Information Theory and Coding
Prof. Suhas Diggavi

EPFL Winter Semester 2009/2010
Handout # 14, Wednesday, 28 October, 2009

Solutions: Homework Set # 4

Problem 1

It's easy to design an optimal code for each state using Huffman procedure. A possible solution is:

U_n U_{n-1}	S_1	S_2	S_3
S_1	0	10	11
S_2	10	0	11
S_3	–	0	1

(1)

and so $\mathbb{E}(L|C_1) = 1.5$ bits per symbol, $\mathbb{E}(L|C_2) = 1.5$ bits per symbol, and $\mathbb{E}(L|C_3) = 1$ bit per symbol. The average message lengths of the next symbol conditioned on the previous state being S_i are just the expected lengths of the codes C_i. Note that this code assignment achieves the conditional entropy lower bound.

To find the unconditional average, we have to find the stationary distribution on the states. Let μ be the stationary distribution:

$$\mu = \mu \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

μ is found to be $\mu = [\frac{2}{9}, \frac{4}{9}, \frac{1}{3}]$. Thus the unconditional average number of bits per source symbol is

$$\begin{aligned} \mathbb{E}L &= \sum_{i=1}^{3} \mu_i \mathbb{E}(L|C_i) \\ &= \frac{2}{9} \times 1.5 + \frac{4}{9} \times 1.5 + \frac{1}{3} \times 1 \\ &= \frac{4}{3} \text{ bits/symbol.} \end{aligned}$$

The entropy rate of the Markov chain is

$$\begin{aligned} \mathcal{H} &= H(X_2|X_1) \\ &= \sum_i \mu_i H(X_2|X_1 = S_i) \\ &= \frac{4}{3} \text{ bits/symbol} \end{aligned}$$

Thus the unconditional average number of bits per source symbol and the entropy rate $\mathcal{H}$ of the Markov chain are equal, because the expected length of each code C_i equals the entropy of the state after state i, $H(X_2|X_1 = S_i)$, and the maximal compression is obtained.

Problem 2

(a) We can write the following chain of inequalities

$$\begin{aligned}
Q^n(\mathbf{x}) &\overset{1}{=} \prod_{i=1}^{n} Q(x_i) \\
&\overset{2}{=} \prod_{a\in\mathcal{X}} Q(a)^{N(a|\mathbf{x})} \\
&\overset{3}{=} \prod_{a\in\mathcal{X}} Q(a)^{nP_{\mathbf{x}}(a)} \\
&\overset{4}{=} \prod_{a\in\mathcal{X}} 2^{nP_{\mathbf{x}}(a)\log Q(a)} \\
&= \prod_{a\in\mathcal{X}} 2^{n(P_{\mathbf{x}}(a)\log Q(a)-P_{\mathbf{x}}(a)\log P_{\mathbf{x}}(a)+P_{\mathbf{x}}(a)\log P_{\mathbf{x}}(a))} \\
&= 2^{n\sum_{a\in\mathcal{X}}(-P_{\mathbf{x}}(a)\log\frac{P_{\mathbf{x}}(a)}{Q(a)}+P_{\mathbf{x}}\log P_{\mathbf{x}})} \\
&= 2^{n(-D(P_{\mathbf{x}}\|Q)-H(P_{\mathbf{x}}))}
\end{aligned}$$

where 1 follows because of X_i's being i.i.d.,2 follows by grouping symbols and 4 is just by definition of type.

(b) In this part, we show that

$$|T(P)| \doteq 2^{nH(P)}$$

for binary alphabet. This means that we have to show

$$\binom{n}{k} \doteq 2^{nH(\frac{k}{n})}.$$

Note that we say that $a_n \doteq b_n$ if

$$\lim_{n\to\infty}\frac{1}{n}\log\frac{a_n}{b_n}=0,$$

and thus it is enough to bound $\binom{n}{k}$ by

$$\frac{1}{n+1}2^{nH(\frac{k}{n})} \le \binom{n}{k} \le 2^{nH(\frac{k}{n})}.$$

Upper bound:
We know that $\sum_{k=0}^{n}\binom{n}{k}p^k(1-p)^{n-k}=1$. Thus considering only one term, and setting $p=\frac{k}{n}$ gives

$$\begin{aligned}
1 &\ge \binom{n}{k}\left(\frac{k}{n}\right)^k\left(1-\frac{k}{n}\right)^{n-k} \\
&= \binom{n}{k}2^{\log\left(\frac{k}{n}\right)^k+\log\left(\frac{n-k}{n}\right)^{n-k}} \\
&= \binom{n}{k}2^{n(\frac{k}{n}\log\frac{k}{n}+\frac{n-k}{n}\log\frac{n-k}{n})} \\
&= \binom{n}{k}2^{-nH(\frac{k}{n})}
\end{aligned}$$

where $H(p) = p\log p + (1-p)\log(1-p)$. So

$$\binom{n}{k} \leq 2^{nH(\frac{k}{n})}$$

Lower bound:

