Final Exam - Moed B

Total time for the exam: 3 hours!

Please copy the following sentence and sign it:

" I am respecting the rules of the exam: Signature:_____ "

1) **Word-Guessing and Model Mismatch (35 Points):** You are playing a word-guessing game. There are 1024 possible words, all equally likely.

(a) (5 points) Before making any guesses, what is the entropy of the random variable representing the word?

Solution:

All words are equally likely on a set of size 1024, so

$$H(W) = \log_2(1024) = 10 \text{ bits.}$$

(b) (10 points) You guess the **first letter** of the word. You are debating between two guesses: T (appears in $1/4$ of the words) vs L (appears in $1/8$ of the words). You may use $\log_2(768) \approx 9.6$ and $\log_2(896) \approx 9.8$. Which letter gives a better average reduction in entropy? By how much?

Solution:

Guess T:

$P(T = \text{correct}) = \frac{1}{4}, \quad P(T = \text{incorrect}) = \frac{3}{4}$

$H(W \mid T = \text{correct}) = \log_2(1024 \cdot \frac{1}{4}) = 8$

$H(W \mid T = \text{incorrect}) = \log_2(1024 \cdot \frac{3}{4}) \approx 9.6$

Thus:

$$H(W \mid T) = \tfrac{1}{4} \cdot 8 + \tfrac{3}{4} \cdot 9.6 = 2 + 7.2 = 9.20 \text{ bits.}$$

Reduction: $\Delta_T = 10 - 9.20 = 0.80$ bits.

Guess L:

$P(L = \text{correct}) = \frac{1}{8}, \quad P(L = \text{incorrect}) = \frac{7}{8}$

$H(W \mid L = \text{correct}) = \log_2(1024 \cdot \frac{1}{8}) = 7$

$H(W \mid L = \text{incorrect}) = \log_2(1024 \cdot \frac{7}{8}) \approx 9.8$

Thus:

$$H(W \mid L) = \tfrac{1}{8} \cdot 7 + \tfrac{7}{8} \cdot 9.8 = 0.875 + 8.575 = 9.45 \text{ bits.}$$

Reduction: $\Delta_L = 10 - 9.45 = 0.55$ bits.

Conclusion: Guessing T yields 0.25 bits more reduction than guessing L.

(c) (10 points) Now consider guessing the letter **R**, which starts **half of the words**. However, 10% of the time you guess R, the computer will not respond at all (the guess is still used). Which is better: guessing R, or your better choice from part (b)? Justify.

Solution:

Probabilities:

$$P(R = \varnothing) = 0.1, \quad P(R = \text{correct}) = 0.9 \cdot \tfrac{1}{2} = 0.45, \quad P(R = \text{incorrect}) = 0.45.$$

Conditional uncertainties:

$$H(W \mid R = \varnothing) = 10, \quad H(W \mid R = \text{correct}) = \log_2(1024 \cdot \tfrac{1}{2}) = 9, \quad H(W \mid R = \text{incorrect}) = 9.$$

Overall:

$$H(W \mid R) = 0.1 \cdot 10 + 0.45 \cdot 9 + 0.45 \cdot 9 = 1.0 + 4.05 + 4.05 = 9.10 \text{ bits.}$$

Reduction: $\Delta_R = 10 - 9.10 = 0.90$ bits.

Conclusion: Guessing R yields 0.10 bits more reduction than guessing T from part (b).

(d) (10 points) You are given a file containing the letters $\{a, b, c, d\}$ whose empirical distribution is

$$P = \left[\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right].$$

You wish to compress this file using a code optimized for one of the following fixed probability models:

$$Q_1 = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right], \quad Q_2 = \left[\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}\right].$$

Which *information measure* should be used to determine which model (Q_1 or Q_2) yields better compression when the true source distribution is P? Compute its value for both Q_1 and Q_2, and state which model is better, with a brief explanation.

Solution:

Both the **cross-entropy** $H(P, Q) = -\sum_i P_i \log_2 Q_i$ and the **Kullback–Leibler divergence** $D_{\mathrm{KL}}(P\|Q) = \sum_i P_i \log_2 \frac{P_i}{Q_i}$ are valid measures for model mismatch in compression. They are related by

$$H(P, Q) = H(P) + D_{\mathrm{KL}}(P\|Q),$$

so minimizing cross-entropy is equivalent to minimizing KL divergence when P is fixed.

