3F7 Information Theory and Coding
Engineering Tripos 2021/22 – Solutions

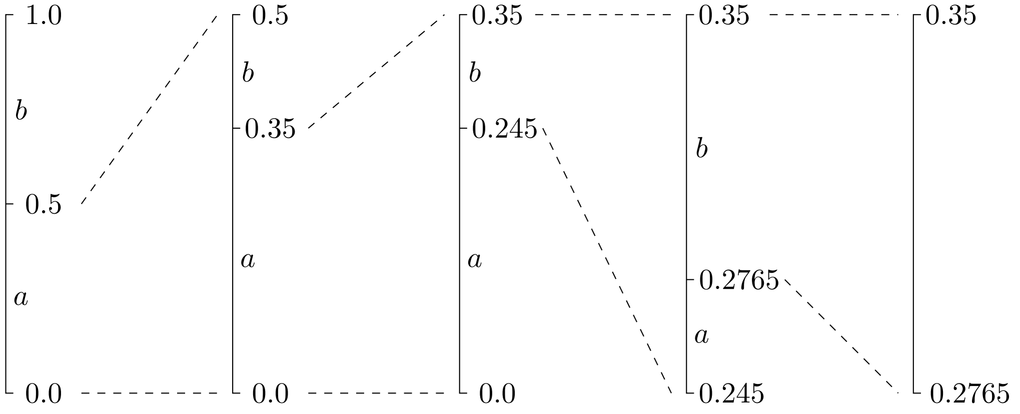
Question 1

(a) i) Noting that $P_Y(i) = P_X(\text{Heads})P(Y = i \,|\, X = H) + P_X(\text{Tails})P(Y = i \,|\, X = \text{Tails})$, for $i \in \{1, 2, 3, 4\}$ we have $P_Y(1) = P_Y(2) = \frac{1}{2} \cdot \frac{1}{3} = \frac{1}{6}$, $P_Y(3) = \frac{1}{2} \cdot \frac{1}{3} + \frac{1}{2} \cdot \frac{1}{2} = \frac{5}{12}$, $P_Y(4) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Using this, $H(Y) = \sum_{i=1}^{4} P_Y(i) \log_2 \frac{1}{P_Y(i)} = 1.89$ bits.

ii) We have $H(Y \mid X) = P_X(\text{Heads})H(Y \mid X = \text{Heads}) + P_X(\text{Tails})H(Y \mid X = \text{Tails}) = \frac{1}{2}\log_2(3) + \frac{1}{2}\log_2(2) = 1.29$ bits.

iii) From the chain rule, $H(X, Y) = H(Y \,|\, X) + H(X) = H(Y) + H(X \,|\, Y)$. Noting that $H(X) = 1$ bit, $H(X \mid Y) = H(Y \mid X) + H(X) - H(Y) = 1.29 + 1 - 1.89 = 0.4$ bits.

(b) i) We divide the first interval in arithmetic coding according to $P_{X_1}(a) = P_{X_1}(b) = \frac{1}{2}$ and subsequent intervals according to the conditional distributiond $P_{X_k|X_{k-1}}(.|.)$



The resulting interval satisfies $0.2765 < 9 \times 2^{-5} < 11 \times 2^{-5} < .35$ so contains two dyadic intervals of size 2^{-5} but no dyadic interval of size 2^{-4}. Hence we have a choice of two codewords of length 5 corresponding to the binary representation of 9 and 10, i.e., 0 1 0 0 1 and 0 1 0 1 0, respectively[1].

ii) A lower bound[2] on the expected codelength is the joint entropy

$$\begin{aligned} H(X_1, X_2, X_3, X_4) &= H(X_1) + H(X_2 \mid X_1) + H(X_3|X_2) + H(X_4|X_3) \\ &= H_2(0.5) + 3H_2(0.7) = 1 + 3 \times 0.88 = 3.64 \text{ bits.} \end{aligned}$$

iii) The upper bound for the expected length of arithmetic codeword is $H(X_1, X_2, X_3, X_4) + 2 = 5.64$ bits. This is not a useful bound because the expected code length is *larger* than the codelength without any compression (4 bits). Arithmetic coding is asymptotically optimal in the sense that the upper bound for the number of bits per symbol $\frac{H(X_1,\ldots,X_n)}{n} + \frac{2}{n}$ tends to the entropy as n grows large, but $n = 4$ is not large enough to yield any benefits even as compared to uncompressed transmission.

