B8.4 Information Theory

Sheet 3 — MT23

Section A

1. Let $|\mathcal{X}| = 100$ and p the uniform distribution on $\mathcal{X}$. How many codewords are there of length $l = 1, 2, \cdots$ in a Huffman binary code?

   Solution: By the Huffman procedure, we can see that there are 28 codewords of length 6 and 72 of length 7.

   Another way to get these numbers is as follows:

   Consider the optimization of l_x for optimal code

   $$\min \sum_{i=1}^{100} p_i l_i \quad \text{subject to } \sum_{i=1}^{100} 2^{-l_i}$$

   The optimal l_i should be integers close to $-\log(p_i)$, i.e. 6 or 7 in this question.

   To prove this, write $\Gamma = \{u = (u_1, \cdots u_{100}) : \sum 2^{-u_i} \leq 1\}$ for the set of feasible solutions (without integer constraint), and $J(u) = \sum u_i$ for the objective function.

   Defining $u^* = \log(100) * (1, 1, 1, \cdots 1)$, $A = \{6, 7\}^{100} \cap \Gamma$, and $\bar{A}$ be the convex hull of A, which is contained in Γ.

   Then for any feasible solution in $\bar{A}$, the segment between u and u^* must intersect with $\bar{A}$, hence intersect with the surface of $\bar{A}$. So, there exists a $\lambda \in (0, 1)$ such that $u^\lambda = \lambda u + (1 - \lambda)u^*$ is on the surface of $\bar{A}$, and $J(u^\lambda) = \lambda J(u) + (1 - \lambda)J(u^*)$. Since $J(u^*) < J(u^*)$, so $J(u^\lambda) < J(u)$. Furthermore, $u^*\lambda$ is on the surface of $\bar{A}$, so there exists a $\hat{u} \in A$ such that $J(\hat{u}) \leq J(u^\lambda)$, which implies u cannot be optimal.

2. Consider an alphabet $\mathcal{X} = \{A, B, C\}$, with probabilities $p(A) = 0.3, p(B) = 0.5, p(C) = 0.2$. Consider building an arithmetic code with these probabilities (in this order). Compute the interval associated with the input string 'ABBA' and the first 5 digits of the string associated with the number $1/\pi$.

   Solution: To encode the string ABBA, we have the sequence of intervals (written as

Sam Cohen: cohens@maths.ox.ac.uk

decimals):

$$\begin{aligned}
A &\mapsto [0, 0.3) \\
AB &\mapsto [0.3 \times (0.3 - 0), 0.8 \times (0.3 - 0)) = [0.09, 0.24) \\
ABB &\mapsto [0.09 + 0.3 \times (0.24 - 0.09), 0.09 + 0.8 \times (0.24 - 0.09)) = [0.135, 0.21) \\
ABBA &\mapsto [0.135 + 0 \times (0.21 - 0.135), 0.135 + 0.3 \times (0.21 - 0.135)) = [0.135, 0.1575)
\end{aligned}$$

As our smallest probability is 0.2, after 5 iterations our interval will have width at least $0.2^5 = 0.00032$, so 10^{-4} is enough accuracy for our codeword! We therefore wish to decode $1/\pi \approx 0.3183$, as

$$\begin{aligned}
0.3183 \in [0.3, 0.8) &\mapsto B \\
\frac{0.3183 - 0.3}{0.5} = 0.0366 \in [0, 0.3) &\mapsto A \\
\frac{0.0366 - 0}{0.3} = 0.122 \in [0, 0.3) &\mapsto A \\
\frac{0.122 - 0.3}{0.3} = 0.40667 \in [0.3, 0.8) &\mapsto B \\
\frac{0.40667 - 0.3}{0.5} = 0.21334 \in [0, 0.3) &\mapsto A
\end{aligned}$$

Hence $1/\pi \mapsto BAABA.....$ Observe that we do not need to construct the full codebook, but require high precision arithmetic here.

3. Consider a DMC with $\mathcal{X} = \mathcal{Y} = \{0, 1, 2, \cdots, 10\}$ and $M = (\mathbb{P}(Y = y|X = x))_{x \in \mathcal{X}, y \in \mathcal{Y}}$. It is known that $Y = (X + Z) \bmod 11$, where Z is independent of X and has pmf $p_Z(i) = \frac{1}{3}$ for $i \in \{1, 2, 3\}$. Find the capacity of this channel and the distribution of X that achieves the capacity.

