Introduction to Information Theory, Fall 2020

Practice problems for exercise class #5

You do **not** have to hand in these exercises, they are for your practice only.

1. **Entropy, essential bit content:** Let X be a random variable with probability distribution P with five possible outcomes A, B, C, D and E and probabilities $P(A) = 1/2$, $P(B) = 1/8$, $P(C) = 1/4$, $P(D) = 1/16$, $P(E) = 1/16$.

   (a) What is the entropy $H(X)$?
   (b) Sketch $H_\delta(X)$ as a function of δ.

   Now let X_i for $i = 1, 2$ be IID random variables, taking values on an alphabet $\{a, b, c, d\}$ with probabilities $P(a) = 1/2$, $P(b) = 1/4$, $P(c) = 1/8$, $P(d) = 1/8$. That means in particular that we have the following table of probabilities and outcomes

x^2	$P(x^2)$
aa	$1/4$
ab, ba	$1/8$
bb, ac, ca, ad, da	$1/16$
bc, cb, bd, db	$1/32$
cc, cd, dc, dd	$1/64$

   (c) Compute $H(X^2)$.
   (d) Let $\varepsilon = 0.6$. Compute the typical set $T_{2,\varepsilon}$
   (e) Give a set S which is such that $\#S < \#T_{2,\varepsilon}$ and and $P(X^2 \in S) \geqslant P(X^2 \in T_{2,\varepsilon})$.

2. **Symbol source codes versus Shannon source codes:** The source coding theorem proved in Lecture 4 can be informally summarized by saying that when we use an optimal prefix code for a *block* of symbols then we can compress at rate arbitrarily close to $H(P)$. Similarly, Shannon's source coding theorem discussed in Lecture 5 informally states that when we use a block coding we can compress at rate arbitrarily close to $H(P)$ for any fixed error probability. How are the two results related? How do they differ?

3. **Reprise I: Properties of typical sets** These properties have also been discussed in the lecture, but it can be very helpful to think about them yourself again! Let $X_1, \ldots, X_N$ be IID random variables with a probability distribution P on a set of outcomes $\mathcal{A}$, and let $X^N = (X_1, \ldots, X_N)$, which has $\mathcal{A}^N$ as set of outcomes. Recall the definition of a *typical set* from the lecture:

   $$T_{N,\varepsilon} = \left\{x^N \in \mathcal{A}^N : |\frac{1}{N} \log \frac{1}{P(x^N)} - H(P)| \leqslant \varepsilon\right\}$$

   Prove the following properties:

   (a) The probability of $x^N \in T_{N,\varepsilon}$ is bounded by

   $$2^{-N(H(P)+\varepsilon)} \leqslant P(x^N) \leqslant 2^{-N(H(P)-\varepsilon)}.$$

   (b) The number of elements in $T_{n,\varepsilon}$ is bounded by

   $$\#T_{N,\varepsilon} \leqslant 2^{N(H(P)+\varepsilon)}.$$

   Hint: Use (a)!

(c) The probability of *not* being in the typical set goes to zero as N goes to infinity, that is

$$\Pr(X^N \notin T_{N,\varepsilon}) \leqslant \frac{\sigma^2}{N\varepsilon^2} \xrightarrow[N\to\infty]{} 0$$

where $\sigma^2 = \mathrm{Var}(\log \frac{1}{P(X_k)})$ for any k (recall that the X_k are all identically distributed).

(d) *Bonus property:* For any $\delta > 0$ and sufficiently large N, the number of elements in $T_{N,\varepsilon}$ is bounded from below by

$$\#T_{N,\varepsilon} \geqslant (1-\delta)2^{N(H(P)-\varepsilon)}.$$

Hint: Use 1(a) and 1(c)!

4. **Reprise II: Shannon's source coding theorem** Recall from the lecture that Shannon's source coding theorem states that for $0 < \delta < 1$ and $X_1, X_2, \ldots$ IID random variables with distribution P,

$$\lim_{N\to\infty} \frac{H_\delta(X^N)}{N} = H(P).$$

In this exercise you will be guided through the proof of this theorem. It has already been discussed in the lecture, but again, it will be much easier to understand if you try to work your way through the proof yourself!

(a) First use 1(c) to show that for all $\varepsilon > 0$ and for sufficiently large N,

$$\frac{H_\delta(X^N)}{N} \leqslant H(P) + \varepsilon. \tag{1}$$

We now want to show that for all $\varepsilon > 0$ and for sufficiently large N

$$\frac{H_\delta(X^N)}{N} \geqslant H(P) - \varepsilon. \tag{2}$$

We will do so in multiple steps:

(b) Explain that to prove Eq. (2), you can just as well show that for any fixed $\varepsilon > 0$ there is *no* infinite sequence $N_1 < N_2 < \ldots$ such that

$$\frac{H_\delta(X^{N_i})}{N_i} < H(P) - \varepsilon. \tag{3}$$

We will now show this claim by 'proof by contradiction'. This means that we will show that if we had such a sequence this would imply a contradiction (thus, no such sequence can possible exist in the first place!).

