ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 12 Information Theory and Coding
Solutions to Homework 5 Oct. 22, 2018

PROBLEM 1.

(a) It is easy to check that W is an i.i.d. process but Z is not. As W is i.i.d. it is also stationary. We want to show that Z is also stationary. To show this, it is sufficient to prove that the distribution of the process does not change by shift in the time domain.

$$\begin{aligned}
&p_Z(Z_m = a_m, Z_{m+1} = a_{m+1}, \cdots, Z_{m+r} = a_{m+r}) \\
&= \frac{1}{2} p_X(X_m = a_m, X_{m+1} = a_{m+1}, \cdots, X_{m+r} = a_{m+r}) \\
&+ \frac{1}{2} p_Y(Y_m = a_m, Y_{m+1} = a_{m+1}, \cdots, Y_{m+r} = a_{m+r}) \\
&= \frac{1}{2} p_X(X_{m+s} = a_m, X_{m+s+1} = a_{m+1}, \cdots, X_{m+s+r} = a_{m+r}) \\
&+ \frac{1}{2} p_Y(Y_{m+s} = a_m, Y_{m+s+1} = a_{m+1}, \cdots, Y_{m+s+r} = a_{m+r}) \\
&= p_Z(Z_{m+s} = a_m, Z_{m+s+1} = a_{m+1}, \cdots, Z_{m+s+r} = a_{m+r}),
\end{aligned}$$

where we used the stationarity of the X and Y processes. This shows the invariance of the distribution with respect to the arbitrary shift s in time which implies stationarity.

(b) For the Z process we have

$$\begin{aligned}
H(Z) &= \lim_{n\to\infty} \frac{1}{n} H(Z_1, \cdots, Z_n) \\
&= \lim_{n\to\infty} \frac{1}{n} H(Z_1, \cdots, Z_n \mid \Theta) \\
&= \frac{1}{2} H(X_0) + \frac{1}{2} H(Y_0) = 1.
\end{aligned}$$

W process is an i.i.d process with the distribution $p_W(a) = \frac{1}{2}p_X(a) + \frac{1}{2}p_Y(a)$. From concavity of the entropy, it is easy to see that $H(W) = H(W_0) \geq \frac{1}{2}H(X_0) + \frac{1}{2}H(Y_0) = 1$. Hence, the entropy rate of W is greater than the entropy rate of Z and the equality holds if and only if X_0 and Y_0 have the same probability distribution function.

PROBLEM 2.

(a) We have $\rho(X_1^\infty) = 0$. We show this by showing that $\rho(X_1^\infty) \leq \delta$ for any $\delta > 0$. To see the last statement, build an invertible FSM which "recognizes" a string of type "ab...ab" for a particular even length, call it L, and outputs lets say "0" at the end of this string and returns to the starting state. Hence this machine will output an infinite string of "0" when the input is X_1^∞. From each state (including the starting state) of the chain which recognizes the special string make an edge back to the starting state in the case the next input is not the correct one. The output for each such edge is $1 + \lceil \log L \rceil$ bits long, the first bit is 1 to indicate that it is not the special path and on the next $\lceil \log L \rceil$ bits we give the index of the state (in binary representation) from which the return edge is drawn. This machine is clearly lossless and has a compressibility of $1/L$ for the desired sequence.

(b) A machine as described above will have $\rho_M(X_1^\infty) = 1/4$. In fact, one cannot do better than this. Consider a cycle, when from a given state we get back to the same state. During such a cycle we have to output at least one symbol, because the machine has to be information lossless. In an L state machine we eventually create such a cycle within at most L steps. This means that we output at least one symbol for every L input symbols, so $\rho_M(X_1^\infty) \geq 1/L$.

(c) We have $\rho_{\mathrm{LZ}} = 0$ since compressibility is non-negative and we know that the compressibility of LZ is at least as good as that of any FSM, i.e., we know that $\rho_{\mathrm{LZ}}(X_1^\infty) \leq \rho(X_1^\infty)$.