[1] We have used the convention that the interval corresponding to a is always the lower interval. As long as encoder and decoder are in agreement, it is possible to adopt other conventions, for example "the lower interval is always larger", resulting in a different outcome: interval $[0.245, 0.3185)$ satisfying $0.245 < 4 \times 2^{-4} < 5 \times 2^{-4} < 0.3185$ and hence a codeword of length 4 corresponding to the binary representation of 4, i.e., 0 1 0 0. There are other valid options.

[2] The examiner intended for students to compute the entropy in this question, which the vast majority did. However, since the question asks for the *minimum* rather than a lower bound, the precise answer to the question is the length of the optimal (Huffman) prefix-free code for the distribution of X_1, X_2, X_3, X_4. A few students computed this and were adequately rewarded in the marking scheme.

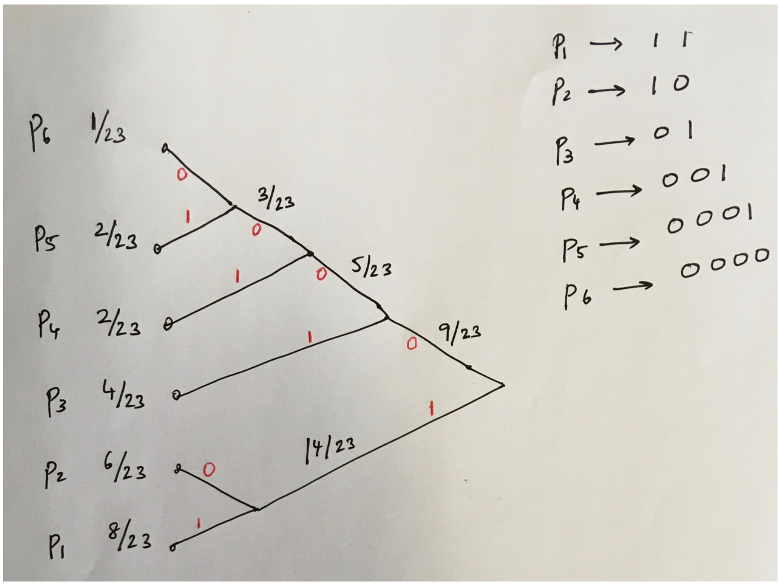
It was a fairly standard question on entropies and data compression and was well solved overall. Part (a) was done well by most with a few stumbling on calculator/mental arithmetic mistakes. In Part (b), performance on the arithmetic encoder was variable, with many struggling to pick the largest dyadic interval inside the source interval. The solution for this part is not unique as it depends on how you pick the interval ordering: any consistent way of doing so ('a' on top or 'a' below, largest interval always below) was accepted but those who arbitrarily picked the order or intervals in a way that could not logically be guessed by a decoder were marked down for it. There was a slight blunder in Part (b)(ii) where the examiner had intended for students to compute the block entropy, but the precise answer to the question as asked ("minimum" instead of "lower bound") was the Huffman code for the block probability distribution. Six students gave or attempted to give the precise answer to this question by computing the Huffman code and were adequately rewarded for this unforeseen extra effort in the marking scheme.

Question 2

(a) i) If we taste one bottle at a time, the expected number of tastings is minimized by first tasting the one most likely to be bad (bottle 1), then the next likeliest (bottle 2), and so on until bottle 5. We stop when we identify the bad bottle. The expected number of tastings for this order ($1 \to 2 \to 3 \to 4 \to 5$) is

$$\bar{T} = 1 \cdot \frac{8}{23} + 2 \cdot \frac{6}{23} + 3 \cdot \frac{4}{23} + 4 \cdot \frac{2}{23} + 5 \cdot \frac{2}{23} + 5 \cdot \frac{1}{23} = \frac{55}{23} = 2.3913$$

ii) When we are allowed to mix and taste, Huffman coding gives the optimal strategy. The Huffman code is given in the figure below.