(c) Suppose that there exist an infinite sequence $N_1 < N_2 < \ldots$ such that Eq. (3) holds. Show that in this case there are sets $S_{N_i} \subseteq \mathcal{A}^{N_i}$ such that

$$P(X^{N_i} \in S_{N_i}) \geqslant 1 - \delta \quad \text{and} \quad \#S_{N_i} \leqslant 2^{N_i(H(P)-\varepsilon)}.$$

(d) Show that

$$\Pr(X^{N_i} \in S_{N_i} \cap T_{N_i,\frac{\varepsilon}{2}}) + \Pr(X^{N_i} \notin T_{N_i,\frac{\varepsilon}{2}}) \geqslant 1 - \delta.$$

On the other hand, show by combining 1(c), 1(a) and the properties of S_{N_i} that

$$\Pr(X^{N_i} \in S_{N_i} \cap T_{N_i,\frac{\varepsilon}{2}}) + \Pr(X^{N_i} \notin T_{N_i,\frac{\varepsilon}{2}}) \xrightarrow[i\to\infty]{} 0.$$

(e) Observe that the bounds in Eqs. (1) and (2) together prove Shannon's theorem, i.e.,

$$\lim_{N\to\infty} \frac{H_\delta(X^N)}{N} = H(P).$$

Hint: If necessary, look up the definition of the limit of a real-valued sequence in the lecture notes (or on Wikipedia).

5. **Enumeration of binary sequences:** In the Thursday lecture we will discuss a universal compression scheme. For this week's homework you will have to implement this scheme, and to help you we will work out an algorithm for the compressor and the decoder in this exercise. Let $\mathcal{A}^N$ be the set of all bitstrings of zeros and ones of length N and let $B(N,k) \subset \mathcal{A}^N$ be set of all strings x^N of length N with k ones. We will then order these sets in an appropriate way, and given x^N we compress by sending over k, the number of ones in x^N, and its index in $B(N,k)$. For the decoder, we just read out the appropriate element from $B(N,k)$. In this exercise we will derive a recursive algorithm for enumerating strings in $B(N,k)$ (notice that these sets will be exponentially large in N so we should not just enumerate over them!). We will use the lexicographic order (denoted $\leqslant_{\text{lex}}$), formally defined as follows: Given bitstrings x and y, we have that $x \leqslant_{\text{lex}} y$ if either $x = y$ or $x_i < y_i$ for the smallest i such that $x_i \neq y_i$. For example, $\mathtt{001} \leqslant_{\text{lex}} \mathtt{010} \leqslant_{\text{lex}} \mathtt{110}$.

(a) To get some intuition, write down $B(4,2)$ in lexicographically increasing order.

(b) Argue that

$$B(N,k) = \begin{cases} \{\mathtt{0...0}\} & \text{if } k = 0, \\ \{\mathtt{1...1}\} & \text{if } k = N, \\ \{\mathtt{0}x \,|\, x \in B(N-1,k)\} \cup \{\mathtt{1}x \,|\, x \in B(N-1,k-1)\} & \text{otherwise.} \end{cases}$$

(c) We want to find an algorithm that assigns to a bitstring in $B(N,k)$ its index in the lexicographical order on $B(N,k)$. Argue that Algorithm 1 gives the right result (notice that we start counting at 0, and we use the convention that $\binom{N}{k} = 0$ if $k > N$).

(d) For the decoding, we need an algorithm that finds the bitstring from k and its index in $B(N,k)$. Argue that Algorithm 2 gives the right answer.

Algorithm 1 Calculate index of a bitstring x

```
procedure INDEX($x$)
    $N \leftarrow$ LENGTH($x$)
    $k \leftarrow$ NUMBER_OF_ONES($x$)
    if $N = 0$ then
        return $0$
    end if
    if $x[0] = \mathtt{0}$ then
        return INDEX($x[1...]$)
    else
        return $\binom{N-1}{k}$+INDEX($x[1...]$)
    end if
end procedure
```

Algorithm 2 Calculate string from length N, number of ones k, and index m

```
procedure STRING($N, k, m$)
    if $N = 0$ then
        return Empty string
    end if
    if $m < \binom{N-1}{k}$ then
        return APPEND($\emptyset$, STRING($N - 1, k, m$))
    else
        return APPEND($1$, STRING($N - 1, k - 1, m - \binom{n-1}{k}$)))
    end if
end procedure
```