$$\begin{aligned}
1 &= \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \\
&\leq (n+1)\max_k \binom{n}{k} p^k (1-p)^{n-k} \\
&\overset{1}{=} (n+1)\binom{n}{np} p^{np}(1-p)^{n-np}.
\end{aligned}$$

where 1 is obtained as follows:
Define $S_i = \binom{n}{i}p^i(1-p)^{n-i}$. We want to show that the i maximizing S_i is np. To this end, let's calculate $\frac{S_{i+1}}{S_i}$ for $i < np$ and $i > np$. One could verify $\frac{S_{i+1}}{S_i} = \frac{n-i}{i+1}\frac{p}{1-p}$. Now see that for $i = np-1$, $\frac{S_{i+1}}{S_i} > 1$ and for $i > np$, $\frac{S_{i+1}}{S_i} < 1$. This says that S_i is increasing untill $i = np$ and decreasing afterwards. so the maximum happens at $i = np$.
So letting $p = \frac{k}{n}$,

$$1 \leq (n+1)\binom{n}{k}\left(\frac{k}{n}\right)^k \left(1-\frac{k}{n}\right)^{n-k}.$$

$\binom{n}{k}\frac{k}{n}^k\left(1-\frac{k}{n}\right)^{n-k} = \binom{n}{k}2^{-nH(\frac{k}{n})}$ as seen above in deriving the upper bound. So

$$\frac{1}{n+1} \leq \binom{n}{k}2^{-nH(\frac{k}{n})},$$

and thus

$$\frac{1}{n+1}2^{nH(\frac{k}{n})} \leq \binom{n}{k}.$$

(c)

$$\begin{aligned}
Q^n(T(P)) &= \sum_{\mathbf{x}\in T(P)} Q^n(\mathbf{x}) \\
&= \sum_{\mathbf{x}\in T(P)} 2^{-n(D(P\|Q)+H(P))} \\
&= |T(P)|2^{-n(D(P\|Q)+H(P))}.
\end{aligned}$$

Using the bounds of part (b) we have,

$$\frac{1}{(n+1)^{|\mathcal{X}|}}2^{-nD(P\|Q)} \leq Q^n(T(P)) \leq 2^{-nD(P\|Q)}$$

and thus

$$Q^n(T(P)) \doteq 2^{-nD(P\|Q)}.$$

Problem 3

We have a stationary Markovian source which produces binary symbols according to the transition matrix P as follows

$$P = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}.$$

To find the stationary distribution of this Markov process we have to solve the following system of linear equations

$$\mu = \mu P.$$

Because of the symmetry that matrix P has, it can be easily guessed that the stationary distribution is $\mu = \left[\frac{1}{2}\ \frac{1}{2}\right]$, so we have $\mathbb{P}\left[X_i = 0\right] = \mathbb{P}\left[X_i = 1\right] = \frac{1}{2}$.

(a) From the definition of distribution function we can write

$$F(01110) = \mathbb{P}\left[0.X_1 \cdots X_5 < 0.01110\right].$$

We can expand the above probability as follows

$$\begin{aligned} F(01110) =& \mathbb{P}\left[X_1 < 0\right] + \mathbb{P}\left[X_1 = 0,\ X_2 < 1\right] + \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 < 1\right] \\ &+ \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 1,\ X_4 < 1\right] \\ &+ \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 1,\ X_4 = 1, X_5 < 0\right] \\ =& \mathbb{P}\left[X_1 = 0,\ X_2 = 0\right] + \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 0\right] \\ &+ \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 1,\ X_4 = 0\right] \\ =& \frac{1}{2}\frac{1}{3} + \frac{1}{2}\frac{2}{3}\frac{2}{3} + \frac{1}{2}\frac{2}{3}\frac{1}{3}\frac{2}{3} \\ =& (0.01110110100000\cdots)_2. \end{aligned}$$

(b) From the source, we have only observed the first five bits, 01110, which can be continued with an arbitrary sequence. However for and arbitrary sequence that starts with the sequence 01110 we have

$$F\left([01110, 00000\cdots]\right) \leq F\left([01110, X_6 X_7 X_8 \cdots]\right) \leq F\left([01110, 11111\cdots]\right),$$

where $[s_1, s_2]$ means concatenation of two sequences s_1 and s_2.

We know that

$$F\left([01110, 00000\cdots]\right) = F(01110) = (0.01110110100000\cdots)_2.$$

So it only remains to find $F\left([01110, 11111\cdots]\right)$. But we know that $0.0111011111\cdots = 0.01111$, so we have

$$\begin{aligned} F\left([01110, 11111\cdots]\right) =& F(01111) \\ & \mathbb{P}\left[X_1 < 0\right] + \mathbb{P}\left[X_1 = 0,\ X_2 < 1\right] + \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 < 1\right] \\ &+ \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 1,\ X_4 < 1\right] \\ &+ \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 1,\ X_4 = 1, X_5 < 1\right] \\ =& \mathbb{P}\left[X_1 = 0,\ X_2 = 0\right] + \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 0\right] \\ &+ \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 1,\ X_4 = 0\right] \\ &+ \mathbb{P}\left[X_1 = 0,\ X_2 = 1,\ X_3 = 1,\ X_4 = 1, X_5 = 0\right] \\ =& \frac{1}{2}\frac{1}{3} + \frac{1}{2}\frac{2}{3}\frac{2}{3} + \frac{1}{2}\frac{2}{3}\frac{1}{3}\frac{2}{3} + \frac{1}{2}\frac{2}{3}\frac{1}{3}\frac{1}{3}\frac{2}{3} \\ =& (0.011111010000\cdots)_2. \end{aligned}$$

So comparing binary representation of $F(01110)$ and $F(01111)$ we observe that we are sure about 4 bits: 0111.

