Introduction to Information Theory and Machine Learning (Prof. Permuter Haim, Mr. Bashar Huleihel, and Mr. Omer Luxembourg)

July 7th, 2022.

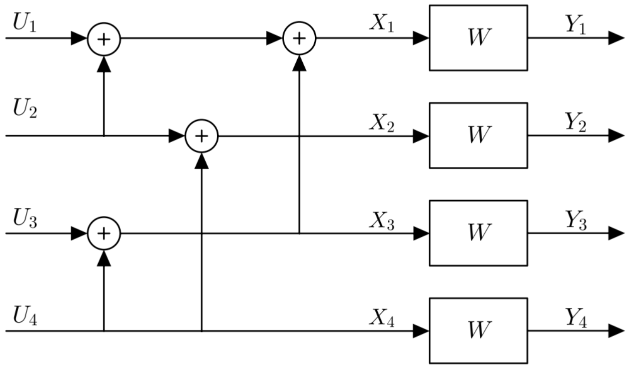
Final Exam - Moed A

Total time for the exam: 3 hours!

Please copy the following sentence and sign it: " I am respecting the rules of the exam: Signature:_________ "

Important: For **True / False** questions, copy the statement to your notebook and write clearly true or false. You should prove the statement if true, or disprove it, e.g. by providing a counter-example, otherwise.

1) **Polar code design (50 Points):** In this problem, we explore how the selection of frozen bits affects the decoding process in the construction of polar codes. Consider a binary erasure channel (BEC) W, and the encoding scheme of block size $N = 4$ in the figure below.



For items (a) to (c), assume that the bits U_1 and U_2 are frozen to 0, while U_3 and U_4 are used for message bits.

a) **(4 points)** Calculate the code rate.
Solution:
The code rate R is given by the ratio of the number of message bits to the total number of bits in the codeword. Accordingly $R = 0.5$.

b) **(4 points)** Given the message bits $(U_3, U_4) = (1, 1)$, perform the encoding to determine the codeword (X_1, X_2, X_3, X_4).
Solution:
The codeword $X^4 = (X_1, X_2, X_3, X_4)$ is given by

$$\begin{aligned} X^4 &= U^4 \cdot G_4 \\ &= \begin{bmatrix} 0, 0, 1, 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0, 1, 0, 1 \end{bmatrix}. \end{aligned}$$

c) **(10 points)** Apply successive cancellation decoding to the received vector $(Y_1, Y_2, Y_3, Y_4) = (1, ?, ?, 1)$. Determine if decoding is successful, and if so, identify the decoded message.
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 successive cancellation (SC) algorithm. The initial step involves computing the log-likelihood ratio (LLR) for the bits $\{x_i\}_{i=1}^4$ denoted as $\{l_{x_i}\}_{i=1}^4$, where $l_{x_i} = \log_2 \left(\frac{P(x_i=0|y_i)}{P(x_i=1|y_i)} \right)$. Using the channel and the received vector $(1, ?, ?, 1)$, we find

$$(l_{x_1}, l_{x_2}, l_{x_3}, l_{x_4}) = (-\infty, 0, 0, -\infty). \tag{1}$$

With the SC algorithm, the LLR for the bit U_3 is calculated as follows:

$$\begin{aligned} l_{u_3} &= f\left(g(l_{x_1}, l_{x_3}, 0), g(l_{x_2}, l_{x_4}, 0)\right) \\ &= \infty. \end{aligned} \tag{2}$$

Since $l_{u_3} > 0$, the decoder sets $\hat{u}_3 = 0$. After decoding u_3, the SC algorithm proceed to decode u_4:

$$\begin{aligned} l_{u_4} &= g\left(g(l_{x_1}, l_{x_3}, 0), g(l_{x_2}, l_{x_4}, 0), \hat{u}_3\right) \\ &= g\left(g(l_{x_1}, l_{x_3}, 0), g(l_{x_2}, l_{x_4}, 0), 0\right) \\ &= -\infty. \end{aligned} \tag{3}$$

Since $l_{u_4} < 0$, the decoder sets $\hat{u}_4 = 1$. Thus, the decoded message $(U_3, U_4) = (0, 1)$.

