TD 1 – Shannon Theory

The following exercises are meant to illustrate chapter 1 of Gabriel Peyré's ***Mathematics of Data*** *course (`mathematical-tours.github.io`). Some of them are adapted from the past years' exams.*

Please refer to Geert-Jan Huizing (`huizing@ens.fr`) for questions regarding these exercises.

Exercises

We recall the definition of the entropy (in bits): $H(p) \stackrel{\text{def.}}{=} -\sum_k p_k \log_2 p_k$ with the convention $0\log(0) = 0$.
This definition extends to matrices by replacing k with (i, j).
For $(a, b) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$, we define $a \otimes b \stackrel{\text{def.}}{=} (a_i b_j)_{ij}$. Finally, we introduce the simplex $\Sigma_n \stackrel{\text{def.}}{=} \left\{p \in \mathbb{R}_+^n : \sum_k p_k = 1\right\}$.

[$\star$] Entropic coding (adapted from exam 2021)

We consider the alphabet $(s_1, s_2, s_3, s_4, s_5)$. The probabilities of appearance of the symbols s_k are $p_1 = 1/3$, $p_2 = 1/4$, $p_3 = 1/6$, $p_4 = 1/6$, $p_5 = 1/12$. You may use $\log_2(3) \approx 1.6$.

1. Compute the entropy $H(p)$ for the distribution of the considered alphabet.
2. If one were to define a fixed length code for this alphabet, how many bits would be needed to code each symbol?
3. What is the optimal average number of bits per symbol for a code on this alphabet? Why is a fixed length code inefficient?
4. Draw a binary prefix coding tree, following the proof of Shannon's theorem. What is the average number of bits per symbol for the associated code?
5. Draw the Huffmann tree for the considered alphabet, explaining each step. Write the associated code. What is the average number of bits per symbol for the associated code?

[$\star$] Entropic coding by blocks (adapted from exam 2020)

We assume X is a discrete random variable with values in $\{1, \ldots, k\}$ with probability distribution $p = (p_1, \ldots, p_k)$.

1. What is the probability distribution $q = (q_{i_1,\ldots,i_n})_{i_1,\ldots,i_n}$ of the random vector $(X_1, \ldots, X_n)$ on $\{1, \ldots, k\}^n$, where the X_i are independant copies of X?
2. Compute the entropy $H(q)$ of q as a function of $H(p)$.
3. We assume an infinite sequence of symbols with distribution p. Show that by using a Huffman code on blocks n consecutive symbols, the average number of bits per symbol tends to $H(p)$ as $n \to \infty$.

[$\star\star$] **Entropy function**

1. Show that $H : p \in \mathbb{R}_+^n \to -\sum_k p_k \log_2 p_k$ is a strictly concave function.
2. For what value of $p \in \mathbb{R}_+^n$ is H maximal?
3. Show that for all $p, q \in \Sigma_n$, $H(p) \leq -\sum_i p_i \log_2(q_i)$. Then, show that $H(p) \leq \log_2(n)$. Finally, find for which $p \in \Sigma_n$ the function H is maximal.
4. For $(a, b) \in \Sigma_n \times \Sigma_m$, compute $H(a \otimes b)$.

[$\star\star$] **Kullback-Leibler divergence (adapted from exam 2017)**

For $q \in \mathbb{R}_{+,*}^n$ (strictly positive) and $r \in \mathbb{R}_+^n$, we define the Kulback-Leibler divergence between the two vectors as

$$\mathrm{KL}(r|q) \stackrel{\text{def.}}{=} \sum_i r_i \log(\frac{r_i}{q_i}) - r_i + q_i.$$

The same expression holds also for matrices, where the sum is on (i, j) instead of just i.

1. Show that the function $\mathrm{KL}(\cdot|q)$ is strictly convex and compute its minimizer.
2. Deduce that KL is "distance-like", i.e. that $\mathrm{KL}(r|q) > 0$ and $\mathrm{KL}(r|q) = 0$ if and only if $r = q$.
3. Show that, if $(a, b) \in \Sigma_n \times \Sigma_m$, $(a', b') \in \mathbb{R}_{+,*}^n \times \mathbb{R}_{+,*}^m$ and $P \in \mathbb{R}_+^{n \times m}$ such that $P\mathbb{1}_m = a$ and $P^\top \mathbb{1}_n = b$, then one has

$$\mathrm{KL}(P|a \otimes b) + \mathrm{KL}(a \otimes b|a' \otimes b') = \mathrm{KL}(P|a' \otimes b').$$

Solutions

Entropic coding

1. By definition,

   $H(p) = -(1/3)\log_2(1/3) - (1/4)\log_2(1/4) - (1/6)\log_2(1/6) - (1/6)\log_2(1/6) - (1/12)\log_2(1/12)$
   $H(p) \approx 2.2.$

2. There are 5 symbols, and $\lceil \log_2 5 \rceil = 3$ so 3 bits are necessary.

3. The optimal average length per symbol is $H(p) = 2.2 < 3$. The fixed-length code is inefficient because the most used symbols are not shorter than the least used symbols.

