ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 11 | Information Theory and Coding
Solutions to homework 5 | October 25, 2011

Problem 1. For a Markov chain, given X_0 and X_n are independent given X_{n-1}. Thus

$$H(X_0|X_nX_{n-1}) = H(X_0|X_{n-1})$$

But, since conditioning reduces entropy,

$$H(X_0|X_nX_{n-1}) \leq H(X_0|X_n).$$

Putting the above together we see that $H(X_0|X_{n-1}) \leq H(X_0|X_n)$.

Problem 2.

(a) Since the words of a valid and prefix condition dictionary reside in the leaves of a full tree, the Kraft inequality must be satisfied with equality: Consider climbing up the tree starting from the root, choosing one of the D branches that climb up from a node with equal probability. The probability of reaching a leaf at depth l_i is then D^{-l_i}. Since the climbing process will certainly end in a leaf, we have

$$1 = \Pr(\text{ending in a leaf}) = \sum_i D^{-l_i}.$$

(b) Multiplying both sides of the expression above by $D^{l_{max}}$, where l_{max} is the maximum length of a string, we have

$$D^{l_{max}} = \sum_i D^{l_{max}-l_i}$$

We also have that $\forall j \geq 0$, $D^j = 1 \mod (D-1)$. Taking $\mod (D-1)$ on both sides of the above expression, we have that

$$\begin{aligned}
1 &= (\sum_i D^{l_{max}-l_i}) \mod (D-1) \\
&= (\sum_i D^{l_{max}-l_i} \mod (D-1)) \mod (D-1) \\
&= (\sum_i 1) \mod (D-1) \\
&= (\text{Number of words}) \mod (D-1)
\end{aligned}$$

(c) If the dictionary is valid but not prefix-free, by removing all words that already have a prefix in the dictionary we would obtain a valid prefix-free dictionary. Since this reduced dictionary would satisfy the Kraft inequality with equality, the extra words would cause the inequality to be violated.

Problem 3.

(a) The number of 100-bit binary sequences with three or fewer ones is

$$\binom{100}{0} + \binom{100}{1} + \binom{100}{2} + \binom{100}{3} = 1 + 100 + 4950 + 161700 = 166751\,.$$

The required codeword length is $\lceil \log_2 166751 \rceil = 18$. (Note that the entropy of the source is $-0.005 \log_2(0.005) - 0.995 \log_2(0.995) = 0.0454$ bits, so 18 is quite a bit larger than the 4.5 bits of entropy per 100 source letters.)

(b) The probability that a 100-bit sequence has three or fewer ones is

$$\sum_{i=0}^{3} \binom{100}{i} (0.005)^i (0.995)^{100-i} = 0.60577 + 0.30441 + 0.7572 + 0.01243 = 0.99833$$

Thus the probability that the sequence that is generated cannot be encoded is $1 - 0.99833 = 0.00167$.

(c) In the case of a random variable S_n that is the sum of n i.i.d. random variables $X_1, X_2, \ldots, X_n$, Chebyshev's inequality states that

$$\Pr(|S_n - n\mu| \geq a) \leq \frac{n\sigma^2}{a^2},$$

where μ and σ^2 are the mean and variance of X_i. (Therefore $n\mu$ and $n\sigma^2$ are the mean and variance of S_n.) In this problem, $n = 100$, $\mu = 0.005$, and $\sigma^2 = (0.005)(0.995)$. Note that $S_{100} \geq 4$ if and only if $|S_{100} - 100(0.005)| \geq 3.5$, so we should choose $a = 3.5$. Then

$$\Pr(S_{100} \geq 4) \leq \frac{100(0.005)(0.995)}{(3.5)^2} \approx 0.04061\,.$$

This bound is much larger than the actual probability 0.00167.

Problem 4.

(a) Since the $X_1, \ldots, X_n$ are i.i.d., so are $q(X_1), q(X_2), \ldots, q(X_n)$, and hence we can apply the strong law of large numbers to obtain

$$\begin{aligned}
\lim -\frac{1}{n} \log q(X_1, \ldots, X_n) &= \lim -\frac{1}{n} \sum \log q(X_i) \\
&= -E[\log q(X)] \quad \text{w.p. } 1 \\
&= -\sum p(x) \log q(x) \\
&= \sum p(x) \log \frac{p(x)}{q(x)} - \sum p(x) \log p(x) \\
&= D(p||q) + H(X).
\end{aligned}$$

(b) Again, by the strong law of large numbers,

$$\begin{aligned}
\lim -\frac{1}{n} \log \frac{q(X_1, \ldots, X_n)}{p(X_1, \ldots, X_n)} &= \lim -\frac{1}{n} \sum \log \frac{q(X_i)}{p(X_i)} \\
&= -E\left[\log \frac{q(X)}{p(X)}\right] \quad \text{w.p. } 1 \\
&= -\sum p(x) \log \frac{q(x)}{p(x)} \\
&= \sum p(x) \log \frac{p(x)}{q(x)} \\
&= D(p||q).
\end{aligned}$$

Problem 5.

(a) Let I be the set of intermediate nodes (including the root), let N be the set of nodes except the root and let L be the set of all leaves. For each $n \in L$ define $A(n) = \{m \in N : m \text{ is an ancestor of } n\}$ and for each $m \in N$ define $D(m) = \{n \in L : n \text{ is a descendant of } m\}$. We assume each leaf is an ancestor and a descendant of itself. Then

$$E[\text{distance to a leaf}] = \sum_{n \in L} P(n) \sum_{m \in A(n)} d(m)$$
$$= \sum_{m \in N} d(m) \sum_{n \in D(m)} P(n) = \sum_{m \in N} P(m) d(m).$$

(b) Let $d(n) = -\log Q(n)$. We see that $-\log P(n_j)$ is the distance associated with a leaf. From part (a),

$$\begin{aligned} H(\text{leaves}) &= E[\text{distance to a leaf}] \\ &= \sum_{n \in N} P(n) d(n) \\ &= -\sum_{n \in N} P(n) \log Q(n) \\ &= -\sum_{n \in N} P(\text{parent of } n) Q(n) \log Q(n) \\ &= -\sum_{m \in I} P(m) \sum_{n:\ n \text{ is a child of } m} Q(n) \log Q(n) \\ &= \sum_{m \in I} P(m) H_{m'} \end{aligned}$$

(c) Since all the intermediate nodes of a valid and prefix condition dictionary have the same number of children with the same set of Q_n, each $H_n = H$. Thus $H(\text{leaves}) = H \sum_{n \in I} P(n) = HE[L]$.