Source entropy.

$$H(P) = -\left(\tfrac{1}{2}\log_2\tfrac{1}{2} + \tfrac{1}{4}\log_2\tfrac{1}{4} + \tfrac{1}{8}\log_2\tfrac{1}{8} + \tfrac{1}{8}\log_2\tfrac{1}{8}\right) = \tfrac{1}{2}\cdot 1 + \tfrac{1}{4}\cdot 2 + \tfrac{1}{8}\cdot 3 + \tfrac{1}{8}\cdot 3 = 1.75 \text{ bits.}$$

KL divergences.

$$\begin{aligned} D_{\mathrm{KL}}(P\|Q_1) &= \tfrac{1}{2}\log_2\frac{1/2}{1/4} + \tfrac{1}{4}\log_2\frac{1/4}{1/4} + \tfrac{1}{8}\log_2\frac{1/8}{1/4} + \tfrac{1}{8}\log_2\frac{1/8}{1/4} \\ &= \tfrac{1}{2}\cdot 1 + 0 + \tfrac{1}{8}\cdot(-1) + \tfrac{1}{8}\cdot(-1) = 0.25 \text{ bits,} \\ D_{\mathrm{KL}}(P\|Q_2) &= \tfrac{1}{2}\log_2\frac{1/2}{1/2} + \tfrac{1}{4}\log_2\frac{1/4}{1/8} + \tfrac{1}{8}\log_2\frac{1/8}{1/8} + \tfrac{1}{8}\log_2\frac{1/8}{1/4} \\ &= 0 + \tfrac{1}{4}\cdot 1 + 0 + \tfrac{1}{8}\cdot(-1) = 0.125 \text{ bits.} \end{aligned}$$

Cross-entropies (optional check).

$$H(P, Q_1) = H(P) + D_{\mathrm{KL}}(P\|Q_1) = 1.75 + 0.25 = 2.00 \text{ bits,} \qquad H(P, Q_2) = 1.75 + 0.125 = 1.875 \text{ bits.}$$

Conclusion. Since $D_{\mathrm{KL}}(P\|Q_2) < D_{\mathrm{KL}}(P\|Q_1)$ (equivalently, $H(P, Q_2) < H(P, Q_1)$), the model Q_2 yields better expected code length and thus better compression for source P.

2) **Perfect Secrecy (35 Points):** Consider a communication scenario where Alice wants to send a message M, randomly drawn from a finite set $\mathcal{M}$, to Bob. To keep the message hidden from eavesdroppers, she encrypts it using a secret key $K \in \mathcal{K}$ that is known *only* to Alice and Bob and is independent of M. The encryption is performed using a deterministic function $C = f(K, M)$, producing encrypted message $C \in \mathcal{C}$. Bob decrypts the encrypted message using another deterministic function $M = g(K, C)$, and throughout the question we assume that such a decryption function g exists. The system is said to achieve *perfect secrecy* if $I(M; C) = 0$.

a) **(5 points)** Briefly explain why a perfectly secure system is safe from an eavesdropper.

Solution:

In a perfectly secure system, the message M and the encrypted message C are statistically independent. This means that knowing C provides no information about M, so an eavesdropper who sees C cannot learn anything about the original message.

b) **(7 points) True/False:** Under any system (whether secure or not), it holds that $H(M \mid C) \leq H(K \mid C)$.[3]
Hint: Use the assumption that there exits g such that $M = g(K, C)$.
Solution:
True. Since $M = g(K, C)$, we have:

$$\begin{aligned} H(K \mid C) &= H(K, g(K, C) \mid C) \\ &= H(K, M|C) \\ &= H(M|C) + H(K|M, C) \\ &\geq H(M|C), \end{aligned}$$

where the inequality follows from the non-negativity of entropy.

c) **(7 points)** Show that $I(M; C) \geq H(M) - H(K)$.
Solution:
We proceed as follows:

$$\begin{aligned} I(M; C) &= H(M) - H(M \mid C) \\ &\overset{(a)}{\geq} H(M) - H(K \mid C) \\ &\overset{(b)}{\geq} H(M) - H(K), \end{aligned}$$

where (a) follows from the result of part (b), and (b) follows from the fact that conditioning reduces entropy.

