actually doesn't consider the null probability element of the probability vector given to him.

Intuitively, the entropy measures the *average information* conveyed by the realisation of a random variable. One could also see it as the *uncertainty* or *disorder* associated to a random variable.

9. Entropy and joint entropy are essentially the *same* quantities. Indeed, the joint probability of  $n$  1-dimensional random variables is equivalent to the marginal probability of one  $n$ -dimensional random variable. Therefore, as `joint_entropy` function, we *reshape* the joint probability distribution matrix into a vector and then apply the `entropy` function to it.

$$P(\mathcal{X}, \mathcal{Y}, \dots) \equiv P(\mathcal{W}) \text{ with } \mathcal{W} = (\mathcal{X}, \mathcal{Y}, \dots)$$

*Remark.* As it has been implemented, the function `joint_entropy` can actually take a matrix of any dimension as input. This will come handy later on.

10. The function `conditional_entropy` has been implemented to replicate the equality (6), *i.e.*

$$H(\mathcal{X} | \mathcal{Y}) = H(\mathcal{X}, \mathcal{Y}) - H(\mathcal{Y})$$

Since both `joint_entropy` and `entropy` have already been implemented, it is only necessary to compute  $P(\mathcal{Y})$  based on the given distribution  $P(\mathcal{X}, \mathcal{Y})$ . But we know that,

$$P(\mathcal{X}) = \sum_{\mathcal{Y}} P(\mathcal{X}, \mathcal{Y})$$

which is fairly easy to compute using `numpy` matrices.

11. The implementation of the `mutual_information` function is based on equality (8), *i.e.* the difference of the (marginal) entropy and the conditional entropy which both have been implemented before.

Thanks to the mutual information measure, it is possible to quantify the influence of one variable on the other. Especially, if these variables are *independent*, their mutual information is necessarily null, meaning that knowing one of these variables doesn't provide any information about the other one.

12. Concerning the `cond_joint_entropy` function, we have implemented the equality (7) which can be computed *exactly* like `conditional_entropy`.

Concerning the `cond_mutual_information` function, we implemented equality (9), *i.e.*

$$I(\mathcal{X}, \mathcal{Y} | \mathcal{Z}) = H(\mathcal{X} | \mathcal{Z}) + H(\mathcal{Y} | \mathcal{Z}) - H(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$$

Once more this is fairly easy to implement using previously implemented functions.

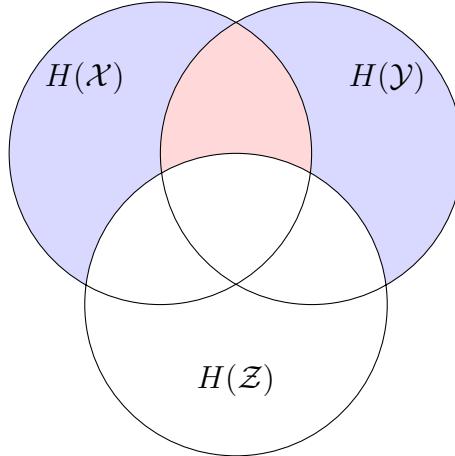


Figure 1 – Entropy Venn diagram. In *red*  $I(\mathcal{X}; \mathcal{Y} | \mathcal{Z})$ , in *blue*  $\cup$  *red*  $H(\mathcal{X}, \mathcal{Y} | \mathcal{Z})$ .

13. Using<sup>2</sup> the `numpy` function `random.choice` we generated  $n$ -samples of  $\mathcal{X}$  and  $\mathcal{Y}$  based on their respective probability distributions. Since  $\mathcal{S}$  and  $\mathcal{Z}$  are deterministic functions of  $\mathcal{X}$  and  $\mathcal{Y}$ , we calculated their value afterwards.

With these samples, we derived several *frequency* (empirical probability) distributions, including the joint frequency distribution of  $\mathcal{X}$ ,  $\mathcal{Y}$ ,  $\mathcal{S}$  and  $\mathcal{Z}$  as a 4-dimensional matrix. This matrix is especially useful since it allows to quickly determine other frequency distributions through dimension reduction.