For items (d) to (f), consider a different scenario where U_2 and U_3 are frozen to 0, while U_1 and U_4 are message bits.

d) **(10 points)** Apply successive cancellation decoding for received vector $(Y_1, Y_2, Y_3, Y_4) = (1, ?, ?, 0)$, and show that the decoding might fail when decoding U_1.
Solution:
Let us compute the LLR for the bit u_1, which in this case is not frozen:

$$\begin{aligned} l_{u_1} &= f\left(f(l_{x_1}, l_{x_3}), f(l_{x_2}, l_{x_4})\right) \\ &= 0. \end{aligned}$$

Since the SC algorithm uses a deterministic decision rule, which checks if the LLR is strictly greater or less than 0, it fails to decode u_1 because $l_{u_1} = 0$ does not provide a clear decision.

e) **(5 points)** Explain the reason for the decoding failure.
Solution:
The decoding failure is primarily due to an improper selection of frozen bits. In polar coding, the effectiveness of decoding, particularly using the SC algorithm, depends significantly on the choice of which bits to freeze. The LLR for the bit u_1 was computed as $l_{u_1} = 0$, indicating that the received information does not provide a clear preference for either bit value (0 or 1). This lack of decisiveness suggests that the synthetic channel associated with $u_1$is not reliable enough to make a confident decision. In an ideal polar coding scheme, frozen bits should be set on the "bad" synthetic channels (those with high error probabilities), ensuring that the "good" channels (with lower error probabilities) are used for transmitting actual information. However, in this scenario, the frozen bits were not appropriately selected, leading to u_1, associated with a "bad" channel, being left unfrozen.

f) **(10 points)** Perform optimal maximum likelihood decoding for the same received vector $(1, ?, ?, 0)$. To do this:
- First, compute the codeword (X_1, X_2, X_3, X_4) for all four possible input messages (U_1, U_4).
- Then, identify the input message(s) (U_1, U_4) that could result in the received vector $(1, ?, ?, 0)$. If more than one message corresponds to the received vector, declare a failure in decoding.

Determine if the decoding is successful, and if so, specify the decoded message.
Solution:
The four possible combinations for (U_1, U_4) are $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$. The corresponding codewords (X_1, X_2, X_3, X_4) or these combinations are:

$$\begin{aligned} (U_1, U_4) = (0, 0) &\rightarrow (X_1, X_2, X_3, X_4) = (0, 0, 0, 0) \\ (U_1, U_4) = (0, 1) &\rightarrow (X_1, X_2, X_3, X_4) = (1, 1, 1, 1) \\ (U_1, U_4) = (1, 0) &\rightarrow (X_1, X_2, X_3, X_4) = (1, 0, 0, 0) \\ (U_1, U_4) = (1, 1) &\rightarrow (X_1, X_2, X_3, X_4) = (0, 0, 0, 1). \end{aligned}$$

Clearly, the only codeword that matches the received vector $(1, ?, ?, 0)$ is $(1, 0, 0, 0)$, which corresponds to the input vector $(U_1, U_4) = (1, 0)$. Accordingly, maximum likelihood decoding is successful in this case.

g) **(7 points)** For a code rate $R = 0.5$, determine which bits should be set as frozen to achieve the best performance in terms of bit error rate when using the successive cancellation decoding algorithm. Justify your selection.
Solution:
For $R = 0.5$, we need to freeze two bits. As previously discussed, 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:

$$\begin{aligned} &W^{--} : BEC(1 - (1 - (1 - (1 - p)^2)^2) \\ &W^{-+} : BEC(1 - ((1 - p)^2)^2) \\ &W^{+-} : BEC(1 - (1 - p^2)^2) \\ &W^{++} : BEC(p^4). \end{aligned}$$

It can be verified that the two least reliable channels are W^{--} and W^{-+}, which correspond to the bits U_1 and U_2. Therefore, these bits should be frozen.