d) **(6 points) True/False:** A student claims that, in order to achieve perfect secrecy, we must have $H(K) \geq H(M)$. If true, do you think this condition is practical in real-world communication systems? If false, provide a counterexample.
Solution:
True. From part (c), if perfect secrecy holds (i.e., $I(M; C) = 0$), then:

$$0 = I(M; C) \geq H(M) - H(K) \Rightarrow H(K) \geq H(M).$$

This means that the key must have entropy at least as large as the message. Usually the message is very long (i.e., $H(M)$ is large), and we need to transmit/store the key which is as long as the message in a perfectly secure system.
Remark: This necessary condition for perfect secrecy was first established by Claude Shannon in his seminal paper "Communication Theory of Secrecy Systems". It reflects the principle that the key must be at least as "informative" (in entropy) as the message to completely hide it.

e) **(10 points)** Now assume that M, K, and C are all n-bit binary strings, i.e., $M = K = C = \{0, 1\}^n$. Let M and K be independent and uniformly distributed over $\{0, 1\}^n$. Suggest encryption and decryption functions $f(K, M)$ and $g(K, C)$ that achieve perfect secrecy.
Solution:
Consider $f(K, M) = K \oplus M$, $g(K, C) = K \oplus C$, where $\oplus$ denotes the bitwise XOR operation. Clearly $g(K, f(K, M)) = M$. Moreover,

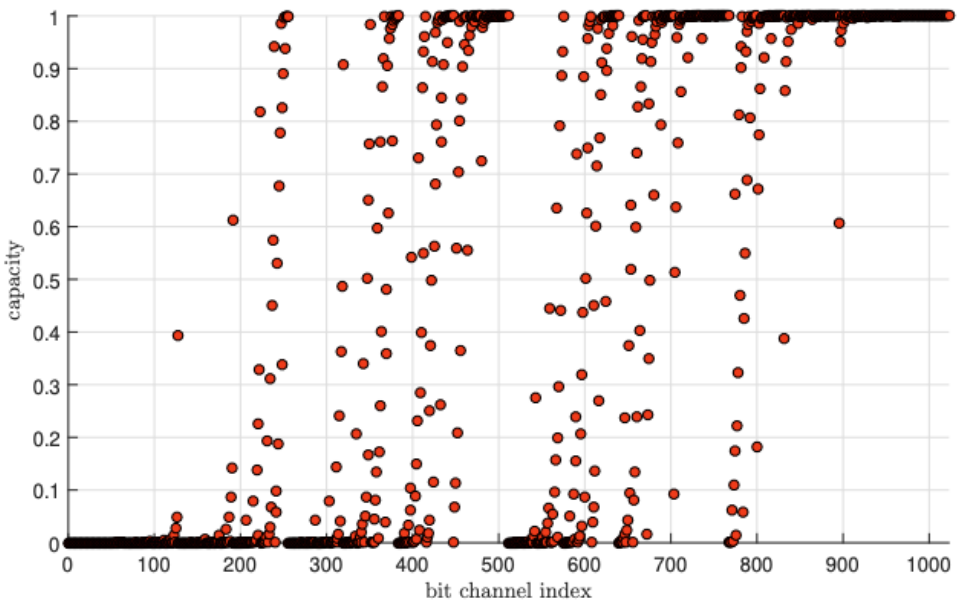
$$\begin{aligned} I(M; C) &= H(C) - H(C \mid M) = H(C) - H(K \oplus M \mid M) \\ &= H(C) - H(K \mid M) \overset{(a)}{=} H(C) - H(K) \overset{(b)}{=} 0, \end{aligned}$$

where step (a) follows from the fact that K and M are independent, and step (b) follows from the fact that both are uniformly distributed over $\{0, 1\}^n$. Hence, this is a perfectly secure system.

3) **Polar Code (35 Points):**

a) **(6 points) True/False** Consider the graph below showing the capacities of the synthetic channels for a polar code of length $N = 1024$. A student claims that using a code rate of 0.8 will result in a block

error probability very close to zero. Justify your answer in one or two sentences.