Then, we simply applied the implemented functions to obtain the following results :

$$\begin{array}{llll} H(\mathcal{X}) = 1.999 & H(\mathcal{Y}) = 1.743 & H(\mathcal{S}) = 2.592 & H(\mathcal{Z}) = 0.805 \\ H(\mathcal{X}, \mathcal{Y}) = 3.742 & H(\mathcal{X}, \mathcal{S}) = 3.742 & H(\mathcal{Y}, \mathcal{Z}) = 2.548 & H(\mathcal{S}, \mathcal{Z}) = 2.865 \\ H(\mathcal{X} | \mathcal{Y}) = 1.999 & H(\mathcal{Z} | \mathcal{X}) = 0.709 & H(\mathcal{S} | \mathcal{X}) = 1.742 & H(\mathcal{S} | \mathcal{Z}) = 2.060 \\ H(\mathcal{X}, \mathcal{Y} | \mathcal{S}) = 1.150 & H(\mathcal{S}, \mathcal{Y} | \mathcal{X}) = 1.742 & & \\ I(\mathcal{X}, \mathcal{Y}) = 0.001 & I(\mathcal{X}, \mathcal{S}) = 0.850 & I(\mathcal{Y}, \mathcal{Z}) = 0.001 & I(\mathcal{S}, \mathcal{Z}) = 0.531 \\ I(\mathcal{X}, \mathcal{Y} | \mathcal{S}) = 1.150 & I(\mathcal{S}, \mathcal{Y} | \mathcal{X}) = 1.742 & & \end{array}$$

As one can see, these values are quite close to the theoretical ones. This is certainly due to the fact that we used a quite high (10 000) number of samples. If we were to lower this number, say to 50, the precision of our estimations would decrease significantly.

---

<sup>2</sup>All codes and results are available in the file `entro.py`.

## Designing informative experiments

14. Let  $\mathcal{X}_1$ ,  $\mathcal{X}_2$  and  $\mathcal{X}_3$  denote random variables associated respectively with the three squares of this subgrid. Given rule (a), we know that any tuple  $(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)$  such that  $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3 \in \{1, 2, \dots, 9\}$  and  $\mathcal{X}_1 \neq \mathcal{X}_2 \neq \mathcal{X}_3$  defines a valid subgrid. Furthermore, we also know that  $\mathcal{X}_1 = 4$ .

A priori, all valid subgrids have the same realisation probability  $p$  and all others have a null realisation probability. Therefore,

$$\sum_{i=1}^N p = 1 \quad (10)$$

with  $N$  the number of possible valid subgrids.

This number is equal to the number of ways to draw *sequentially* 2 different objects ( $\mathcal{X}_2$  and  $\mathcal{X}_3$ ) among 8 (1 to 9 but 4), i.e. the *arrangements* of 2 within 8.

$$N = A_8^2 = \frac{8!}{(8-2)!} = 8 \cdot 7 = 56$$

Finally, keeping in mind that only  $N$  subgrids have a non-null realisation probability, we compute the entropy  $H^3$  of this subgrid through its definition :

$$H(\text{subgrid}) = - \sum_i p_i \log_2 p_i = - \sum_{i=1}^N p \log_2 p = - \log_2 \frac{1}{N} = \log_2 N = 5.8074 \quad (11)$$

15. With respect to the previous subgrid, adding the constraint  $r_1 = 14$  drastically reduces the number of valid subgrids. Indeed, among the 56 previous solutions only those that satisfies the rule (b) are still considered valid.

$$\mathcal{X}_1 + \mathcal{X}_2 + \mathcal{X}_3 = r_1 \Rightarrow \mathcal{X}_2 = 10 - \mathcal{X}_3$$

In this situation, the value of  $\mathcal{X}_2$  entirely determines the one of  $\mathcal{X}_3$  and vice versa.

$$H(\mathcal{X}_2 | X_3) = H(\mathcal{X}_2 | X_3) = 0 \Leftrightarrow H(\mathcal{X}_2) = H(\mathcal{X}_3) = I(\mathcal{X}_2; \mathcal{X}_3) \quad (12)$$

The possible pairs can easily be enumerated by hand :

$$(\mathcal{X}_2, \mathcal{X}_3) \in \{(1, 9), (2, 8), (3, 7), (7, 3), (8, 2), (9, 1)\}$$

Here as well, the a priori probability to realize one of the  $N = 6$  valid subgrids is uniform.

Once more, keeping in mind that only  $N$  subgrids have a non-null realisation probability, we compute the entropy  $H^4$  of this subgrid through its definition :

$$H(\text{subgrid}) = - \sum_{i=1}^N p \log_2 p = - \log_2 \frac{1}{N} = \log_2 N = 2.5850 \quad (13)$$

---

<sup>3</sup>Actually, the entropy we computed could/should be denoted as  $H(\text{subgrid} | \mathcal{X}_1 = 4)$  which shouldn't be mingled with  $H(\text{subgrid} | \mathcal{X}_1)$ .

