ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 17 Information Theory and Coding
Midterm Solutions Nov. 8, 2011

Problem 1.

(a) Since $\mathcal{C}_0$ is a prefix-free code for the non-negative integers, the decoder, given a binary string, can 'climb the tree for $\mathcal{C}_0$' until it reaches a leaf and discover $\ell(u)$. It can then read $\ell(u)$ bits from the binary string which form $\mathcal{C}(u)$, since $\mathcal{C}$ is non-singular this uniquely identifies u. Thus the code $\tilde{\mathcal{C}}$ is uniquely decodable. Furthermore, since the decoder never needs to read any additional bits from the input while decoding u, we see that the $\tilde{\mathcal{C}}$ is *instantaneous*, and consequently prefix-free.

(b) Observe that

$$\sum_{n=0}^{\infty} 2^{-\ell_0(n)} \le \sum_{n=0}^{\infty} 2^{-2\log_2(n+1)-1} = \sum_{n=0}^{\infty} \frac{1}{2(n+1)^2} < 1.$$

Thus length function ℓ_0 satisfies the Kraft inequality, hence a prefix-free code with these lengths exist.

(c) We would order the binary strings from the shortest to longest: ϕ, 0, 1, 00, 01, 10, 11, 000, 001, ..., and assign them to the letters in the order of decreasing probability so the most probable letter gets the shortest codeword. In this assignment, we have:

Letters	length of the assigned string
1	0
2, 3	1
4, 5, 6, 7	2
8, ..., 15	3
...	...
$2^n, \dots, 2^{n+1}-1$	n
...	...

so we see that letter j is assigned a codeword of length $\lfloor \log_2 j \rfloor$.

(d) We have $1 = \sum_{i=1}^{K} p_i \ge \sum_{i=1}^{j} p_i \ge jp_j$, the last inequality because the sum has j terms, the smallest of which is p_j.

(e) Using part (b) we know that there is a code $\mathcal{C}_0$ for the non-negative integers with

$$\ell_0(n) = \lceil 2\log_2(n+1) + 1 \rceil \le 2\log_2(n+1) + 2.$$

With this code for the non-negative integers, we see that in the code $\tilde{\mathcal{C}}$ as in part (a) the letter j is assigned a codeword of length

$$\begin{aligned}
\tilde{\ell}(j) &= \ell_0(\lfloor \log_2 j \rfloor) + \ell(j) \\
&\le 2\log_2(\lfloor \log_2 j \rfloor + 1) + 2 + \ell(j) \\
&\le 2\log_2(\log_2 j + 1) + 2 + \ell(j) \\
&\le 2\log_2(\log_2(1/p_j) + 1) + 2 + \ell(j) && \text{by part (d).}
\end{aligned}$$

(f) It suffices to show the inequality for the expected length of the non-singular code $\mathcal{C}$ in part (c). Since $\tilde{\mathcal{C}}$ is uniquely decodable, $H(U) \le E[\tilde{\ell}(U)]$. Thus,

$$\begin{aligned} H(U) &\le E[\tilde{\ell}(U)] \\ &\le \sum_j p_j \bigl(2\log_2(\log_2(1/p_j)+1) + 2 + \ell(j)\bigr) \\ &= 2\sum_j p_j \bigl(\log_2(\log_2(1/p_j)+1)\bigr) + 2 + E[\ell(U)] \\ &\le 2\log_2\Bigl(\sum_j p_j \log_2(1/p_j) + 1\Bigr) + 2 + E[\ell(U)] \\ &= 2\log_2(H(U)+1) + 2 + E[\ell(U)]. \end{aligned}$$

PROBLEM 2.

(a) Since for large enough n we have

$$\Pr\bigl(U^n \in A\bigr) > 1-\epsilon,$$

we see that $1 - \Pr(U^n \in A \cap S) = \Pr(U^n \in A^c \cup S^c) < \epsilon + \delta$.