Solution: $C = \max_{p_X}\{I(X, Y)\}$. For any p_X over $\mathcal{X}$,

$$\begin{aligned}
I(X, Y) &= H(Y) - H(Y|X) = H(Y) - H(X + Z|X) \\
&= H(Y) - H(Z|X) = H(Y) - H(Z) \\
&= H(Y) - \log(3) \\
&\leq \log(11) - \log(3),
\end{aligned}$$

and the equality holds in the last inequality if and only if Y follows the uniform distribution, which is realised when p_X is the uniform distribution.

Section B

4. Consider an alphabet $\mathcal{X} = A, B$ with pmf $p(A) = 1 - 2^{-3}$ and $p(B) = 2^{-3}$, and binary output alphabet $\mathcal{Y} = 0, 1$.

   (a) Construct the binary Shannon codes for blocks of length 1, 2 and 3.

   (b) For each of these codes, determine whether it is an optimal code.

   (c) Construct an optimal block code for this data with length three blocks. Compute its average blocklength and compare with $H(X)$, where $X \sim p$.

5. Let X be uniformly distributed over a finite set $\mathcal{X}$ with $|\mathcal{X}| = 2^n$ for some $n \in \mathbb{N}$. Given a sequence $A_1, A_2, \cdots$ of subsets of $\mathcal{X}$ we ask a sequence of questions of the form $X \in A_1, X \in A_2$, etc.

   (a) We can choose the sequence of subsets, but cannot vary them depending on the answers to previous questions. How many questions do we need to determine the value of X? What is the most efficient way to do so?

   [Note: If we regard all questions as a mapping from $\mathcal{X}$ to $\{\text{Yes}, \text{No}\}^*$, we can even think about how to design the sequence of subsets to minimise the expected number of questions to ask to get the value of a random variable X with any given distribution.]

   (b) We now randomly (i.i.d. and uniformly) draw a sequence of sets $A_1, A_2, \cdots$ from the set of all subsets of $\mathcal{X}$. Fix $x, y \in \mathcal{X}$. Conditional on $\{X = x\}$:

      (i) What is the probability that x and y are indistinguishable after the first k random questions?

      (ii) What is the expected number of elements in $\mathcal{X} \backslash \{x\}$ that are indistinguishable from x after the first k questions?

6. Consider a DMC $(\mathcal{X}, M, \mathcal{Y})$ with $|\mathcal{X}| = |\mathcal{Y}| = 3$ and the stochastic matrix

$$M = \begin{pmatrix} 2/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 \\ 0 & 1/3 & 2/3 \end{pmatrix}.$$

   (a) Calculate the capacity of this DMC.

   (b) Give an intuitive argument why the capacity is achieved with a distribution that places zero probability on an input symbol.

Sam Cohen: cohens@maths.ox.ac.uk

7. Consider the Block Arithmetic Code (BAC) constructed in lectures. Either modify the code provided, or develop your own implementation, which implements a version of the BAC with a one-step Markov model. In other words, the algorithm should follow the algorithm:

   - Accept as arguments a codeblock length, unconditional probability vector, and matrix of conditional probabilities.
   - For the first character in the block, use the unconditional probability distribution over symbols, and the BAC approximation, to split the set of codewords into initial character subsets
   - For subsequent characters, use the conditional probability distribution based on the previous character, and the BAC approximation, to split the set of codewords.

   You should have both an on-the-fly encoding and decoding method. An example of transition probabilities (for our familiar 27 character alphabet) is given in the file `transitions.csv`. This dataset is stored with the convention that entry $t(i, j)$ is the probability that $X_t = i$ given $X_{t-1} = j$. (You can check this by considering the column associated with 'Q'.)

   Using your method, for an output block of length 8, give the length of the encodings of `QQQQ` and `A_A_`, and comment on why these lengths are as expected.