<sup>4</sup>Once again, this could/should be denoted as  $H(\text{subgrid} | \mathcal{X}_1 = 4, r_1 = 14)$ . From now on, for the sake of conciseness, the computed entropies will always be those of *this* grid with *these* clues and constraints.

16. As seen above, adding or taking into account a row/column constraint reduces the number of valid (sub)grids. Therefore, it logically reduces the entropy of the whole (sub)grid. It also reduces the entropy of individual squares as their value is restrained to the ones that could produce a valid (sub)grid.

For example, in question 15, the domain of  $\mathcal{X}_2$  is restrained because there is some value, correct without constraints, of  $\mathcal{X}_2$  such that there is no value of  $\mathcal{X}_3$  producing a valid (sub)grid, e.g. 5 and 6.

The extreme case is when there is enough clues and constraints to actually solve deterministically the (sub)grid which means there is no uncertainty. As a consequence, the entropy of such (sub)grid is null.

17. Under assumption  $A$ , the entropy of a single square is only determined by  $N$  the number of valid values it could take which itself is influenced only by the clues and constraints.

For example,  $\mathcal{X}_2$  has to be different from  $\mathcal{X}_1 = 4$  and  $\mathcal{X}_9 = 1$  and smaller or equal than  $r_1 - \mathcal{X}_1$  and  $c_2$ .<sup>5</sup>

$$\begin{aligned}\mathcal{X}_2 &\in \{\forall x \in [1, 9] \mid x \notin (\mathcal{X}_1, \mathcal{X}_9), x \leq r_1 - \mathcal{X}_1, x \leq c_2\} \\ &\in \{2, 3, 5, 6, 7, 8, 9\}\end{aligned}$$

In this case  $N = 7$  and  $H(\mathcal{X}_2) = \log_2 N = 2.8074$ . For  $\mathcal{X}_3$ ,  $N = 6$  (9 is forbidden) and  $H(\mathcal{X}_3) = 2.5850$ . All other entropies are available in `fubuki.py`.

*Remark.* It should be noted that the entropy of squares with a clue is null. Indeed,  $H = \log_2 1 = 0$ .

18. The entropy of the unsolved grid is the joint entropy of all squares (without the ones with clues).

$$H(\text{grid}) = H(\mathcal{X}_2, \mathcal{X}_3, \dots, \mathcal{X}_8) \quad (14)$$

But, under assumption  $A$ , these random variables are considered independent. Therefore, knowing (1),

$$\begin{aligned}H(\text{grid}) &= \mathbb{E}_{\mathcal{X}_2, \dots, \mathcal{X}_8} (h(P(\mathcal{X}_2, \dots, \mathcal{X}_8))) \\ &= \mathbb{E}_{\mathcal{X}_2, \dots, \mathcal{X}_8} (h(P(\mathcal{X}_2)) \dots P(\mathcal{X}_8))) \\ &= \mathbb{E}_{\mathcal{X}_2, \dots, \mathcal{X}_8} (h(P(\mathcal{X}_2)) + \dots + h(P(\mathcal{X}_8))) \\ &= \mathbb{E}_{\mathcal{X}_2, \dots, \mathcal{X}_8} (h(P(\mathcal{X}_2))) + \dots + \mathbb{E}_{\mathcal{X}_2, \dots, \mathcal{X}_8} (h(P(\mathcal{X}_8))) \\ &= \mathbb{E}_{\mathcal{X}_2} (h(P(\mathcal{X}_2))) + \dots + \mathbb{E}_{\mathcal{X}_8} (h(P(\mathcal{X}_8))) \\ &= H(\mathcal{X}_2) + \dots + H(\mathcal{X}_8)\end{aligned} \quad (15)$$

But we already know these entropies (knowing  $\mathcal{X}_1 = 4$ ,  $\mathcal{X}_9 = 1$ ,  $r_1 = 14$ , ...). Thus,

$$H(\text{grid}) = 2.8074 + 2.5850 + \dots = 18.7619 \quad (16)$$

19. Under assumption  $A$ , for questions 14 and 15, the values of  $\mathcal{X}_2$  and  $\mathcal{X}_3$  are no longer linked. Automatically, the number of valid subgrids increases, so does the entropy.