Solution:
False. From the graph, a code rate of 0.8 means using the 819 most reliable bit-channels, but many of these have capacities significantly below 1, indicating they are unreliable. Including such low-capacity channels will cause a high block error probability, far from zero.

b) **(6 points)** We want to transmit the information bits $(1, 0)$ using a polar code of length $N = 4$ over a binary erasure channel (BEC). Select frozen bits to achieve the best performance, and explain your choice. Then, compute the codeword (X_1, X_2, X_3, X_4).

Solution:
For $R = 0.5$, we need to freeze two bits. we should freeze the bits corresponding to the two least reliable synthetic channels. For a binary erasure channel (BEC) with parameter p, the synthetic channels are given by:

$$W^{--} : \mathrm{BEC}\Big(1 - \big(1 - (1-p)^2\big)^2\Big),$$
$$W^{-+} : \mathrm{BEC}\Big(1 - \big((1-p)^2\big)^2\Big),$$
$$W^{+-} : \mathrm{BEC}\Big(1 - \big(1 - p^2\big)^2\Big),$$
$$W^{++} : \mathrm{BEC}(p^4).$$

Since

$$p^4 \le 1 - (1-p^2)^2 \le 1 - ((1-p)^2)^2 \le 1 - \big(1 - (1-p)^2\big)^2, \quad p \in [0, 1],$$

Thus, the two least reliable channels are W^{--} and W^{-+}, which correspond to bits U_1 and U_2. Therefore, we freeze $u_1 = u_2 = 0$ and place the information bits in u_3 and u_4.
Given the information bits $(1, 0)$, we have: $u = (0, 0, 1, 0)$.
The polar encoding is $x = uG_4$, where $G_4 = F^{\otimes 2}$ and $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. This gives:

$$x_1 = u_1 \oplus u_2 \oplus u_3 \oplus u_4 = 1,$$
$$x_2 = u_2 \oplus u_4 = 0,$$
$$x_3 = u_3 \oplus u_4 = 1,$$
$$x_4 = u_4 = 0.$$

Hence, the codeword is:

$$\boxed{(X_1, X_2, X_3, X_4) = (1, 0, 1, 0)}.$$

c) **(7 points)** Assume the codeword from part (b) is sent over $\mathrm{BEC}(p)$ and the receiver observes $y = (?, ?, 1, 0)$. Preform successive cancellation (SC) decoding and show if the decoder succeeded in decoding the bits.
Remark: You may use the SC decoder functions: $g(r_1, r_2, b) = r_2 + (1 - 2b)r_1$ and $f(r_1, r_2) = \text{sign}(r_1)\text{sign}(r_2)\min(|r_1|, |r_2|)$.
Solution:
First, note that u_1 and u_2 are frozen bits set to 0. Therefore, we only need to decode u_3 and u_4 using the SC algorithm.
The first step is to compute the log-likelihood ratios (LLRs) for the codeword bits x_1^4, denoted as l_1^4, where

$$l_i = \log \frac{P(x_i = 0 \mid y_i)}{P(x_i = 1 \mid y_i)}.$$

For a $\mathrm{BEC}(p)$, the LLR values are 0 for erasures, $+\infty$ for certain zeros, and $-\infty$ for certain ones. Given the received vector $y = (?, ?, 1, 0)$, we have:

$$l_1^4 = (0, 0, -\infty, +\infty).$$

Decoding u_3:
Using the SC update rules:

$$g(r_1, r_2, b) = r_2 + (1 - 2b)r_1, \quad f(r_1, r_2) = \text{sign}(r_1) \cdot \text{sign}(r_2) \cdot \min(|r_1|, |r_2|),$$

and knowing $u_1 = 0$, $u_2 = 0$:
1. Compute

$$g(l_1, l_3, 0) = -\infty, \quad g(l_2, l_4, 0) = +\infty.$$

2. Then

$$l_{u_3} = f(-\infty, +\infty) = -1 \cdot +1 \cdot \min(\infty, \infty) = -\infty.$$

Since $l_{u_3} < 0$, we decide $\hat{u}_3 = 1$.
Decoding u_4:
1. Using $\hat{u}_3 = 1$:

$$l_{u_4} = g(-\infty, +\infty, 1) = +\infty + (1 - 2 \cdot 1)(-\infty) = +\infty - (-\infty) = +\infty.$$