We can translate this to a tasting strategy by first tasting a mixture to determine the first bit of the codeword (0/1), then the second bit and so on. The expected length is

$$\bar{T}^{*} = 2 \cdot \frac{8}{23} + 2 \cdot \frac{6}{23} + 2 \cdot \frac{4}{23} + 3 \cdot \frac{2}{23} + 4 \cdot \frac{2}{23} + 4 \cdot \frac{1}{23} = \frac{54}{23} = 2.3478$$

The sequence to identify the fourth bottle is: Taste mixture of bottles 1 and 2 (result is no); Taste bottle 3 (no); Taste bottle 4 (yes)

(b) i) The capacity of the channel is $\mathcal{C} = \max_P I(X;Y)$ where the maximum is over all input distributions over $\{0,1,2\}$. We have

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y \mid X) \\ &= H(Y) - \sum_{i=0}^{2} P(X=i) H(Y \mid X=i) \\ &\overset{(a)}{=} H(Y) - \sum_{i=0}^{2} P(X=i) H(\{0.7, 0.2, 0.1\}) \\ &= H(Y) - 1.157. \end{aligned}$$

Equality (a) above holds because each row of the transition probability matrix is a permutation of $\{0.7, 0.2, 0.1\}$ and hence has the same entropy. Now $H(Y) \leq \log_2 3$ with equality of all the Y symbols have equal probability ($\frac{1}{3}$), which is achieved with an equiprobable input distribution: $P(X=0) = P(X=1) = P(X=2) = \frac{1}{3}$. Therefore, $\mathcal{C} = \log_2 3 - 1.157 = 0.428$ bits.

ii) Fano's inequality (in the information databook) gives a lower bound on the probability of error P_e:

$$P_e \geq \frac{H(X|Y) - 1}{\log_2 |\mathcal{X}|} = \frac{H(X|Y) - 1}{\log_2 3}. \tag{1}$$

To calculate $H(X|Y)$ we can use the chain rule: $H(X|Y) + H(Y) = H(X) + H(Y|X)$. This gives

$$H(X|Y) = H(X) + H(Y|X) - H(Y) = H(\{0.4, 0.2, 0.4\}) + 1.157 - H(Y) = 1.5219 + 1.157 - H(Y)$$

where we use the value of $H(Y|X)$ computed above, noting that for this channel it does not depend on the input distribution. To compute $H(Y)$, we calculate the output distribution

$$\begin{aligned} &P(Y=0) = P_X(0)(0.7) + P_X(1)(0.1) + P_X(2)(0.2) = 0.38, \\ &P(Y=1) = P_X(0)(0.2) + P_X(1)(0.7) + P_X(2)(0.1) = 0.26, \quad P(Y=2) = 0.36. \end{aligned}$$

With this, we compute $H(Y) = 1.556$, which gives $H(X|Y) = 1.1125$. Using this in (1), we obtain the lower bound $P_e \geq 0.0710$.

Note: We can also use the stronger version of Fano's inequality, for which the denominator in (1) is $\log_2 |\mathcal{X} - 1|$ (rather than $\log_2 |\mathcal{X}|$); see Q.1 in Examples Paper 3. This gives the improved lower bound $P_e \geq 0.1125$. (Either version receives full marks.)

Part (a).(i) was done correctly by most candidates. Part (a).(ii) was generally well-answered, although some candidates did not realise that the optimal strategy was provided by Huffman coding. Part (b).(i) we done correctly by most candidates. Several candidates did not realise that in order to obtain the bound asked in Part (b).(ii) Fano's inequality was needed.

Question 3

(a) i) An (n,k) channel code of rate $R = \frac{k}{n}$ for the channel $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ consists of:

- A set of messages $\{1, \ldots, 2^k = 2^{nR}\}$,
- An encoding function $X^n : \{1, \ldots, 2^{nR}\} \to \mathcal{X}^n$ that assigns a codeword to each message. The set of codewords $\{X^n(1), \ldots, X^n(2^{nR})\}$ is called the codebook,
- A decoding function $g : \mathcal{Y}^n \to \{1, \ldots, 2^{nR}\}$, which produces a guess of the transmitted message for each received vector.

ii) The channel capacity formula is $\mathcal{C} = \max_{P_X} I(X;Y)$, where the maximum is computed over all distributions over the input alphabet $\mathcal{X}$. The channel coding theorem states that:

- Fix $R < \mathcal{C}$ and pick any $\epsilon > 0$. Then, for all sufficiently large n there exists a length-n code of rate R with error probability less than ϵ.

