ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Handout 15 Information Theory and Coding
Solution 7 November 9, 2010, SG1 – 15:15-17:00

Problem 1. (a) $p_k(x) \leq \hat{p}(x)$. Therefore, $1 = \sum_x p_k(x) \leq \sum_x \hat{p}(x) = A$. Also $\hat{p}(x) \leq 1$. Therefore, $A = \sum_x \hat{p}(x) \leq \sum_x 1 = K$.

(b) Assume that the code is in a D-ary alphabet and the logarithm is with respect to base D. Then,

$$\sum_x D^{-l(x)} = \sum_x D^{-\lceil -\log_D \hat{p}(x) + \log_D A \rceil} \leq \sum_x D^{\log_D \hat{p}(x) - \log_D A} = \frac{\sum_x \hat{p}(x)}{A} = 1.$$

As $l(x)$ satisfies Kraft inequality, there exists a prefix-free code for X with codeword lengths equal to $l(x)$.

(c) As it is a prefix-free code, it satisfies $\bar{L}_k \geq H_k$. For the upper bound,

$$\begin{aligned}
\bar{L}_k &= \sum_x p_k(x) l(x) \\
&= \sum_x p_k(x) \lceil -\log_D \underbrace{\hat{p}(x)}_{\geq p(x)} + \log_D A \rceil \\
&\leq \sum_x p_k(x) \lceil -\log_D p(x) + \log_D A \rceil \\
&\leq \sum_x p_k(x) (-\log_D p(x) + \log_D A + 1) \\
&= H_k + \log_D A + 1.
\end{aligned}$$

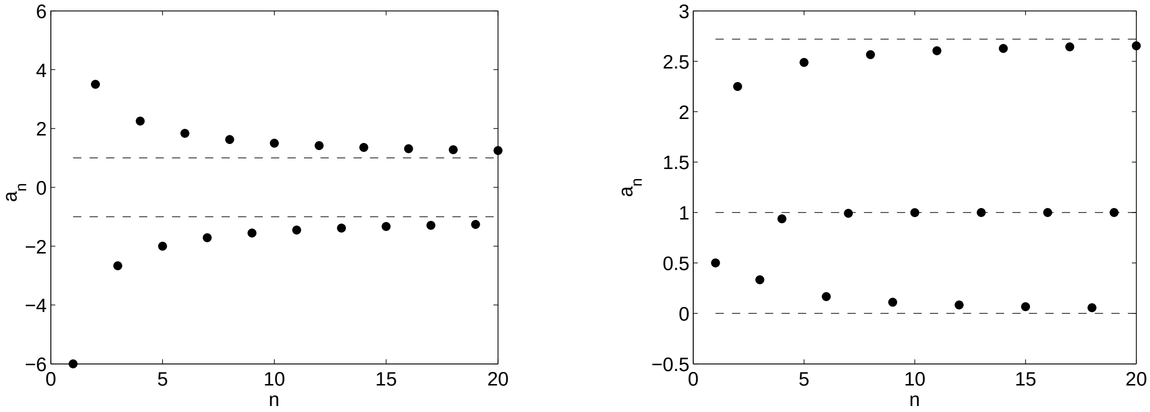
(d) For each symbol x, we choose the binary Huffman code with the shortest codeword length, that is, $l(x) = \min_k l_k(x)$ and we add $\lceil \log_2 K \rceil$ bits to the codeword to describe which of the K Huffman codes we used. As the distribution of X is one of the K different distributions $p_1, \cdots, p_K$, the shortest Huffman code will be the one corresponding to the the actual distribution of X. Therefore the chosen codeword will use no more than $H(X) + 1$ bits/symbol on average. As we add $\lceil \log_2 K \rceil$ bits to the codeword to describe which of the K Huffman codes we used, the total average length of the codeword can be up to $H(X) + \lceil \log_2 K \rceil + 1$.

Problem 2. Refer Handout 13 - Notes by Prof. Emre Telatar on the Lempel-Ziv Algorithm.

Problem 3. Let $s(m) = 0 + 1 + \cdots + (m-1) = m(m-1)/2$.

(a) Suppose we have a string of length $n = s(m)$. Then, we can certainly parse it into m words of lengths $0, 1, \cdots, (m-1)$, and since these words have different lengths, this is distinct parsing. As a parsing with the maximal number of distinct words will have at least as many words as this particular parsing, we conclude that whenever $n = m(m-1)/2$, $c \geq m$.