2. Since $l_{u_4} > 0$, we decide $\hat{u}_4 = 0$.
The decoded information bits are $(\hat{u}_3, \hat{u}_4) = (1, 0)$, which matches the transmitted bits. Hence, the SC decoder successfully recovers the message.

d) **(16 points)** We aim to develop an SC-based neural network model, where the goal is to learn the check-node function f_θ and the bit-node function g_θ. Suppose the analytic forms of f and g are unknown, but you are provided with a large dataset of input and log-likelihood ratio (LLR) pairs $D = \{(x_i, l_i)\}_{i=1}^M$, where $x_i \in \{0, 1\}$ are transmitted bits and $l_i \in \mathbb{R}$ are the corresponding LLR obtained after transmission through the channel, i.e., $l_i = \log\left(\frac{P(x_i=0|y_i)}{P(x_i=1|y_i)}\right)$, with y_i denoting the received channel output. The transmitted bits are independent and uniformly distributed, i.e., $P(x_i = 0) = P(x_i = 1) = 0.5$.

i) Describe how to generate a new training dataset for training f_θ and g_θ from $\{(x_i, l_i)\}_{i=1}^M$.
Hint: Recall f takes two LLRs and outputs one LLR; g takes two LLRs and a decoded bit, and outputs one LLR.

ii) Propose a method to learn f_θ and g_θ from the dataset in part (i), specifying the cost function, and provide a block diagram illustrating each model's inputs and outputs.

Solution:

(i) Creating the training datasets. From the original dataset $\mathcal{D} = \{(x_i, l_i)\}_{i=1}^M$, randomly sample two independent elements (x_a, l_a) and (x_b, l_b). Compute the SC node labels:

$$u_1 = x_a \oplus x_b, \qquad u_2 = x_b.$$

Repeating this procedure N times yields two new datasets:

$$\mathcal{D}_f = \{\big((l_a^{(n)}, l_b^{(n)}),\ u_1^{(n)}\big)\}_{n=1}^N,$$
$$\mathcal{D}_g = \{\big((l_a^{(n)}, l_b^{(n)}, u_1^{(n)}),\ u_2^{(n)}\big)\}_{n=1}^N,$$

(ii) Learning f_θ and g_θ.
Approach: Since f and g output LLR values, we can model them as neural networks that output a *logit* s (linear output). The LLR is

$$\hat{p}(y) \approx P(x = 1 \mid y), \quad \widehat{\mathrm{LLR}}_x(y) = \log \frac{1 - \hat{p}(y)}{\hat{p}(y)} = -s.$$

Cost function: Using binary cross-entropy (BCE):

$$\mathrm{Cost}_f = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{\mathrm{BCE}}\big(\sigma(f_\theta(l_a^{(n)}, l_b^{(n)})), u_1^{(n)}\big),$$

$$\mathrm{Cost}_g = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{\mathrm{BCE}}\big(\sigma(g_\phi(l_a^{(n)}, l_b^{(n)}, u_1^{(n)})), u_2^{(n)}\big).$$

where σ is sigmoid.
Model LLR outputs:

$$\widehat{L}_{u_1} = -f_\theta(l_a, l_b), \quad \widehat{L}_{u_2} = -g_\phi(l_a, l_b, u_1).$$

Block diagrams:

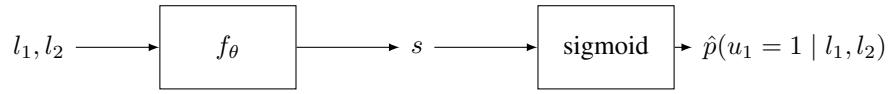


Fig. 1: Check-node NN model

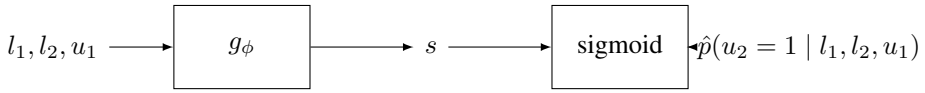


Fig. 2: bit-node NN model

Good Luck!