- Conversely, any sequence of length-n codes of rate R with probability of error tending to 0 as $n \to \infty$ must have $R \leq \mathcal{C}$.

(b) i) With a uniform input distribution over the set of 4-bit sequences

$$H(Y \mid X) = \sum_{x \in \mathcal{X}} \frac{1}{16} H(Y \mid X = x),$$

where $\mathcal{X}$ is the set of all 4-bit sequences. Now for any 4-bit input x, there are at most 4 possible outputs. Therefore, we have $H(Y \mid X = x) \leq \log_2 4 = 2$ for each $x \in \mathcal{X}$. However, there are some inputs such as $x = 0000$, for which the number of possible Y-sequences is smaller than 4: for such inputs x, $H(Y \mid X = x)$ is strictly smaller than 2. Therefore

$$H(Y \mid X) = \sum_{x \in \mathcal{X}} \frac{1}{16} H(Y \mid X = x) < 2.$$

ii) With a uniform input distribution all 4-bit inputs are equally likely. Since each of the four input bits is equally likely to be a 0 or 1, the deleted bit as well as the non-deleted bits are each equally likely to be 0 or a 1. This means that the induced output distribution assigns equal probability (1/8) to each of the 8 possible output sequences. Therefore $H(Y) = \log_2 8 = 3$ bits. Therefore, with a uniform input distribution,

$$I(X;Y) = H(Y) - H(Y \mid X) = 3 - H(Y \mid X) > 3 - 2 = 1 \text{ bit},$$

where the inequality uses the result in part(i) above. Therefore, the capacity $\mathcal{C}$ which is the mutual information (maximized over all input distributions) is strictly greater than 1 bit.

iii) The key is to observe that the four inputs with non-zero probability are non-confusable. This can be seen by listing the the possible outputs for each of these:

$$0000 \to \{000\}, \quad 0011 \to \{011, 001\}, \quad 1100 \to \{100, 110\}, \quad 1111 \to \{111\}.$$

With the given input distribution, since the output Y uniquely determines the input X, $H(X \mid Y) = 0$ and therefore

$$I(X;Y) = H(X) - H(X \mid Y) = \log_2 4 - 0 = 2.$$

Since the capacity is the maximum over all possible input distributions, it is at least 2 bits.

iv) The input distribution in part (iii) does not use the outputs 101 and 010. We expect that an optimal input distribution will induce an output distribution with nonzero probabilities over all the 3-bit outputs. Hence we expect that above input distribution is not optimal, and therefore the capacity is strictly greater than 2 bits. We expect that the optimal input distribution will assign larger probability $P(X = x)$ to inputs x that have smaller values of $H(Y \mid X = x)$ (such as 0000 and 1111), so as to minimize $H(Y|X)$. (But this is just a heuristic, and needs to be verified numerically).

Part (a) was surprisingly not well answered; most answers lacked precision in their description in Part (a)(i) and statement of the channel coding theorem Part (a)(ii). Part (b).(i) was not answered well by most candidates as some found it difficult to realise that for any x, $H(Y|X = x) < 2$. Part (b)(ii) was answered correctly by many candidates. Some candidates realised that $H(Y|X = x) = 0$ for every x in Part (b)(iii), but not all. The discussion in Part (b)(iv) was mixed.

Question 4

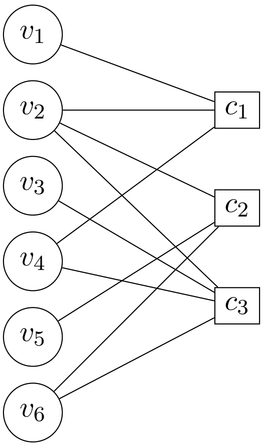
(a) Code dimension $k = 3$, rate $R = \frac{1}{2}$.

