Introduction to Information Theory, Fall 2020

Practice problems for exercise class #6

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

0. **Exercises from MacKay:** 6.4, 6.5, 6.6

1. **Lossless universal block codes:** Recall the universal block coding algorithm discussed in the lecture which compresses sequences x^N with values in $\{\mathtt{0}, \mathtt{1}\}$ (i.e. bitstrings). This worked by considering k, the number of occurences of 1 in x^n, and the index p of x^N in $B(N, k)$, the set of all bitstrings of length N with k ones and $N - k$ zeros. The compression of x^N is given by expressing (p, k) in binary. As an example, we consider the bitstring $x^N = \mathtt{0001000}$.

   (a) What are k and p? What order on $B(N, k)$ are you using?
   (b) What is the compression of x^N? How many bits do you need to use for k and p? *Hint: be careful about how many bits you use for* k.

2. **Lempel-Ziv for a constant sequence:** Consider a source X which is 1 with probability 1.

   (a) What is the entropy $H(X)$?
   (b) How does Lempel-Ziv encode the sequence $x_N = 1111\ldots111$ of N times the symbol 1?
   (c) If M_N is the number of bits needed in the Lempel-Ziv encoding of x_N, argue that the rate M_N/N goes to zero.

3. **Average compression rate of Lempel-Ziv (mathematics challenge):** In class, we claimed that the average rate when compressing an IID source using the LZ algorithm is close to the entropy for large N. The proof is described in the lecture notes and is somewhat tricky. Have a look and try to understand it!

4. **Worst case analysis of Lempel-Ziv compression (mathematics challenge):** This exercise is a real challenge, and it is optional. In the lecture we showed that the LZ algorithm performs well on average. That means that it necessarily makes some messages longer, but in this problem you will show that it will not make them too much longer (which is of course a nice property for a compression algorithm). To be precise, you will show that the worst case rate is $R \leqslant 1 + \mathcal{O}(\frac{1}{\log(N)})$. For simplicity we will assume that our set of symbols has only two elements.

   (a) Consider the string x_λ which is constructed by enumerating all phrases up to length λ, ordered from short to long, and concatenating them. For instance, if we enumerate all phrases up to length 2 we have $\{A, B, AA, AB, BA, BB\}$ and x_2 would be $ABAAABBABB$. Argue that x_λ gives

$$c_\lambda = \sum_{k=1}^{\lambda} 2^k = 2^{\lambda+1} - 2$$

phrases in the LZ encoding (hint: geometric series) and that x_λ has length

$$N_\lambda = \sum_{k=1}^{\lambda} k2^k = (\lambda-1)2^{\lambda+1} + 2$$

(hint: induction for the second equality) and hence

$$c_\lambda \leqslant \frac{N_\lambda}{\lambda - 1}$$

for $\lambda \geqslant 1$.

(b) Argue that c_λ is the worst case number of phrases for strings of length N_λ in the LZ encoding.

(c) For the rest of the exercise, we will denote by N the string length, and by c (as a function of N) the worst case number of phrases for a message of length N in the LZ encoding. Argue that if $N_\lambda \leqslant N < N_{\lambda+1}$, then c satisfies

$$c \leqslant c_\lambda + \frac{N - N_\lambda}{\lambda + 1}.$$

(d) Using (a) and (c), show that c satisfies

$$c \leqslant \frac{N}{\lambda - 1}$$

for $\lambda > 1$ where λ satisfies

$$\lambda \geqslant \log(c) - 2.$$

(e) Deduce from (d) that

$$c \leqslant \frac{N}{\log(c) - 3}$$

and show that this implies that

$$c \leqslant \mathcal{O}(\frac{N}{\log(N)})$$

(hint: look at a similar argument in the lecture).

(f) Recall from the lecture that the length of the encoding l is bounded by

$$l \leqslant c\log(c) + 2c$$

and use this to show that

$$l \leqslant N + 5c \leqslant N + \mathcal{O}(\frac{N}{\log(N)}).$$

Conclude that the rate $R = \frac{l}{N}$ is bounded by

$$R \leqslant 1 + \mathcal{O}(\frac{1}{\log(N)}).$$