---

<sup>5</sup>We could have considered  $r_1 - \mathcal{X}_1 - 1$  and  $c_2 - 2$  as the minimal value of squares is 1, but, for the sake of simplicity, we didn't.

Indeed, for question 14, instead of the arrangements of 2 within 8, we now have *twice* a draw within 8 possibilities, *i.e.*

$$N = 8^2 = 64$$

and  $H(\text{subgrid}) = \log_2 N = 6$ . This is also valid for question 15 as the constraint  $r_1 = 14$  doesn't restrain the domain of  $\mathcal{X}_2$  and/or  $\mathcal{X}_3$  under assumption *A*.

Under no assumption, questions 17 and 18 are much harder to answer. Indeed, it is now impossible to compute *separately* the valid values of each square as they influence each other. Therefore, the only way to know the valid values (hence the entropy) of each square is to compute *all* the valid possible grids, *i.e.* to solve the Fubuki.

Obviously, without assumption *A*, the number of possible grid has decreased dramatically, so does the entropy of the grid and squares. In fact, for *this* particular grid, there is only one solution (the one of the statement) meaning that, knowing the clues and constraints, one can deterministically find all squares. As a consequence, there is no uncertainty, *i.e.* the entropy of the grid (and squares) is null.

However, it doesn't mean that finding the solution(s) is trivial. Indeed, in a very general case, finding all the solutions comes back to enumerating all grids and selecting only the ones that are valid. Depending on the way the enumeration is done the complexity (with respect to the number of empty squares  $n$ ) may vary :

- a. Without taking rules into account : all combinations of  $n$  numbers taking values in a set of size  $n$  leads to  $n^n$  grids to visit;
- b. Taking rule (a) into account : all permutations of the  $n$  numbers of a set leads to  $n!$  grids to visit.

As comparison, under assumption *A*, computing the number of valid values (and therefore the entropy) of a square consisted in enumerating all  $n$  values for *that* square. Therefore, the computation of the entropy of the whole grid required the order of  $n^2$  operations.

20. As we have seen above, to solve a Fubuki grid, we have to explore the space (or a smartly chosen subset) of all grids in order to the solution(s). However, it doesn't mean that this (sub)space cannot be visited efficiently.

Let assume that some omnipotent entity (that feels like sharing its knowledge) knows the current probability distributions associated to the empty squares of the grid.

Lets draw a *guess* for one of the empty squares from its probability distribution. From now on, this guess acts as a clue and therefore modifies the number of possible values of all other squares, which we request to the hypothetical omnipotent entity.

Performing this guessing operation *recursively*, we either succeed to fill the whole grid, which means we found a solution, or a square hasn't possible values anymore, which means that at least the previous guess is wrong.

To avoid this situation as much as possible, each guess should carry the less risk, *i.e.* we should have the most *information* about the chosen square. Therefore, the square that should be filled first (and at each step) should be the one with the *lowest entropy* which we can compute with the probability distributions.

However, even with this policy, it is not possible to guarantee good guesses. Therefore, when reaching a dead end, the most recent guess is reverted and another guess is made. This method is called *backtracking* and will always find a solution if there is one.

*Remark.* A backtracking solver has been implemented in `fubuki.py` and is used to solve the given grid.

Since we don't have access to an omnipotent entity, the probability distribution is assumed uniform over the values that are valid in the sense of the assumption *A*.

21. The wisest choice would be to use the clue to reveal as much *information* about the grid as possible. To do so, we should choose the (empty) square  $\mathcal{Y}_1$  that maximizes its *mutual information* with the whole grid.

$$\begin{aligned}\mathcal{Y}_1 &= \arg \max_{\mathcal{X}_i} I(\text{grid}; \mathcal{X}_i) \\ &= \arg \max_{\mathcal{X}_i} H(\text{grid}) - H(\text{grid} \mid \mathcal{X}_i) \\ &= \arg \max_{\mathcal{X}_i} H(\text{grid}) - H(\text{grid}, \mathcal{X}_i) + H(\mathcal{X}_i)\end{aligned}$$

But  $\mathcal{X}_i$  being part of the grid,  $H(\text{grid}, \mathcal{X}_i) = H(\text{grid})$  and

$$\mathcal{Y}_1 = \arg \max_{\mathcal{X}_i} H(\mathcal{X}_i) \tag{17}$$

Therefore, we should choose to reveal the empty square with the *highest entropy* (which is kindly provided by our omnipotent entity), *i.e.* the one we are the least confident about.