(d) The dictionary increases by 1 every time and has size 2 in the beginning. Hence, if we look at lets say c steps of the algorithm then we need in total

$$\sum_{i=1}^{c} \lceil \log(1+i) \rceil \leq c \log(2(c+1))$$

bits to describe the output.

What are the words which we are using. Note that the parsing is a, b, ab, aba, ba, bab,... Note that in average at most every second step the length of the used dictionary word increases by 1, i.e., we have a linear increase in the used dictionary words. Therefore, if we compute the total length which we have parsed after c steps, this length increases like the square of c.

It follows that the ratio of the total number of bits used divided by the total length described behaves like $1/c$, i.e., it tends to 0.

Problem 3.

(a) Let $p_i = \frac{n_i}{n}$. Then

$$\begin{aligned}
1 = (p_1 + p_2 + \cdots + p_K)^n &\overset{(a)}{=} \sum_{\substack{n_1, n_2, \ldots, n_K \\ \text{s.t. } \sum n_i = n}} \binom{n}{n_1 n_2 \ldots n_K} p_1^{n_1} p_2^{n_2} \ldots p_K^{n_K} \\
&\geq \binom{n}{n_1 n_2 \ldots n_K} p_1^{n_1} p_2^{n_2} \ldots p_K^{n_K} \\
&\geq \binom{n}{n_1 n_2 \ldots n_K} 2^{n_1 \log(p_1) + n_2 \log(p_2) + \cdots + n_K \log(p_K)} \\
&\geq \binom{n}{n_1 n_2 \ldots n_K} 2^{-n \mathrm{h}(p_1, p_2, \ldots, p_K)},
\end{aligned}$$

where (a) is the binomial expansion. This proves our claim.

(b) For a random sequence of length n, $U_1 U_2 \ldots U_n$, we encode the number of occurrences of the first $(K-1)$ letters, denoted $N_1, N_2, \ldots, N_{K-1}$, since we get the last letter for free ($N_K = n - N_1 - N_2 - \cdots - N_{K-1}$). For each letter we need at most $\lceil \log(n+1) \rceil$ bits. Now that we know the number of times each letter appeared in the sequence we need to encode the index of this specific sequence among all sequences having the same numbers of letter occurrences $(N_1, \ldots, N_{K-1})$. Since there are $\binom{n}{N_1 N_2 \ldots N_K}$ of those sequences then we need at most $\left\lceil \log\left(\binom{n}{N_1 N_2 \ldots N_K}\right) \right\rceil$ bits.

Hence the total length L of the codeword is

$$L = (K-1)\lceil \log(n+1) \rceil + \left\lceil \log\left(\binom{n}{N_1 N_2 \dots N_K}\right)\right\rceil.$$

The expected length is

$$\begin{aligned}
\mathbb{E}(L) &= (K-1)\lceil \log(n+1) \rceil + \mathbb{E}\left(\left\lceil \log\left(\binom{n}{N_1 N_2 \dots N_K}\right)\right\rceil\right) \\
&\leq (K-1)\left(\log(n+1)+1\right) + \mathbb{E}\left(\log\left(\binom{n}{N_1 N_2 \dots N_K}\right)\right) + 1 \\
&\overset{(a)}{\leq} (K-1)\log(n+1) + K + \mathbb{E}\left(n\mathrm{h}\left(\frac{N_1}{n}, \frac{N_2}{n}, \dots, \frac{N_K}{n}\right)\right) \\
&\overset{(b)}{\leq} (K-1)\log(n+1) + K + n\mathrm{h}\left(\mathbb{E}\left(\frac{N_1}{n}, \frac{N_2}{n}, \dots, \frac{N_K}{n}\right)\right)
\end{aligned}$$

where (a) is due to the first part of the exercise and (b) is due to Jensen's inequality.