Sam Cohen: cohens@maths.ox.ac.uk

Section C

8. Prove the following weaker version of the Kraft-McMillan theorem (called Kraft's theorem) using rooted trees

   (a) Let $c : \mathcal{X} \mapsto \{0, \cdots, d-1\}^*$ be a prefix code. Consider its code-tree and argue that $\sum_{x\in\mathcal{X}} d^{-|c(x)|} \leq 1$. [Note that the assumption that c is a prefix code is crucial here, otherwise the code-tree cannot be defined to begin with. In the Kraft-McMillan theorem from the lecture we only require c to be uniquely decodable].

   (b) Assume that $\sum_{x\in\mathcal{X}} d^{-l_x} \leq 1$ with $l_x \in \mathbb{N}$. Show that there exists a prefix code c with codeword lengths $|c(x)| = l_x$ for $x \in \mathcal{X}$ by constructing a rooted tree.

   Solution: A prefix code is equivalent to a rooted tree, where each codeword corresponds to a path from a leave to the root.

   (a) We call a d-ary tree semi-complete if every non-leaf vertex has d direct descendants. In a semi-complete d-ary tree for any leaf x, denote $h(x)$ as the height from the root to the tree with $h(root) = 0$. It is easy to check that $\sum_{\text{every leaf } x} d^{-h(x)} = 1$.

   For the code-tree of a prefix code, it can be expanded to a semi-complete tree by adding some leaves to a non-leave vertex. Hence $\sum_{\text{every leaf } x} d^{-h(x)} \leq 1$.

   (b) We call a d-ary tree complete with height h if it is semi-complete, the distance from each leaf to the root is h.

   Given l_x satisfies the condition, denote $h = \max_x l_x$, then we can construct a d-ary complete tree with maximal height h.

   Suppose $l_1 \leq l_2 \leq \cdots \leq l_m$. We mark nodes and cut branches of a complete tree as follows:

   (1) Take $i = 1$.

   (2) Find the first non-marked node on the left of the tree with height l_i, cut off its descendant vertices, and mark all ancestral vertices (including itself) and their edges down to the root.

   (3) Set $i = i + 1$ and repeat (2) until $i = m + 1$.

   For each x, we find a vertex with height l_x, cut off its descendant vertices and mark it the leaf of x, and mark all ancestral vertices and edges between the leaf x and the root.

   By the assumption $\sum_{i=1}^{m} d^{-l_i} \leq 1$, we know we can run this construction for all $k \leq m$ (otherwise, if we cannot find a node with height l_k at some $k \leq m$, then it must happen that $\sum_{i=1}^{k} d^{-l_x} > 1$).

The labels of all marked vertices and the i^{th} leaf in the algorithm corresponds to the codeword i.

9. For a set $\mathcal{X} = \{1, ..., m\}$ with corresponding pmf p, give a necessary and sufficient condition on p such that there exists a d-ary code with average per-character length $H_d(X)$.

Solution: We know that $H(X)$ is an upper bound on the average per-character length, and from the source coding theorem we have equality iff there exists a code with $c(x) = -\log_d(p(x))$ for all $x \in \mathcal{X}$. Therefore, clearly, it is necessary that all probabilities are (negative) powers of d.

Conversely, suppose all probabilities are negative powers of d, that is $p(x) = d^{-l_x}$. Let $l_{\max} = \max_x l_x$. We know that $1 = \sum_x d^{-l_x}$, in particular

$$1 = \sum_x d^{-l_x} = \sum_{x: l_x < l_{\max}} d^{-l_x} + d^{-l_{\max}} \#\{x : l_x = l_{\max}\}$$

and hence,

$$d^{l_{\max}-1} - \sum_{x: l_x < l_{\max}} d^{l_{\max}-l_x-1} = d^{-1} \#\{x : l_x = l_{\max}\}$$

The left hand side of this equation is an integer, so we know that $\#\{x : l_x = l_{\max}\}$ is divisible by d. We can therefore group the least likely symbols together, as in the Huffman construction, without using any symbols with higher probability. Repeating this argument, we see that the Huffman construction will always yield a codeword of length precisely l_x for each codeword. Finally, we observe that $\sum_x p_x l_x = -\sum_x p_x \log_d(p_x) = H_d(X)$, as desired.

10. **(Information theory and gambling)** Suppose m horses run a race, and the ith horse wins with probability p_i. An investment of one pound returns $o(i)$ pounds if horse i wins, otherwise the investment is lost. A gambler distributes all of his wealth across the horses: $b(i) \geq 0$ denotes the fraction of the gambler's wealth that he bets on horse i and $\sum_{i=1}^m b(i) = 1$. We now consider repeating this game over and over.