22. Here as well the goal is to maximize the information revealed by the clues. Therefore, at each step/reveal  $i$ , we should select the square  $\mathcal{Y}_i$  that maximizes its *mutual information* with the whole grid *knowing all previous clues*.

$$\begin{aligned}\mathcal{Y}_i &= \arg \max_{\mathcal{X}_j} I(\text{grid}; \mathcal{X}_j \mid \mathcal{Y}_1 = y_1, \dots, \mathcal{Y}_{i-1} = y_{i-1}) \\ &= \arg \max_{\mathcal{X}_j} H(\text{grid} \mid \mathcal{Y}_1 = y_1, \dots, \mathcal{Y}_{i-1} = y_{i-1}) - H(\text{grid} \mid \mathcal{X}_j, \mathcal{Y}_1 = y_1, \dots, \mathcal{Y}_{i-1} = y_{i-1}) \\ &= \dots \\ &= \arg \max_{\mathcal{X}_j} H(\mathcal{X}_j \mid \mathcal{Y}_1 = y_1, \dots, \mathcal{Y}_{i-1} = y_{i-1})\end{aligned} \tag{18}$$

Therefore, the strategy is the same at each step and is, once more, to reveal the square with the *highest entropy* knowing all previous clues.

23. Here the problem is different. We have to choose a *subset* of  $k$  squares to reveal at once. It is therefore this subset  $(\mathcal{Y}_1, \dots, \mathcal{Y}_k)$  that should maximize its *mutual information* with the whole grid.

$$\begin{aligned}\mathcal{Y}_1, \dots, \mathcal{Y}_k &= \arg \max_{\mathcal{X}_i, \mathcal{X}_j, \dots} I(\text{grid}; (\mathcal{X}_i, \mathcal{X}_j, \dots)) \\ &= \arg \max_{\mathcal{X}_i, \mathcal{X}_j, \dots} H(\text{grid}) - H(\text{grid} \mid \mathcal{X}_i, \mathcal{X}_j, \dots) \\ &= \arg \max_{\mathcal{X}_i, \mathcal{X}_j, \dots} H(\text{grid}) - H(\text{grid}, \mathcal{X}_i, \mathcal{X}_j, \dots) + H(\mathcal{X}_i, \mathcal{X}_j, \dots) \\ &= \arg \max_{\mathcal{X}_i, \mathcal{X}_j, \dots} H(\mathcal{X}_i, \mathcal{X}_j, \dots)\end{aligned} \tag{19}$$

Thus, the strategy is to reveal the subset of empty squares with the *highest joint entropy*.

## Appendice - Details of the calculation by hand

## Exercise 1

We know that

$$P(X, Y) = P(X) \cdot P(Y)$$

This yields

		$y_j$			
	0	1	2	3	
$x_i$	0	$\frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$	$\frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$
	1	$\frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$	$\frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$
	2	$\frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$	$\frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$
	3	$\frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$	$\frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$	$\frac{1}{4} \cdot \frac{1}{8} = \frac{1}{32}$

We know that

$$P(S = S_i) = \sum_{X_j, Y_k | S_i = X_j + Y_k} P(X = X_j, Y = Y_k)$$

Hence,

$S_i$	$P(S = S_i)$
0	$\frac{1}{8}$
1	$(\frac{1}{16}) + (\frac{1}{18}) = \frac{3}{16}$
2	$(\frac{1}{32}) + (\frac{1}{16}) + (\frac{1}{18}) = \frac{7}{32}$
3	$2 \cdot (\frac{1}{32}) + (\frac{1}{16}) + (\frac{1}{18}) = \frac{8}{32} = \frac{1}{4}$
4	$2 \cdot (\frac{1}{32}) + (\frac{1}{16}) = \frac{4}{32} = \frac{1}{8}$
5	$2 \cdot (\frac{1}{32}) = \frac{1}{16}$
6	$\frac{1}{32}$

We know that

$$P(\tilde{Z} = Z_i) = \sum_{X_j, Y_k | Z_i = 1} P(X = X_j, Y = Y_k)$$

Hence,

$Z_i$	$P(\tilde{Z} = Z_i)$
0	$6(1/32) + 3(1/16) + 3(1/8) = 24/32 = 3/4$
1	$2(1/32) + (1/16) + (1/8) = 8/32 = 1/4$

