3F7 Information Theory and Coding
Engineering Tripos 2022/23 – Solutions

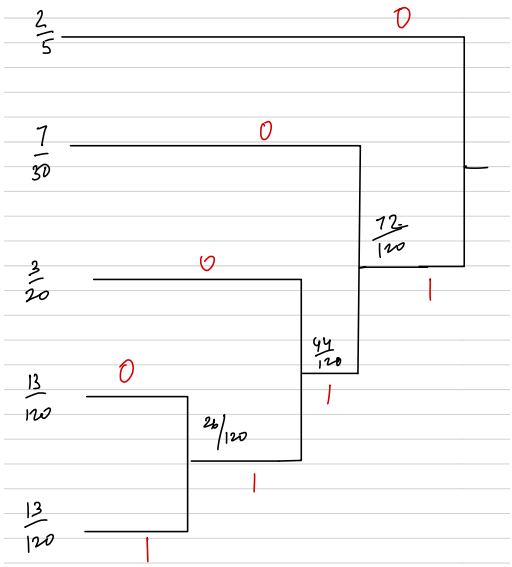
Question 1

(a) For $x \in \mathcal{U} = \{a, b, c, d, e\}$, we have $P_X(x) = \frac{1}{3}P_1(x) + \frac{2}{3}P_2(x)$. Using this we obtain: [15%]

$$P_X(a) = \frac{1}{15} + \frac{1}{3} = \frac{2}{5}, \quad P_X(b) = \frac{1}{15} + \frac{1}{6} = \frac{7}{30}, \quad P_X(c) = \frac{1}{15} + \frac{1}{12} = \frac{3}{20},$$
$$P_X(d) = P_X(e) = \frac{1}{15} + \frac{1}{24} = \frac{13}{120}.$$

(b) Huffman coding provides an optimal symbol code. With the Huffman tree shown below, the codewords are:

$$a \to 0, \quad b \to 10, \quad c \to 110, \quad d \to 1110, \quad e \to 1111.$$



The expected codelength is $L = \sum_{x \in \mathcal{U}} P_X(x)\ell(x) = 2.1833$ bits/symbol. [25%]

(c) The entropy $H(X) = -\sum_{x \in \mathcal{U}} P_X(x) \log_2 P_X(x) = 2.124$ bits. An upper bound on the length of the arithmetic codeword is $nH(X) + 2 = 2.124n + 2$ bits, or $2.124 + \frac{2}{n}$ bits/symbol. [10%]

(d) Suppose that n_1 of the n source symbols $X_1, X_2, \ldots, X_n$ came from source U_1 and the remaining $(n - n_1)$ from U. We note that $n_1 \approx \frac{n}{3}$ for large n. Since both the encoder and them decoder know the selector sequence, we can split the sequence into two subsequences – one with length n_1 with the symbols from U_1 and the other with length n_2 with the symbols from U_2 – and compress them separately with arithmetic encoders for P_1 and P_2, respectively. They can be decoded separately with the corresponding arithmetic decoders, and the sequence $X_1, \ldots, X_n$ can then be reconstructed using knowledge of the selector sequence.

The expected length of the codeword for $(X_1, \ldots, X_n)$ with this scheme satisfies: [20%]

$$L_n \leq \left(n_1 H(U_1) + \frac{2}{n_1}\right) + + \left((n - n_1)H(U_2) + \frac{2}{n - n_1}\right)$$
$$= n_1 \log_2 5 \; + \; (n - n_1)(1.875) \; + \; \frac{2}{n_1} \; + \; \frac{2}{(n - n_1)}.$$

As $n \to \infty$, we have $n_1/n \to \frac{1}{3}$. Therefore, the expected number of bits per source symbol as $n \to \infty$:

$$L = \lim_{n\to\infty} \frac{L_n}{n} = \frac{1}{3}H(U_1) + \frac{2}{3}H(U_2) = 2.024 \text{ bits/symbol.}$$

Alternative scheme: For each symbol X_i, the arithmetic encoder could use sub-intervals defined via the appropriate source distribution, i.e., P_1 if the symbol X_i came from U_1 and P_2 if it came from U_2. In this case, expected codeword length satisfies $L_n \leq n_1 H(U_1) + (n - n_1)H(U_2) + \frac{2}{n}$.