For a random sequence $U_1 U_2 \dots U_n$, the number of occurrences of a particular letter u_i is a random variables that can be written as the sum of indicator functions which take the value 1 with probability q_i

$$N_i = \sum_{j=1}^{n} 1_{\{U_j = u_i\}}.$$

Hence

$$\mathbb{E}\left(\frac{N_i}{n}\right) = \frac{\sum_{j=1}^{n} \mathbb{E}\left(1_{\{U_j=u_i\}}\right)}{n} = q_i.$$

Therefore, the expected codeword length per letter is

$$\frac{1}{n}\mathbb{E}(L) \leq (K-1)\frac{\log(n+1)}{n} + \frac{K}{n} + \mathrm{h}\left(q_1, q_2, \dots, q_K\right).$$

This shows that $\lim_{n\to\infty} \frac{1}{n}\mathbb{E}(L) \leq \mathrm{h}(q_1, q_2, \dots, q_K) = \mathrm{H}(U)$. Since the source is i.i.d then $\mathrm{H}(U) = \lim_{n\to\infty} \mathrm{H}(U_1 U_2 \dots U_n)$. This proves the optimality of this compression code for i.i.d sources.

(c) If the source is not i.i.d then $\mathrm{H}(U) \neq \lim_{n\to\infty} \mathrm{H}(U_1 U_2 \dots U_n)$. Hence, the code is not necessarily optimal.

Problem 4. Since given X, one can determine Y from Z and vice versa, $H(Y|X) = H(Z|X) = H(Z) = \log 3$, regardless of the distribution of X. Hence the capacity of the channel is

$$\begin{aligned}
C &= \max_{p_X} I(X;Y) \\
&= \max_{p_X} H(Y) - H(Y|X) \\
&= \log 11 - \log 3
\end{aligned}$$

which is attained when X has uniform distribution. The same result can also be seen by observing that this channel is symmetric.

Problem 5. Denote the capacity achieving distributions for Channel 1 and 2 as $p^*_{X_1}(x_1)$ on $\mathcal{X}_1$ and $p^*_{X_2}(x_2)$ on $\mathcal{X}_2$ respectively. This means

$$I(X_1;Y_1)\big|_{p^*_{X_1}(x_1)} = C_1 \text{ and } I(X_2;Y_2)\big|_{p^*_{X_2}(x_2)} = C_2$$

With time sharing (or flipping a coin Λ with λ probability of selecting Channel 1), we can use Channel 1 for λ fraction of the time and Channel 2 for $\bar{\lambda} = 1 - \lambda$ fraction of the time. So this new random variable can be described as

$$p_\Lambda(k) = \begin{cases} \lambda \text{ if } k = 1 \\ \bar{\lambda} \text{ if } k = 2 \end{cases}$$

The joint probability distribution of Λ, X, Y can now be written as the following.

$$p_{\Lambda,X,Y}(k,x,y) = p_\Lambda(k)p_{X|\Lambda}(x|k)p_{Y|X,\Lambda}(y|x,\lambda)$$

For any choice of input distribution on X_1 and X_2, the marginal distribution of Y is expressed as

$$\begin{aligned}
p_Y(y) &= \sum_{x\in\mathcal{X},k\in\{1,2\}} p_{\Lambda,X,Y}(k,x,y) = \sum_{x\in\mathcal{X},k\in\{1,2\}} p_\Lambda(k)p_{X|\Lambda}(x|k)p_{Y|X,\Lambda}(y|x,\lambda) \\
&= \sum_{x\in\mathcal{X}} p_\Lambda(1)p_{X|\Lambda}(x|1)p_{Y|X,\Lambda}(y|x,1) + \sum_{x\in\mathcal{X}} p_\Lambda(2)p_{X|\Lambda}(x|2)p_{Y|X}(y|x,2) \\
&= \sum_{x\in\mathcal{X}} \lambda p_{X|\Lambda}(x|1)p_{Y|X,\Lambda}(y|x,1) + \sum_{x\in\mathcal{X}} \bar{\lambda} p_{X|\Lambda}(x|2)p_{Y|X,\Lambda}(y|x,1) \\
&= \sum_{x_1\in\mathcal{X}_1} \lambda p_{X_1}(x_1)p_{Y_1|X_1}(y|x_1) + \sum_{x_2\in\mathcal{X}_2} \bar{\lambda} p_{X_2}(x_2)p_{Y_2|X_2}(y|x_2) \\
&= \lambda p_{Y_1}(y) + \bar{\lambda} p_{Y_2}(y)
\end{aligned}$$