Exercise 2

$$\cdot H(x) = - \sum_{x_i} P(x_i) \log_2 P(x_i)$$

$$= - \left[ \left( \frac{1}{4} \log_2 \frac{1}{4} \right) \cdot 4 \right]$$

$$= 2$$

$$\cdot H(y) = - \sum_{y_i} P(y_i) \log_2 P(y_i)$$

$$= - \left[ \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{4} \log_2 \frac{1}{4} + 2 \left( \frac{1}{8} \log_2 \frac{1}{8} \right) \right]$$

$$= 1,75$$

$$\cdot H(s) = - \sum_{s_i} P(s_i) \log_2 P(s_i)$$

$$= - \left[ \frac{1}{8} \cdot \log_2 \frac{1}{8} + \frac{3}{16} \log_2 \frac{3}{16} + \frac{4}{32} \log_2 \frac{4}{32} + \frac{1}{4} \log_2 \frac{1}{4} \right. \\ \left. + \frac{1}{8} \log_2 \frac{1}{8} + \frac{1}{16} \log_2 \frac{1}{16} + \frac{1}{32} \log_2 \frac{1}{32} \right]$$

$$= 2,5887$$

$$\cdot H(\tilde{z}) = - \sum_{z_i} P(z_i) \log_2 P(z_i)$$

$$= - \left[ \frac{3}{4} \log_2 \frac{3}{4} + \frac{1}{4} \log_2 \frac{1}{4} \right]$$

$$= 0,8113$$

## Calculations

For the rest of the developments, we will need:

- $P(X, S)$
- $P(Y, Z)$
- $P(S, Z)$
- $P(Z, X)$

Let's calculate them.

We know that

$$\bullet P(X, S) = \sum_Y P(X, Y, S)$$

$$\bullet P(Y, Z) = \sum_X P(X, Y, Z)$$

$$\bullet P(Z, X) = \sum_Y P(X, Y, Z)$$

so let's calculate  $P(X, Y, S)$  and  $P(X, Y, Z)$ .

$P(X, Y, S)$

$$P(X, Y, S) = P(X, Y) \cdot I(S = X + Y)$$

→ equals 0 when  $S \neq X + Y$  so we only calculate value when  $X + Y = S$

$S_k$	$X_i$	$Y_j$	$P(X_i, Y_j, S_k)$ when $S_k = X_i + Y_j$
0	0	0	1/8
1	0	1	1/16
	1	0	1/8
2	0	2	1/32
	2	0	1/8
	1	1	1/16
3	0	3	1/32
	3	0	1/8
	1	2	1/32
	2	1	1/16
4	1	3	1/32
	3	1	1/16
	2	2	1/32
5	2	3	1/32
	3	2	1/32
6	3	3	1/32

### P(x, y, z)

When  $Z_k = 1$ , we only have values of  $P(x, y)$  otherwise  $X_i = Y_j$  (the rest is equal to 0)

When  $Z_k = 0$ , we only have values of  $P(x, y)$  otherwise  $X_i \neq Y_j$  (the rest is equal to 0)

So, we have

$Z_k$	$X_i$	$Y_j$	$P(X_i, Y_j, Z_k)$
1	0	0	1/8
	1	1	1/16
	2	2	1/32
	3	3	1/32
0	0	1	1/16
	0	2	1/32
	0	3	1/32
	1	0	1/8
	1	2	1/32
	1	3	1/32
	2	0	1/8
	2	1	1/16
	2	3	1/32
	3	0	1/8
	3	1	1/16
	3	2	1/32

### P(x, s)

$$P(x, s) = \sum_y P(x, y, s)$$

It gives the same values as in table  $P(x, y, s)$

because for one couple  $(x_i, s_j)$  we only have one value

$y_k$

### P(y, z)

$$P(y, z) = \sum_x P(x, y, z)$$

		$y_i$		
$z_j$	0	1	2	
0	$\frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{3}{8}$	$\frac{1}{16} + \frac{1}{16} + \frac{1}{16} = \frac{3}{16}$	$\frac{1}{32} + \frac{1}{32} + \frac{1}{32} = \frac{3}{32}$	$\frac{1}{32} + \frac{1}{32} + \frac{1}{32} = \frac{3}{32}$
1	1/8	1/16	1/32	1/32

