Board questions set 7

Problem 1: Chain rule for entropy

Prove the chain rule for entropy, namely that $H(X,Y) = H(X|Y) + H(Y)$.

Data compression For the rest of today, we are studying the problem of *data compression*. Assume we have a source of information which emits four different symbols a, b, c, d with probabilities $1/2, 1/4, 1/8, 1/8$, respectively. We model our source as iid realisation of a categorical random variable X with distribution P_X. A typical sequence of symbols from this source could look like this: $babacdbbaabadbaaaa$. Our task is to *compress* such sequences as much as possible. Formally, we would like to map every source symbol to a binary string such that (i) we can recover the original source symbol again and (ii) the average encoding length is minimal.

Problem 2: Codes

The following are four (binary symbol) codes C, D, E, F for the categorical random variable X, with $\mathcal{X} = \{a, b, c, d\}$:

x	$P(X = x)$	$C(x)$	$D(x)$	$E(x)$	$F(x)$
a	$1/2$	0	0	0	00
b	$1/4$	10	010	01	01
c	$1/8$	110	01	011	10
d	$1/8$	111	10	111	11

These codes can be used to encode strings of symbols by concatenation. For instance, the encoding of string "adba" under code E is

$$E(adba) = E(a)E(d)E(b)E(a) = 0\ 111\ 01\ 0 = 0111010$$

(a) What is the encoding of $adba$ under codes D and F?

(b) What is the decoding of 001001110 under code C?

(c) What is the decoding of 0100100 under code D? Is it unique?

(d) What is the decoding of 001111 under code E? Is it unique? What happens if you learn that the next bit is 1 (so you have to decode 0011111 under E)?

(e) Can you prove that arbitrary concatenations of codewords of C are uniquely decodable? What about concatenations of codewords of E or F?

(f) Which of the above codes is the most convenient to work with in terms of encoding and decoding? Why?

Problem 3: Code Length

The *average code length* of a binary symbol code is defined as follows. Let $\ell(s)$ denote the length of a string $s \in \{0,1\}^*$. The (average) length of a code C for a source X is defined as

$$\ell_C(X) := \mathbb{E}[\ell(C(X))] = \sum_{x \in \mathrm{supp}(X)} P(X = x)\ell(C(x))\,.$$

(a) Compute $\ell_C(X)$, $\ell_D(X)$, $\ell_E(X), \ell_F(X)$ for the codes of the previous section.

(b) Compute the entropy $H(X)$ for the distribution P_X above. Compare the obtained values $H(X)$ and $\ell_C(X)$ and the way you have computed them.

Problem 4: Optimal Codes

In the Information Theory course, we will prove Shannon's source-coding theorem: If P_X is a distribution and $\ell_{\min}(X) := \min_C \ell_C(X)$ the minimal average codeword length among all uniquely decodable codes, then

$$H(X) \leq \ell_{\min}(X) \leq H(X) + 1\,.$$

In other words, the Shannon entropy pretty much determines the optimal average codeword length.

(a) Show that code C from question 2 is optimal in terms of average coding length.

(b) Construct an optimal symbol code for the following distribution:

y	a	b	c	d	e	f	g
$P(Y = y)$	$1/4$	$1/4$	$1/8$	$1/8$	$1/8$	$1/16$	$1/16$

Hint: should symbols with high probability to occur receive long or short codewords?

(c) Prove that the code you found is optimal!

(d) Look up on the internet what Huffman coding is and use it to find an optimal binary symbol code for the following distribution:

z	a	b	c	d	e
$P(Z = z)$	0.25	0.25	0.2	0.15	0.15

Problem 5: Randomness-Efficient Sampling

Let's consider a different problem, namely how to efficiently sample iid from a distribution P_X. Explain how to repeatedly sample from P_X given an optimal binary code and access to uniformly distributed random bits. How many random bits per sampled symbol do you need on average?