2) **Combine linear regression with MINE (50 Points):** In this question, we will create and analyze a model for weather prediction. The task is as follows: given sequences of size N denoted as $X^N = (x_1, x_2, \ldots, x_N)$ where x_i is the temperature on the i-th

day we would like to predict the next day x_{N+1}. To solve this problem, the following model was suggested:

$$x_{i+1} = \sum_{j=0}^{D} w_j \cos\left(\frac{2\pi j}{365} \cdot i\right) + \sum_{j=0}^{L} \alpha_j x_{i-j} + b_{i+1}, \tag{-3}$$

where D, L are constants, and b_i is an i.i.d random variable.

a) **Model Analysis**

i) **(7 points)** In class, we learned that a linear regression problem can be written as

$$Y = \theta^T \cdot X. \tag{-2}$$

Can you build a linear estimator for $\hat{x}_{i+1}$ based on previous samples using linear regression? If yes, provide Y, θ, and X. If not, explain why.

Solution:

Yes,

$$\theta = (w_0, w_1 \ldots, w_D, \alpha_0, \alpha_1 \ldots \alpha_L)$$

$$X = (\cos\left(\frac{2\pi 0}{365} \cdot i\right), \cos\left(\frac{2\pi 1}{365} \cdot i\right), \ldots \cos\left(\frac{2\pi D}{365} \cdot i\right), x_i, x_{i-1}, \ldots x_{i-L})$$

$$Y = \theta^T \cdot X$$

ii) **(4 points)** The model in Eq (1) assume some assumptions, explain what are the assumptions.

Solution:

The model has two main assumptions. The first assumption is that the data (the weather) is periodic with a period of 365 days. The second assumption is that the weather/temperature is linearly affected by the previous temperatures.

iii) **(13 points)** Write the gradient descent formula for updating the weights using the MSE cost function. **Solution:**

$$C = \sum_{x \in \text{data}} (y - \hat{y}(x))^2$$

$$\theta \leftarrow \theta - \eta \nabla C = (\frac{dC}{dw_0}, \frac{dC}{dw_1} \cdots \frac{dC}{dw_D}, \frac{dC}{d\alpha_i} \cdots \frac{dC}{d\alpha_L})$$

$$\frac{dC}{dw_j} = \frac{dC}{d\hat{y}} \cdot \frac{d\hat{y}}{dw_j} = \sum_{x_i \in \text{data}} -2(y - \hat{y}(x)) cos(\frac{2\pi j}{365} i)$$

$$\frac{dC}{d\alpha_j} = \frac{dC}{d\hat{y}} \cdot \frac{d\hat{y}}{d\alpha_j} = \sum_{x_i \in \text{data}} -2(y - \hat{y}(x)) x_{i-j}$$

iv) **(9 points)** The researcher decided to change the cost function to the following:

$$\text{cost} = \sum_{x \in \text{data}} (y - \hat{y}(x))^2 + \lambda \sum_{i=1}^{M} \theta_i^2 - 2\theta_i.$$

Where $\theta = (\theta_1, \theta_2 \ldots, \theta_M)$ is the vector defined in Eq (2). How would we expect the model and model performance to change under the following λ:

- $\lambda \to \infty$
- λ is a positive finite number

Remember, when first initializing the model, the weights are assigned at random.

Solution:

We can see that the current cost function is an L2 variation. For a larger λ, the model will prioritize minimizing the second term

$$\lambda \sum_{i=1}^{M} \theta_i^2 - 2\theta_i$$

more and more, which means taking the weights closer to the values of 1. For a large λ, the model will set the weights to be close to 1, and for a finite λ the model will most likly move the weights to the range (0,2).

b) **Probability Analysis** Let $z_{i+1} = x_{i+1} - \hat{x}_{i+1}$ be the error process with $P_z : [-50, 50] \to [0, 1]$ with i.i.d distribution. Let $Q_z \sim U[-50, 50]$ represent a uniformly distributed random variable.

i) **(5 points)** Express the KL divergence $D_{KL}(P_z, Q_z)$ in terms of the entropy of $Z \sim P_z$.