MEURISSE  
Maschine

P(Z, X)

$$P(Z, X) = \sum_Y P(X, Y, Z)$$

ROZET  
François

X<sub>i</sub>

0

1

2

3

$$0 \quad \frac{1}{16} + \frac{1}{32} + \frac{1}{32} = \frac{4}{32} \quad \frac{1}{8} + \frac{1}{32} + \frac{1}{32} = \frac{6}{32} \quad \frac{1}{8} + \frac{1}{16} + \frac{1}{32} = \frac{7}{32} \quad \frac{1}{8} + \frac{1}{16} + \frac{1}{32} = \frac{7}{32}$$

Z<sub>j</sub>

1 -1/8

-1/16

-1/32

-1/32

P(S, Z)

$$P(S, Z) = \sum_{X, Y} P(X, Y, S, Z)$$

To calculate these values, one thinks according to S and Z.

When S = S<sub>i</sub>, we take all the values in P(X, Y) such that X<sub>j</sub> + Y<sub>k</sub> = S<sub>i</sub>. Let's call this set of values W.

Then, when we have W, if Z<sub>i</sub> = 0, we only keep values in W such that X<sub>j</sub> ≠ Y<sub>k</sub> and we sum them. If Z<sub>i</sub> = 1, we do the same but with values such that X<sub>j</sub> = Y<sub>k</sub>.

If there is no value, the sum equals 0.

This gives us:

			S <sub>i</sub>					
	0	1	2	3	4	5		
0	0	0	3/16	5/32	8/32	3/32	2/32	0
Z <sub>j</sub>	1	-1/8	0	-1/16	0	-1/32	0	-1/32

## Exercice 3

$$\bullet H(x, y) = - \sum_{x_i, y_j} P(x_i, y_j) \log_2 P(x_i, y_j)$$

$$= - \left[ 4 \left( \frac{1}{8} \log_2 \frac{1}{8} \right) + 4 \left( \frac{1}{16} \log_2 \frac{1}{16} \right) + 8 \left( \frac{1}{32} \log_2 \frac{1}{32} \right) \right]$$

$$= 3,75$$

$$\bullet H(x, s) = - \sum_{x_i, s_j} P(x_i, s_j) \log_2 P(x_i, s_j)$$

$$= - \left[ \left( \frac{1}{8} \cdot \log_2 \frac{1}{8} \right) \cdot 4 + \left( \frac{1}{16} \cdot \log_2 \frac{1}{16} \right) \cdot 4 + \left( \frac{1}{32} \cdot \log_2 \frac{1}{32} \right) \cdot 8 \right]$$

$$= 3,75$$

$$\bullet H(y, z) = - \sum_{y_i, z_j} P(y_i, z_j) \cdot \log_2 P(y_i, z_j)$$

$$= - \left[ \frac{3}{8} \log_2 \frac{3}{8} + \frac{3}{16} \log_2 \frac{3}{16} + 2 \left( \frac{3}{32} \log_2 \frac{3}{32} \right) \right.$$

$$\left. + \frac{1}{8} \log_2 \frac{1}{8} + \frac{1}{16} \log_2 \frac{1}{16} + 2 \left( \frac{1}{32} \log_2 \frac{1}{32} \right) \right]$$

$$= 2,5613$$

$$\bullet H(s, z) = - \sum_{s_i, z_j} P(s_i, z_j) \log_2 P(s_i, z_j)$$

$$= - \left[ \frac{3}{16} \log_2 \frac{3}{16} + \frac{5}{32} \log_2 \frac{5}{32} + \frac{8}{32} \log_2 \frac{8}{32} + \frac{3}{32} \log_2 \frac{3}{32} \right.$$

$$\left. + \frac{2}{32} \log_2 \frac{2}{32} + \frac{1}{8} \log_2 \frac{1}{8} + \frac{1}{16} \log_2 \frac{1}{16} + 2 \left( \frac{1}{32} \log_2 \frac{1}{32} \right) \right]$$

$$= 2,8789$$

MEURISSE  
Maschine

#### Exercise 4

$$\begin{aligned} \bullet H(x|y) &= H(x, y) - H(y) \\ &= 3,75 - 1,75 \\ &= 2 \end{aligned}$$

$$\bullet H(z|x) = \underbrace{H(z, x)}_{\text{We need } H(z, x)} - H(x)$$

$$\text{We need } H(z, x) = - \sum_{z_i, x_j} p(z_i, x_j) \underbrace{\log_2 p(z_i, x_j)}_{\text{We need } p(z_i, x_j)!}$$