4. First, we determine the lengths of code-words using $l_k = \lceil -\log_2(p_k) \rceil$. We have $l_1 = 2$, $l_2 = 2$, $l_3 = 3$, $l_4 = 3$, $l_5 = 4$. An associated prefix code is thus $c_1 = 00$, $c_2 = 01$, $c_3 = 100$, $c_4 = 101$, $c_5 = 1100$. The average number of bits per symbol is 2.5, which is greater than the entropy lower bound.

5. At each step we sort the symbols by probability and merge 2 symbols with lowest probability.

```
p_1 >= p_2 >= p_3 >= p_4 >= p_5
                                      /\
                                   s_4  s_5

p_1 >= p_2 >= p_{4, 5} >= p_3
                                        /\
                                       /  s_3
                                      /\
                                   s_4  s_5

p_{3,4,5} >= p_1 >= p_2

                                       /\         /\
                                      /  s_3   s_1  s_2
                                     /\
                                  s_4  s_5


p_{1,2} >= p_{3,4,5}
                                            /\
                                           /  \
                                          /    \
                                         /      \
                                        /        \
                                       /\         /\
                                      /  s_3   s_1  s_2
                                     /\
                                  s_4  s_5
```

The average number of bits per symbol is 2.25, which is greater than the entropy lower bound.

Entropic coding by blocks

1. $q_{i_1,\dots,i_n} = p_{i_1} \times \dots \times p_{i_n}$
2. $$\begin{aligned} H(q) &= -\sum_{i_1,\dots,i_n} q_{i_1,\dots,i_n} \log_2(q_{i_1,\dots,i_n}) \\ H(q) &= -\sum_{i_1,\dots,i_n} p_{i_1} \times \dots \times p_{i_n} \sum_k \log_2(p_{i_k}) \\ H(q) &= -\sum_k \sum_i p_{i_1} \times \dots \times p_{i_n} \log_2(p_{i_k}) \\ H(q) &= nH(p) \end{aligned}$$
3. The average number of bit per group of symbol, using Shannon bound, is thus smaller than $nH(p)+1$ and the average number of bit per symbol is thus smaller than $H(p)+1/n$.

Entropy function

1. For $p > 0$, $h'(p) = -(1+\log p)/\log 2$, and $h''(p) = -1/(p \times \log 2) < 0$ so H is strictly concave.
2. $h'(p) = 0 \iff \log p = -1 \iff p = 1/e$. So $\arg\max_p H(p) = \mathbb{1}_n/e$.
3. Since $\log(u) \leq u-1$, $H(p) + \sum_i p_i \log_2(q_i) = \sum_i p_i \log_2(q_i/p_i) \leq \sum_i p_i(q_i/p_i - 1) = 0$. Applying this inequality to $q = \frac{1}{n}\mathbb{1}_n$ gives the expected result, which is reached for $p = \frac{1}{n}\mathbb{1}_n$.
4. $H(a \otimes b) = -\sum_{ij} a_i b_j \log_2 a_i b_j = -\sum_{ij} a_i b_j \log_2 a_i - \sum_{ij} a_i b_j \log_2 b_j$. Since a and b sum to 1, this simplifies to $H(a \otimes b) = H(a) + H(b)$.

Kullback-Leibler divergence

1. The second order derivate is positive, and the minimizer is $r = q$.
2. The function is strictly convex and its minimum is 0, only reached if $r = q$.
3. We have

$$\begin{aligned} \mathrm{KL}(P|a \otimes b) + \mathrm{KL}(a \otimes b|a' \otimes b') &= \sum_{i,j} \left(P_{i,j} \log \frac{P_{i,j}}{a_i b_j} + a_i b_j \log \frac{a_i b_j}{a'_i b'_j} - P_{i,j} + a'_i b'_j \right) \\ &= \sum_{i,j} \left(P_{i,j} \log(P_{i,j}) - P_{i,j} \log(a_i b_j) + a_i b_j \log \frac{a_i b_j}{a'_i b'_j} + a'_i b'_j - P_{i,j} \right) \\ &= \sum_{i,j} \left(P_{i,j} \log(P_{i,j}) + a'_i b'_j - P_{i,j} \right) - \sum_i a_i \log a'_i - \sum_j b_j \log b'_j \\ &= \sum_{i,j} \left(P_{i,j} \log(P_{i,j}) + a'_i b'_j - P_{i,j} \right) - \sum_{i,j} P_{i,j} \log a'_i b'_j \\ &= \sum_{i,j} \left(P_{i,j} \log \frac{P_{i,j}}{a'_i b'_j} + a'_i b'_j - P_{i,j} \right) \\ &= \mathrm{KL}(P|a' \otimes b') \end{aligned}$$