To calculate $H(Y)$, the following steps can be used.

$$\begin{aligned}
H(Y) &= -\sum_{y\in\mathcal{Y}} p_Y(y)\log p_Y(y) \\
&= -\sum_{y\in\mathcal{Y}_1} \lambda p_{Y_1}(y)\log \lambda p_{Y_1}(y) - \sum_{y\in\mathcal{Y}_2} \bar{\lambda} p_{Y_2}(y)\log \bar{\lambda} p_{Y_2}(y) \\
&= -\lambda\log\lambda - \bar{\lambda}\log\bar{\lambda} - \lambda\sum_{y\in\mathcal{Y}_1} p_{Y_1}(y)\log p_{Y_1}(y) - \bar{\lambda}\sum_{y\in\mathcal{Y}_2} p_{Y_2}(y)\log p_{Y_2}(y) \\
&= h_2(\lambda) + \lambda H(Y_1) + \bar{\lambda} H(Y_2)
\end{aligned}$$

Here, $h_2(\lambda)$ is the binary entropy function as $h_2(\lambda) = -\lambda\log\lambda - \bar{\lambda}\log\bar{\lambda}$. Similarly, to calculate $H(Y|X)$, observe the following steps.

$$
\begin{aligned}
H(Y|X) &= -\sum_{y\in\mathcal{Y},x\in\mathcal{X}} p_{Y,X}(y,x)\log p_{Y|X}(y|x) \\
&= -\sum_{y\in\mathcal{Y},x\in\mathcal{X}} p_{Y|X}(y|x)p_X(x)\log p_{Y|X}(y|x) \\
&= -\sum_{y\in\mathcal{Y},x\in\mathcal{X}} p_{Y|X}(y|x)(\lambda p_{X_1}(x)+\bar{\lambda}p_{X_2}(x))\log p_{Y|X}(y|x) \\
&= -\lambda\sum_{y\in\mathcal{Y}_1,x\in\mathcal{X}_1} p_{Y_1|X_1}(y_1|x_1)p_{X_1}(x_1)\log p_{Y_1|X_1}(y_1|x_1) \\
&\quad -\bar{\lambda}\sum_{y\in\mathcal{Y}_2,x\in\mathcal{X}_2} p_{Y_2|X_2}(y_2|x_2)p_{X_2}(x_2)\log p_{Y_2|X_2}(y_2|x_2) \\
&= \lambda H(Y_1|X_1)+\bar{\lambda}H(Y_2|X_2)
\end{aligned}
$$

This shows that

$$
\begin{aligned}
I(X;Y) &= H(Y)-H(Y|X) = h_2(\lambda)+\lambda(H(Y_1)-H(Y_1|X_1))+\bar{\lambda}(H(Y_2)-H(Y_2|X_2)) \\
&= h_2(\lambda)+\lambda I(X_1;Y_1)+\bar{\lambda}I(X_2,Y_2)
\end{aligned}
$$

Again, for this generic expression which is valid for any input distribution on X_1 and X_2, we can find the optimal λ by taking the derivative and setting to 0 as $I(X;Y)$ is concave with respect to λ.

$$
\begin{aligned}
\frac{\partial I(X;Y)}{\partial\lambda}\Big|_{\lambda^*} &= \log\left(\frac{1-\lambda^*}{\lambda^*}\right)+I(X_1;Y_1)-I(X_2;Y_2)=0 \\
\lambda^* &= \frac{1}{1+2^{I(X_2;Y_2)-I(X_1;Y_1)}}
\end{aligned}
$$

The substitution of λ^* in $I(X;Y)$ gives us

$$
I(X;Y)\big|_{\lambda^*} = \log_2(2^{I(X_1;Y_1)}+2^{I(X_2;Y_2)}).
$$

Note that this generic expression can be maximized using the capacity achieving distributions as $p^*_{X_1}(x_1)$ and $p^*_{X_2}(x_2)$ both maximize $I(X_1;Y_1)$ and $I(X_2;Y_2)$ independently. Therefore

$$
C = \log_2(2^{C_1}+2^{C_2}).
$$