    If S_n denotes the gambler's wealth after the n^{th} race, then

    $$S_n = \Pi_{i=1}^n b(X_i)o(X_i),$$

    where X_i is the horse that wins the i^{th} race and $S_0 = 1$ is the start capital.

    (a) If X_i are i.i.d., show that for given $\mathbf{b} = (b(1), \cdots, b(m)), \mathbf{p} = (p_1, \cdots, p_m)$, the wealth evolves exponentially, i.e. $\lim_{n\to+\infty} \frac{1}{n} \log\left(\frac{S_n}{2^{nW(\mathbf{b},\mathbf{p})}}\right) = 0$ almost surely, where $W(\mathbf{b}, \mathbf{p})$ is to be determined. `[Hint: Strong law of large numbers.]`

    (b) Define $W^*(\mathbf{p}) := \max_{\mathbf{b}:\sum b(i)=1, b(i)\geq 0} W(\mathbf{b}, \mathbf{p})$ and find $\mathbf{b}$ that achieves this maximum. `[Hint: You can find a candidate by using Lagrange multipliers.]`

    (c) (Informal.) We can regard $q_i := \frac{1}{o(i)}$ as the "probabilities" the bookmaker implicitly assigns to $o(i)$ outcomes. Considering the cases $\sum q_i = 1$, $\sum q_i < 1$ and $\sum q_i > 1$, discuss the fairness of the game.

Solution:

(a) Since

$$\frac{1}{n}(\log(S_n/2^{nW(\mathbf{b},\mathbf{p})}) = \frac{1}{n}\sum_{i=1}^n \log(\mathbf{b}(X_i)o(X_i)) - W(\mathbf{b}, \mathbf{p}),$$

and by the law of large number

$$\lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^n \log(\mathbf{b}(X_i)o(X_i)) = \mathbb{E}[\log(\mathbf{b}(X_1)o(X_1))] = \sum_{i=1}^m \log(\mathbf{b}(i)o(i))p_i.$$

Hence $W(\mathbf{b}, \mathbf{p}) = \sum_{i=1}^m \log(\mathbf{b}(i)o(i))p_i$.

(b) By the last part, we have

$$\begin{aligned} W(\mathbf{b}, \mathbf{p}) &= \sum_{i=1}^m \log(\mathbf{b}(i)o(i))p_i \\ &= \sum_{i=1}^m \log(p(i)))p_i + \sum_{i=1}^m \log\left(\frac{\mathbf{b}(i))}{p_i}\right)p_i + \sum_{i=1}^m \log(o(i))p_i \\ &= -H(\mathbf{p}) - D(\mathbf{p}||\mathbf{b}) + \sum_{i=1}^m \log(o(i))p_i \\ &\leq -H(\mathbf{p}) + \sum_{i=1}^m \log(o(i))p_i, \end{aligned}$$

and the equality in the last inequality holds iff $\mathbf{b} = \mathbf{p}$.

(c) We know $W^*(\mathbf{p}) = \sum_{i=1}^m \log(p_i o(i)) p_i$. In terms of $q_i = \frac{1}{o(i)}$, we can write it into

$$W^*(\mathbf{p}) = \sum_{i=1}^m \log(p_i/q(i)) p_i.$$

Denote $K = \sum_{j=1}^m q_i$, then we can define $\hat{q}_i = \frac{q_i}{K}$, with which $\hat{\mathbf{q}} = (\hat{q}_1, \cdots, \hat{q}_m)$ is a pmf, and

$$\begin{aligned} W^*(\mathbf{p}) &= \sum_{i=1}^m \log(p_i/\hat{q}(i)) p_i - \log(K) \\ &= D(\mathbf{p}||\hat{\mathbf{q}}) - \log(K). \end{aligned}$$

In conclusion,

- If $K < 1$, then $W^*(\mathbf{p}) > 0$, which is favourable for the gambler;
- If $K = 1$, this game is still favourable unless $\mathbf{q}$ is parallel to $\mathbf{p}$.
- If $K > 1$, then this game can be favourable for the bookmaker.

Sam Cohen: cohens@maths.ox.ac.uk