(e) i) The entropy for U_2 is

$$H(U_2) = \frac{1}{2}\log(2) + \frac{1}{4}\log(4) + \frac{1}{8}\log(8) + \frac{1}{16}\log(16) + \frac{1}{16}\log(16) = 1.875.$$

The size of the typical set for the source U_2 is close to $2^{mH(U_2}$. With high probability, the sequence $\underline{Z} = (Z_1, \ldots, Z_m)$ will belong to the typical set. A rate R channel code of block length n has 2^{nR} codewords. Therefore, if $2^{nR} > 2^{mH(U_2)}$, then we can assign a unique channel codeword to each source sequence in the typical set. Hence, we need [15%]

$$nR > mH(U_2) \quad \Rightarrow \quad n > \frac{1.875m}{R}.$$

ii) For transmission at rate $R < \mathcal{C}$, we pick a code length $n = 1.05 \times \frac{1.875m}{R}$ (we need to round up to an integer, but the effect is negligible for large m). The total number of channel codewords is $2^{nR} = 2^{1.97m}$. The total number of source sequences is 5^m which is larger than $2^{1.97m}$. Therefore there are (many) source sequences that cannot be assigned a unique codeword.

However for any $\epsilon > 0$ and sufficiently large m, the observed $(Z_1, \ldots, Z_m)$ will belong to the typical set with probability at least $(1 - \epsilon)$. Moreover the typical set has size at most $2^{m(H(U_2)+\epsilon)}$ (see lecture notes). Since $2^{m(H(U_2)+\epsilon)} = 2^{m(1.875+\epsilon)} < 2^{1.97m}$ for small values of ϵ, with high probability the observed source sequence will belong to the typical set and hence be assigned a unique codeword. Then with code length n and an optimal channel encoder and decoder, since $R < \mathcal{C}$ we know that the transmitted codeword (and hence the source sequence) can be recovered with high probability. [15%]

Assessor's comment: Very few students gave correct answers for part (e), where the key point is that to reconstruct the source sequence at the receiver with high probability, we only need to assign and transmit codewords corresponding to typical source sequences. The set of typical source sequences is much smaller than the set of all possible sequences.

.

Question 2

(a) i) Z takes values in $\{0, 1, \ldots, r\}$. For we have [10%]

$$P_Z(0) = P_X(0) = p, \qquad P_Z(z) = P_X(1)P_Y(r) = (1-p)q_z, \quad z \in \{0, 1, \ldots, r\}.$$

ii) The entropy of Z is [10%]

$$\begin{aligned} H(Z) &= p \log \frac{1}{p} + \sum_{z=1}^{r} (1-p) q_z \log \frac{1}{(1-p)q_z} \\ &= p \log \frac{1}{p} + (1-p) \log \frac{1}{(1-p)} + (1-p) \sum_{z=1}^{r} q_z \log \frac{1}{q_z} = H(X) + (1-p)H(Y). \end{aligned}$$

(b) The code has parameters $n = 5$ and $k = 1$, i.e., two codewords of length 5. There are four possible codes, one for each pair of values of a and b. For each code we evaluate the two codewords as $x\mathbf{G}$ for $x \in \{0, 1\}$, and check whether $\lceil \frac{d_{\min}-1}{2} \rceil \geq 2$ as we know that $\lceil \frac{d_{\min}-1}{2} \rceil$ is the guaranteed error correcting capability of the code. [30%]

1) $a = 0, b = 0$: Here $\mathbf{G} = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$, and the two codewords are identical: $[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$. This code is trivial and clearly cannot correct errors.

2) $a = 1, b = 0$. The two codewords are $[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$ and $[1 \;\; 1 \;\; 1 \;\; 1 \;\; 0]$. We have $d_{\min} = 4$, therefore $\lceil \frac{d_min-1}{2} = 1$, therefore it cannot correct all patterns of two channel errors. (For example, decoding fails when any bits of first four bits the codeword are flipped.)

3) $a = 0, b = 1$: The two codewords are $[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$ and $[0 \;\; 0 \;\; 0 \;\; 0 \;\; 1]$. We have $d_{\min} = 1$, so the code cannot correct any errors.