Problem 4

The parsed string is as follows

$$\begin{array}{rcccccccccc}
\text{Original sequence:} & 0 & 00 & 000 & 1 & 10 & 101 & 0000 & 01 & 1010 & 1 \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\text{Using dictionary:} & D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 & D_8 & D_9 & ? \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\text{Compressed sequence:} & 0 & 00 & 00 & 100 & 100 & 101 & 000 & 0100 & 0111 & ?
\end{array}$$

The dictionaries that have been constructed are as follows

$$\begin{aligned}
D_1 &= \{0,\ 1\}, \\
\text{respectively encoded to: } E_{D_1} &= \{0,\ 1\}, \\
\\
D_2 &= \{00,\ 01,\ 1\}, \\
\text{respectively encoded to: } E_{D_2} &= \{00,\ 01,\ 10\}, \\
\\
D_3 &= \{000,\ 001,\ 01,\ 1\}, \\
\text{respectively encoded to: } E_{D_3} &= \{00,\ 01,\ 10,\ 11\}, \\
\\
D_4 &= \{0000,\ 0001,\ 001,\ 01,\ 1\}, \\
\text{respectively encoded to: } E_{D_4} &= \{000,\ 001,\ 010,\ 011,\ 100\}, \\
\\
D_5 &= \{0000,\ 0001,\ 001,\ 01,\ 10,\ 11\}, \\
\text{respectively encoded to: } E_{D_5} &= \{000,\ 001,\ 010,\ 011,\ 100,\ 101\}, \\
\\
D_6 &= \{0000,\ 0001,\ 001,\ 01,\ 100,\ 101,\ 11\}, \\
\text{respectively encoded to: } E_{D_6} &= \{000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110\}, \\
\\
D_7 &= \{0000,\ 0001,\ 001,\ 01,\ 100,\ 1010,\ 1011,\ 11\}, \\
\text{respectively encoded to: } E_{D_7} &= \{000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111\}, \\
\\
D_8 &= \{00000,\ 00001,\ 0001,\ 001,\ 01,\ 100,\ 1010,\ 1011,\ 11\}, \\
\text{respectively encoded to: } E_{D_8} &= \{0000,\ 0001,\ 0010,\ 0011,\ 0100,\ 0101,\ 0110,\ 0111,\ 1000\}, \\
\\
D_9 &= \{00000,\ 00001,\ 0001,\ 001,\ 010,\ 011,\ 100,\ 1010,\ 1011,\ 11\}, \\
\text{respectively encoded to: } E_{D_9} &= \{0000,\ 0001,\ 0010,\ 0011,\ 0100,\ 0101,\ 0110,\ 0111,\ 1000,\ 1001\}, \\
\\
D_{10} &= \{00000,\ 00001,\ 0001,\ 001,\ 010,\ 011,\ 100,\ 10100,\ 10101,\ 1011,\ 11\}, \\
\text{respectively encoded to: } E_{D_{10}} &= \{0000,\ 0001,\ 0010,\ 0011,\ 0100,\ 0101,\ 0110,\ 0111,\ 1000,\ 1001,\ 1010\}.
\end{aligned}$$

To encode the last "1" which is not in the dictionary D_{10}, one way is to use a special character or flag reserved to determine the last part of the sequence.

Problem 5

(a) The maximum length of the window is W, so to represent the pointer P we need $\lceil \log_2 W \rceil$ bits.The maximum matching length is M, so to represent the length of the match, L, we need $\lceil \log_2 M \rceil$ bits.

(b) The number of bits that are needed to represent a sequence of length L symbols with the matching method is

$$l_{\text{matching}} = 1 + \lceil \log_2 W \rceil + \lceil \log_2 M \rceil \quad \text{bits.}$$

The number of bits to represent a sequence of length L symbols using writing the characters themselves is

$$l_{\text{non-mathching}} = (1 + 8) \times L \quad \text{bits.}$$

So the sequence should be encoded using the matching method if

$$l_{\text{matching}} \leq l_{\text{non-matching}},$$

which means

$$1 + \lceil \log_2 W \rceil + \lceil \log_2 M \rceil \leq 9 \times L,$$

or

$$\frac{1 + \lceil \log_2 W \rceil + \lceil \log_2 M \rceil}{9} \leq L.$$

Thus, for any sequences longer that $\frac{1+\lceil \log_2 W \rceil + \lceil \log_2 M \rceil}{9}$, we have to encode it using the matching method unless it is better to encode with the format (F, C).