Solution:

$$D_{KL}(P_z \| Q_z) = \int_{-50}^{50} P_z(z) \log \frac{P_z(z)}{Q_z(z)}\, dz.$$

Since Q_z is uniformly distributed over $[-50, 50]$, we have:

$$Q_z(z) = \frac{1}{100}, \quad \text{for all } z \in [-50, 50].$$

Thus, the KL divergence becomes:

$$\begin{aligned} D_{KL}(P_z \| Q_z) &= \int_{-50}^{50} P_z(z) \log\left(P_z(z) \cdot 100\right)\, dz \\ &= \int_{-50}^{50} P_z(z) \log P_z(z)\, dz + \int_{-50}^{50} P_z(z) \log 100\, dz \\ &\overset{(a)}{=} \int_{-50}^{50} P_z(z) \log P_z(z)\, dz + \log 100 \\ &\overset{(b)}{=} -H(P_z) + \log 100, \end{aligned}$$

where (a) follows by

$$\int_{-50}^{50} P_z(z) \log 100\, dz = \log 100 \int_{-50}^{50} P_z(z)\, dz = \log 100,$$

and (b) follows by the definition of relative entropy.

We sampled a large number of samples $x_1, \dots, x_M$. In the class we learned the Donsker-Varadhan representation

$$D_{KL}(P \parallel Q) = \sup_{T:\Omega\to\mathbb{R}} \left(\mathbb{E}_P[T] - \log(\mathbb{E}_Q[e^T])\right). \tag{-1}$$

ii) **(6 points)** Offer a method to estimate the entropy of the process z_i.

Solution:

Using subsection b.i, we get that $D_{KL}(P_z \parallel Q_z) = -H(P_z) + \log 100$, which means that in order to estimate $H(P_z)$, we can estimate $D_{KL}(P_z \parallel Q_z)$. Using the Donsker-Varadhan representation, we get the following equation:

$$D_{KL}(P \parallel Q) = \sup_{T:\Omega\to\mathbb{R}} \left(\mathbb{E}_P[T] - \log(\mathbb{E}_Q[e^T])\right)$$

We replace T with a neural network to estimate the supremum. First, we will take the data $(x_1, \dots, x_n)$ and calculate $(z_1, \dots, z_n)$, which provide us with samples from P_z. Then, we will sample out of the range $(-50, 50)$ uniformly to get samples $\bar{z}_1, \dots, \bar{z}_M$, which correspond to samples from Q_z. Combining everything and using the law of large numbers to estimate expectations, we get:

$$D_{KL}(P \parallel Q) \approx \sup_{T:\Omega\to\mathbb{R}} \left(\frac{1}{M}\sum_{i=1}^{M} T(z_i) - \log\left(\frac{1}{M}\sum_{i=1}^{M} e^{T(\bar{z}_i)}\right)\right).$$

Therefore:

$$H(P_z) \approx \log(100) - \sup_{T:\Omega\to\mathbb{R}} \left(\frac{1}{M}\sum_{i=1}^{M} T(z_i) - \log\left(\frac{1}{M}\sum_{i=1}^{M} e^{T(\bar{z}_i)}\right)\right).$$

iii) **(6 points)** Offer a method to estimate the probability density of the process z_i .

Solution:

Given the previous prediction, we end up with the optimized network T_θ. As learned in class, $T_\theta^*(x) = \log\left(\frac{P_z}{Q_z}\right)$, which means we can deduce that $P_z(x) = e^{T_\theta^*(x)} Q_z(x) = \frac{1}{100} e^{T_\theta^*(x)}$.

Good Luck!