4) $a = 1, b = 1$: The two codewords are $[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$ and $[1 \;\; 1 \;\; 1 \;\; 1 \;\; 1]$. We have $d_{\min} = 5$, therefore $\lceil \frac{d_{\min}-1}{2} \rceil = 2$. The code can correct two channel errors.

Since $a = 1, b = 1$ is the only case where two errors can be corrected, the probability that the generated code can correct two errors is $P(a = 1)P(b = 1) = \frac{2}{3}\frac{2}{3} = \frac{4}{9}$.

(c) i) We have [10%]

$$\begin{aligned} P_{Y|X}(0 \mid 0) = P_{Y|X}(1 \mid 1) &= P(S = p)(1-p) + P(S = q)(1-q) = 1 - \frac{(p+q)}{2} \\ P_{Y|X}(1 \mid 0) = P_{Y|X}(0 \mid 1) &= P(S = p)p + P(S = q)q = \frac{(p+q)}{2}. \end{aligned}$$

ii) Since $P_{Y|X}$ is a binary symmetric channel with crossover probability $\frac{(p+q)}{2}$, its capacity is $1 - H_2((p+q)/2)$ where $H_2(\cdot)$ is the binary entropy function. [10%]

iii) Since the state sequence is known at both the encoder and the decoder, the capacity formula is $\max I(X; Y|S)$ where the maximum is over all input distributions of the form $P_{X|S}$. We have [20%]

$$I(X; Y|S) = H(Y|S) - H(Y|X, S). \tag{1}$$

We have

$$H(Y|X, S) = P(S = p)H(Y|X, S = p) + P(S = q)H(Y|X, S = q) = \frac{1}{2}H_2(p) + \frac{1}{2}H_2(q).$$

Note that $H(Y|S) \leq 1$ since Y is binary valued. Since the channel acts symmetrically on $X = 1$ and $X = 0$ (for either value of S), the symmetric input distribution $P_{X|S}(0|s) = P_{X|S}(1|s) = \frac{1}{2}$ results in $P_{Y|S}(0|s) = P_{Y|S}(1|s) = \frac{1}{2}$, for $s \in \{p, q\}$. Thus, with this input distribution $H(Y|S) = \frac{1}{2}H(Y|S = p) + \frac{1}{2}H(Y|S = q) = 1$. Therefore, from (1), the capacity is $\mathcal{C} = 1 - \frac{(H_2(p)+H_2(q))}{2}$.

iv) Out of n channel uses, let n_p be the number of channel uses where the crossover probability is p. For the remaining $(n - n_p)$ the crossover probability is q. Note that $n_p/n \to \frac{1}{2}$ as $n \to \infty$. Since the state sequence is known to both encoder and decoder, we can use two separate capacity-achieving codes, one of length n_p for the BSC(p) and the other of $(1 - n_p)$ for the BSC(q). Since the capacities of these two channels are $(1-H_2(p))$ and $(1-H_2(q))$ bits/transmission, respectively, the overall transmission rate can be arbitrarily close to $\mathcal{C} = 1 - \frac{(H_2(p)+H_2(q))}{2}$ [10%]

Assessor's comment: A lot of students did not interpret part (b) correctly, with some being confusing code" with "codeword". Recall that a $k \times n$ generator matrix defines a code which consists of 2^k codewords each of length n

Question 3

(a) i) FALSE. Since $H(Y^n \mid Z) = H(Y^n \mid g(Y^n))$, this conditional entropy is zero only when $g(Y^n)$ is an invertible function, i.e., Y^n can be recovered from Z. In general, this is not possible (e.g., when $Z = \sum_{i=1}^n Y_i$)) . [10%]

ii) TRUE. Using the formula of mutual information and the chain rule for entropy, we have [15%]

$$I(X^n\,;Y^n) = H(X^n) - H(X^n|Y^n) = \sum_{i=1}^{n} H(X_i|X^{i-1}) - H(X^n) \tag{2}$$

Since each X_i is binary, we have $H(X_i) \leq 1$, and since conditioning cannot increase the entropy, $H(X_i|X^{i-1}) \leq H(X_i) \leq 1$ for $i = 1, \ldots, n$. Using this for each term in (2), we obtain $I(X^n\,;Y^n) \leq n - H(X^n)$.

iii) TRUE. Since $H(g(Y^n)) \leq H(Y^n)$ for any function g, we have: [10%]

$$H(Z) = H(g(Y^n)) \leq H(Y^n) = \sum_{i=1}^{n} H(Y_i|Y^{i-1}) \leq \sum_{i=1}^{n} 1 = n.$$

where the inequality above holds because Y_i is binary, and since conditioning cannot increase entropy $H(Y_i|Y^{i-1}) \leq H(Y_i) = 1$.