(b) $\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \xrightarrow{R_3=R_1+R_3} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{R_2=R_2+R_3} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = \mathbf{H}_{sys}$

(c) If $\mathbf{H}_{sys} = [P^T \mid I]$, then $\mathbf{G}_{sys} = [I \mid P] = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$

There are 8 codewords of the form $[x_1, x_2, x_3]\mathbf{G}_{sys}$, for $x_1, x_2, x_3 \in \{0, 1\}$. The codewords are $[0,0,0,0,0,0]$, $[1,0,0,1,1,1]$, $[0,1,0,1,1,0]$, $[0,0,1,0,1,1]$, $[1,1,0,0,0,1]$, $[1,0,1,1,0,0]$, $[0,1,1,1,0,1]$, and $[1,1,1,0,1,0]$.

(d) Factor graph:



(e) (i) A maximum likelihood codeword is any of the codewords with smallest Hamming distance from the received word, e.g. $[1,1,1,0,1,0]$ ($[0,1,1,1,0,1]$ and $[1,0,0,1,1,1]$ are also valid answers.)

ii) We have

$$\frac{P(v_3 = 0 \mid \underline{y})}{P(v_3 = 1 \mid \underline{y})} = \frac{\sum_{\underline{c}:v_3=0} P(\underline{y} \mid \underline{c})}{\sum_{\underline{c}:v_3=1} P(\underline{y} \mid \underline{c})} = \frac{(0.2)^6 + (0.2)^2(0.8)^4 + 2(0.2)^3(0.8)^3}{2(0.2)^2(0.8)^4 + 2(0.2)^3(0.8)^3} = \frac{0.0246}{0.0410} = 0.6.$$

In the above, the second equality is obtained by noting that codewords 1,2,3, and 5 have a 0 in the third bit, and codewords 4,6,7,8 have a 1 in the third bit. Since the likelihood ratio is larger than 1, the bit is decoded as a 1.

iii) The third code bit connected only to the third check node. Therefore, the final LLR for the third code bit is

$$L_3 = L(y_3) + L_{c_3 \to v_3},$$

where $L(y_3)$ is the channel LLR for the third code bit (based on y_3), and $L_{c_3 \to v_3}$ is the message sent by the third check node to the third v-node. We have

$$L(y_3) = \ln \frac{P(y_3 = 1 \mid c_3 = 0)}{P(y_3 = 1 \mid c_3 = 1)} = \ln \frac{0.2}{0.8} = -1.386.$$

The message $L_{c_3 \to v_3}$ is determined by the initial LLRs received from v_2, v_4, v_6 (the v-nodes other than v_3 connected to the third check node):

$$L_{c_3 \to v_3} = 2 \tanh^{-1} \left[\left(\tanh \frac{-1.386}{2} \right)^3 \right] = -0.439$$

Hence $L_3 = L(y_3) + L_{c_3 \to v_3} = -1.825$. Since the final LLR is negative, the third code bit will be decoded as a 1.

iv) The log of the likelihood ratio we computed in part (ii) is $L_3^\star = -0.5108$, which has the same sign of the LLR L_3 computed by one iteration of the sum product algorithm in (iii). Accordingly, the two decoders agree on the value of bit 3. Since variable 3 is only involved in one check node in the factor graph, after one iteration the sum-product decoder only has the contribution from that parity-check equation, which happens to be satisfied by the received word whereas the other two parity-check equations aren't. The two decoders operate differently. The bitwise optimal decoder takes into account the whole codebook, while sum-product decoding takes only local information into account. If further iterations of the sum-product decoder were performed, it may well be that the magnitude of the LLR becomes similar to that of the bitwise optimal decoder.

Generally well-answered question, although very few candidates scored full marks. Most candidates correctly answered Parts (a), (b), (c), (d) and (e)(i). Very few candidates derived the optimal bitwise decoder in Part (e)(ii) and instead calculated the uncoded likelihood. Many candidates answered correctly Part (e)(iii). The discussion in Part (e)(iv) was mixed, as a result of many candidates considering the uncoded likelihood in Part (e)(ii).