(b) An all zero string of length $s(m)$ can be parsed into at most m words. In this case, distinct words have distinct lengths.



(a) Problem 4(a) (b) Problem 4(b)

(c) Given any n, we can find m such that $s(m-1) \leq n < s(m)$. A string with n letters can be parsed into $(m-1)$ distinct words by parsing its initial segment of $s(m-1)$ letters with the above procedure, and concatenating the leftover letters to the last word. Thus, if a string an be parsed in $c = (m-1)$ distinct words, then $n < s(m)$, and in particular, $n < s(c+1) = c(c+1)/2$.

Problem 4. (a) Let $b_n = \inf\{a_k : k \geq n\}$. Then

$$\begin{aligned} b_n &= \inf\left\{(-1)^n\frac{(n+5)}{n}, (-1)^n\frac{(n+6)}{n+1}, \cdots\right\} \\ &= \begin{cases} -\frac{(n+5)}{n} & \text{for } n \text{ odd} \\ -\frac{(n+6)}{(n+1)} & \text{for } n \text{ even} \end{cases} \end{aligned}$$

Therefore $\liminf a_n = -1$. Similarly, $\limsup a_n = 1$.

(b) It can be seen that the limit points of $\{a_n\}$ are 0, 1, and e. Similar to (a), it can be shown that $\liminf a_n = 0$ and $\limsup a_n = e$.

The figures above give a graphical view of the lim sup, lim inf and the other limit points in this problem.

Problem 5.

$$\begin{aligned} Y_n &= \left(\prod_{i=1}^{n} X_i\right)^{\frac{1}{n}} \\ &= \left(2^{\log_2\left(\prod_{i=1}^{n} X_i\right)}\right)^{\frac{1}{n}} \\ &= 2^{\left(\frac{1}{n}\sum_{i=1}^{n}\log_2 X_i\right)} \\ &\to 2^{E[\log_2 X]} \qquad \text{as } n \to \infty \text{ with probability one.} \end{aligned}$$

Now, $E[\log_2 X] = \frac{1}{2}\log_2 1 + \frac{1}{4}\log_2 2 + \frac{1}{4}\log_2 3 = \log_2 6^{1/4}$. Therefore, $Y \to 6^{1/4}$ as $n \to \infty$ with probability one.

Problem 6. Since the probability of going to any of the other valid squares from a given square is equal, the stationary distribution is given by $\mu_i = E_i/E$, where E_i is the number of valid moves from square i and $E = \sum_{i=1}^{9} E_i$. From the 3×3 chessboard, it can be seen that each of the corners have 3 valid moves for the king, that is, $E_1 = E_3 = E_7 = E_9 = 3$, each of the edges have 5 valid moves for the king, that is, $E_2 = E_4 = E_6 = E_8 = 5$, and the center square has 8 valid moves for the king, that is, $E_5 = 8$. Therefore, $E = 40$, and so $\mu_1 = \mu_3 = \mu_7 = \mu_9 = 3/40$, $\mu_2 = \mu_4 = \mu_6 = \mu_8 = 5/40$, and $\mu_5 = 8/40$. As each of the valid moves are chosen with equal probability, $H(X_2|X_1 = i) = \log_2 3$ bits for $i = 1, 3, 7, 9$, $H(X_2|X_1 = i) = \log_2 5$ bits for $i = 2, 4, 6, 8$, and $H(X_2|X_1 = i) = \log_2 3$ bits for $i = 5$. Therefore, the entropy rate is

$$\begin{aligned} \mathcal{H} &= \sum_{i=1}^{9} \mu_i H(X_2|X_1 = i) \\ &= 0.3 \log_2 3 + 0.5 \log_2 5 + 0.2 \log_2 8 \\ &= 2.2365 \text{ bits/move.} \end{aligned}$$

Entropy rates of a rook and an even bishop are easier to compute as they always have the same number of squares to move to from any valid position on the chessboard. For a rook, there are always 4 valid moves from any square, and so a rook's stationary distribution can be shown to be uniformly distributed over all the squares of the chessboard. Therefore, its entropy rate can be easily computed as $\log_2 4 = 2$ bits/move. For an even bishop, there are always 2 valid moves from any valid square, and so an even bishop's stationary distribution can be shown to be uniformly distributed over all its valid squares, which are $2, 4, 6$, and 8. Its entropy rate is $\log_2 2 = 1$ bit/move.