(b) The capacity of the channel is

$$\max_{P_X} I(X;Y) = H(Y) - H(Y|X),$$

where P_X is a distribution over 8-bit vectors. Since there are 8 equiprobable outputs for each input vector X, we have have

$$H(Y|X) = \sum_{\underline{x}\in\{0,1\}^8} P_X(\underline{x}) H(Y|X = \underline{x}) = \log_2 8 = 3.$$

Since Y is an eight bit vector, it can take on at most 2^8 different values, hence $H(Y) \leq \log_2 2^8 = 8$. Therefore [30%]

$$I(X;Y) = H(Y) - 3 \leq 8 - 3 = 5. \tag{3}$$

Suppose we take P_X to be the uniform distribution assigning equal probability ($\frac{1}{256}$) to all eight-bit vectors. Then for each eight bit vector $\underline{y} \in \{0,1\}^8$, let $\mathcal{S}(\underline{y})$ denote the set of input vectors which can result in $\underline{y}$ (by flipping exactly one bit). Note that for any $\underline{y}$, there are exactly 8 vectors in $\mathcal{S}(\underline{y})$. We therefore have

$$P_Y(\underline{y}) = \sum_{\underline{x}\in\mathcal{S}(\underline{y})} P_X(\underline{x}) P_{Y|X}(\underline{y}|\underline{x}) = \sum_{\underline{x}\in\mathcal{S}(\underline{y})} \frac{1}{256} \cdot \frac{1}{8} = \frac{1}{256},$$

which gives $H(Y) = 8$. Therefore, with the uniform input distribution, we can achieve equality in (3) and the capacity is $\mathcal{C} = 5$ bits/channel use.

(c) The Hamming code maps $k = 4$ information bits to 7 code bits and has the parity check matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

From the parity check matrix, we observe that the code has minimum distance $d_{\min} = 3$ (as the minimum number of columns that add up to get the all-zeros vector is 3). Hence the code can correct one error in a block of 7 code bits. [35%]

For the 8-bit vector channel, we can transmit five information bits $(b_1, b_2, b_3, b_4, b_5)$ as follows.

Encoder: Encode the first 4 bits (b_1, b_2, b_3, b_4) into a 7-bit Hamming codeword denoted by $(x_1, \ldots, x_7)$. Set $x_8 = b_5$. The input $\underline{x} = (x_1, \ldots, x_8)$ is transmitted over the 8-bit channel.

Decoder: The output symbol is $\underline{y} = (y_1, \ldots, y_8)$. Using the Hamming code, decode the first 7 bits $(y_1, \ldots, y_7)$ to a Hamming codeword and the corresponding information sequence is $(\hat{b}_1, \ldots, \hat{b}_4)$. If there has been a bit flip in the first 7 bits $(y_1, \ldots, y_7)$, then set $\hat{b}_5 = \bar{y}_8$ (i.e., the complement), otherwise set $\hat{b}_5 = y_8$.

The last step works because we know that the 8-bit channel flips exactly one bit, and the Hamming code is guaranteed to correct the bit flip if it occurred in the first 7 bits. If it did not, the bit flip must be in the eighth bit. In this way, we can transmit 5 bits error-free in each use of the eight-bit channel.