$\rightarrow$  already calculated!

and so,

$$\begin{aligned} H(z, x) &= - \left[ 2 \left( \frac{4}{32} \log_2 \frac{4}{32} \right) + 2 \left( \frac{1}{32} \log_2 \frac{1}{32} \right) + \frac{1}{8} \log_2 \frac{1}{8} \right. \\ &\quad \left. + \frac{1}{16} \log_2 \frac{1}{16} + \frac{4}{32} \log_2 \frac{4}{32} + \frac{6}{32} \log_2 \frac{6}{32} \right] = 2,7246 \\ \text{finally.} \end{aligned}$$

$$H(z|x) = 0,7246 = 2,7246 - 2$$

$$\begin{aligned} \bullet H(s|x) &= H(s, x) - H(x) \\ &= H(x, s) - H(x) \\ &= 3,75 - 2 \\ &= 1,75 \end{aligned}$$

$$\begin{aligned} \bullet H(s|z) &= H(s, z) - H(z) \\ &= 2,8789 - 0,8113 \\ &= 2,0676 \end{aligned}$$

### Exercise 5

- $H(X, Y | S) = H(X, Y, S) - H(S)$
- $H(S, Y | X) = \underbrace{H(S, Y, X)}_{\text{We need } H(X, Y, S)!} - H(X)$

We need  $H(X, Y, S)!$

$$H(X, Y, S) = - \sum_{X_i, Y_j, S_k} P(X_i, Y_j, S_k) \cdot \log_2 P(X_i, Y_j, S_k)$$

We need  $P(X_i, Y_j, S_k)!$

$$P(X, Y, S) = P(X, Y) \cdot \mathbb{1}(S = X + Y)$$

→ equals 0 when  $S \neq X + Y$  so we only compute value when  $X + Y = S$ .

$S_k$	$X_i$	$Y_j$	$P(X_i, Y_j, S_k)$ when $S_k = X_i + Y_j$
0	0	0	$1/8$
1	0	1	$1/16$
	1	0	$1/8$
2	0	2	$1/32$
	2	0	$1/8$
	1	1	$1/16$
3	0	3	$1/32$
	3	0	$1/8$
	1	2	$1/32$
	2	1	$1/16$
4	1	3	$1/32$
	3	1	$1/16$
	2	2	$1/32$
5	2	3	$1/32$
	3	2	$1/32$
6	3	3	$1/32$

and so,

$$H(X, Y, S) = - \left[ 4 \left( \frac{1}{8} \log_2 \frac{1}{8} \right) + 4 \left( \frac{1}{16} \log_2 \frac{1}{16} \right) + 8 \left( \frac{1}{32} \log_2 \frac{1}{32} \right) \right]$$

$$= 3,75$$

finally,

$$H(X, Y | S) = 3,75 - 2,5887 = 1,1613$$

$$H(S, Y | X) = 3,75 - 2 = 1,75$$

## Exercise 6

$$\begin{aligned} \bullet I(x; y) &= H(x) - H(x|y) \\ &= 2 - 2 \\ &= 0 \end{aligned}$$

$$\begin{aligned} \bullet I(x; s) &= H(s) - H(s|x) \\ &= 2,5887 - 1,75 \\ &= 0,8387 \end{aligned}$$

$$\begin{aligned} \bullet I(y; z) &= H(y) + H(z) - H(y, z) \\ &= 1,75 + 0,8113 - 2,5613 \\ &= 0 \end{aligned}$$

$$\begin{aligned} \bullet I(s; z) &= H(s) - H(s|z) \\ &= 2,5887 - 2,0676 \\ &= 0,5211 \end{aligned}$$

## Exercise 7

$$\begin{aligned} \bullet I(x; y|s) &= H(x|s) - H(x|y, s) \\ &= [H(x, s) - H(s)] - [H(x, y, s) - H(y, s)]^* \\ &= (3,75 - 2,5887) - (3,75 - 3,75) \\ &= 1,1613 \end{aligned}$$

$$\begin{aligned} \bullet I(s; y|x) &= H(s|x) - H(s|y, x) \\ &= H(s|x) - [H(s, y, x) - H(y, x)] \\ &= 1,75 - (3,75 - 3,75) \\ &= 1,75 \end{aligned}$$

\*  $H(y, s)$  was calculated in the same way as  $H(x, s)$  in y (the value of  $P(x)$  has been replace by those of  $P(y)$ ).