Question 4

(a) i)Each parity check equation corresponds to a row of the parity check matrix. [15%]

Therefore the $1 \times n$ parity check matrix is given by:

$$\mathbf{H} = [1\ 1\ \ldots\ 1].$$

The code rate is $(n-1)/n$.

ii) Using the formula (in Information data book) relating systematic generator and parity check matrices, the $(n-1) \times n$ systematic generator matrix is given by

$$\mathbf{G} = \left[\mathbb{I}_{(n-1)\times(n-1)} \quad \mathbf{1}_{n-1}\right],$$

where $\mathbb{I}_{(n-1)\times(n-1)}$ is the $(n-1) \times (n-1)$ identity matrix, and $\mathbf{1}_{n-1}$ is the all-ones column vector of length $(n-1)$. [10%]

(b) i) The $1 \times n$ generator matrix for this code is given by [10%]

$$\mathbf{G} = [1\ 1\ \ldots\ 1].$$

ii) As above, the corresponding $(n-1) \times n$ systematic parity check matrix is [10%]

$$\mathbf{H} = \left[\mathbf{1}_{n-1} \quad \mathbb{I}_{(n-1)\times(n-1)}\right],$$

(c) From the figure in the question, we have $g(y) = 0.25(2-y)$ for $y \in [0,2]$ and $g(y) = 0.25(2+y)$ for $y \in [-2,0]$. Therefore,

$$f(y \mid x=1) = g(y-1) = \begin{cases} 0.5 - \frac{(y-1)}{4} = 0.75 - 0.25y, & y \in (1,3) \\ 0.5 + \frac{(y-1)}{4} = 0.25 + 0.25y, & y \in [-1,1] \\ 0, & \text{otherwise.} \end{cases}$$

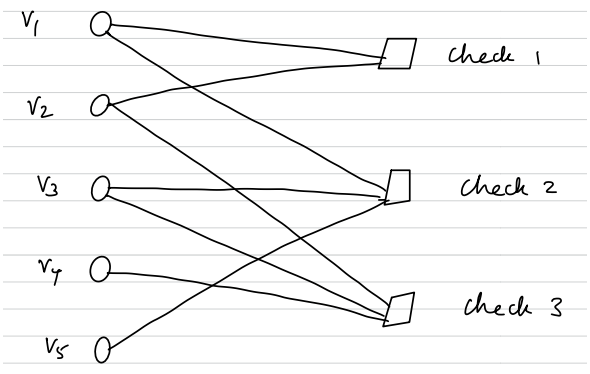
And, [20%]

$$f(y \mid x=-1) = g(y+1) = \begin{cases} 0.5 - \frac{(y+1)}{4} = 0.25 - 0.25y, & y \in [-1,1] \\ 0.5 + \frac{(y+1)}{4} = 0.75 + 0.25y, & y \in (-3,-1) \\ 0, & \text{otherwise.} \end{cases}$$

Therefore the likelihood ratio is

$$\frac{f(y \mid x=1)}{f(y \mid x=-1)} = \begin{cases} \infty, & y \in (1,3) \\ \frac{(1+y)}{(1-y)}, & y \in [-1,1] \\ 0, & y \in (-3,-1). \end{cases}$$

(d) The factor graph for the code is as shown below.



Using part (c), the initial LLRs for the five code bits are

$$L(y_1) = 0.2007, \quad L(y_2) = 0.619, \quad L(y_3) = -0.8473, \quad L(y_4) = 2.944, \quad L(y_5) = -\infty.$$

The second code bit receives messages from check 1 and check 3. Therefore its final LLR after one complete iteration of message passing is

$$L_2 = L(y_2) + L_{c_1 \to v_2} + L_{c_3 \to v_2} = 0.619 + L_{c_1 \to v_2} + L_{c_3 \to v_2}.$$

From the message passing rules we have: [35%]

$$L_{c_1 \to v_2} = 2 \tanh^{-1} \left[\tanh(L(y_1)/2) \right] = 0.2007$$
$$L_{c_3 \to v_2} = 2 \tanh^{-1} \left[\tanh(L(y_3)/2) \tanh(L(y_4)/2) \right] = -0.7537$$

Therefore, the final LLR for code bit 2 is:

$$L_2 = 0.619 + 0.2007 - 0.7537 = -0.066.$$

That is, after one complete iteration, the decoder thinks the second codebook is slightly more likely to be a 1 rather than a 0. (We say slightly more likely because the magnitude of the LLR is small.)

Assessor's comment: Well done overall, but many forgot to convert the likelihood values derived using part (d) to log-likelihood format before applying message passing.