(b) Since for $u^n \in A$ we have $\Pr(U^n = u^n) \le 2^{-nH(p)(1-\epsilon)}$, we have

$$\begin{aligned} 1-\delta-\epsilon < \Pr(U^n \in S\cap A) &= \sum_{u^n \in S\cap A} \Pr(U^n = u^n) \\ &\le \sum_{u^n \in S\cap A} 2^{-nH(p)(1-\epsilon)} = |S\cap A| 2^{-nH(p)(1-\epsilon)}. \end{aligned}$$

(c) For $u^n \in A$ we have $\Pr(\tilde{U}^n = u^n) \ge 2^{-n[D(p\|\tilde{p})+H(p)](1+\epsilon)}$. Thus

$$\begin{aligned} \Pr\bigl(\tilde{U}^n \in S\bigr) &\ge \Pr\bigl(\tilde{U}^n \in S\cap A\bigr) \\ &= \sum_{u^n\in S\cap A} \Pr\bigl(\tilde{U}^n \in u^n\bigr) \\ &\ge \sum_{u^n\in S\cap A} 2^{-n[D(p\|\tilde{p})+H(p)](1+\epsilon)} \\ &\ge |S\cap A| 2^{-n[D(p\|\tilde{p})+H(p)](1+\epsilon)} \\ &\ge (1-\delta-\epsilon) 2^{-n(1+\epsilon)D(p\|\tilde{p})}\, 2^{-n2\epsilon H(p)}. \end{aligned}$$

(d) Letting

$$S = \{u^n : \text{the device decides } p\},$$

we see that α_n is exactly the probability that an i.i.d. sequence distributed with p falls outside S. When $\alpha_n \le \delta$, we see that S satisfies the conditions of the problem statement. Furthermore β_n is exactly the probability that an i.i.d. sequence distributed with $\tilde{p}$ falls in S, so, by part (c)

$$\beta_n \dot{\ge} 2^{-nD(p\|\tilde{p})}.$$

PROBLEM 3. Note that while $U_1, U_2, \dots$ is a Markov chain $V_1, V_2, \dots$ may not be. Consequently it is not an easy task to compute the entropy rate of the V process.

(a)

(A1) Conditioning further on $U_1, \dots, U_n$ reduces entropy, and when $U_1, \dots, U_n$ are given, $V_1, \dots, V_n$ are determined and can be dropped without changing entropy.

(A2) Given U_n, the future $U_{n+1}, U_{n+2}, \dots$ are independent of the past $U_1, \dots, U_{n-1}$. Since $V_{n+1}, \dots, V_{n+2}, \dots$ are functions of $U_{n+1}, U_{n+2}, \dots$, they are also independent of $U_1, \dots, U_{n-1}$ once U_n is given. Thus, $U_1, \dots, U_{n-1}$ can be dropped from the conditioning without changing entropy.

(A3) By stationarity, the time index can be shifted by $n-1$.

(A4) V_1 is determined by U_1 so it can be added without changing entropy.

(b) Taking the limit as $n \to \infty$ of the both sides of the inequality shown in (a)

$$H(V_{m+n}|V_{m+n-1}, \dots, V_1) \geq H(V_{m+1}|V_m, \dots, V_1, U_1)$$

and noting that the right hand side has no n, we see that

$$H_V \geq H(V_{m+1}|V_m, \dots, V_1, U_1).$$

(c) This is by definition of conditional mutual information.

(d) (D1) because $H(U_1|V_1, V_2, \dots)$ is non-negative.

(D2) chain rule for mutual information.

(e) Defining $r_m = I(U_1; V_{m+1}|V_1, \dots, V_m)$, we see from (d) that r_m is a sequence with $\sum_m r_m < \infty$. Thus r_m converges to zero.

(f) By part (c) and (e) we see that the sequence $a_m = H(V_{m+1}|V_m, \dots, V_1, U_1)$ has the same limit as the sequence $b_m = H(V_{m+1}|V_m, \dots, V_1)$. But b_m converges to H_V. Thus a_m also coverges to H_V and by (b) it does so from below.

Note that we know that b_m coverges to H_V from above, so the conclusion that a_m converges to H_V from below gives us a computational method to approximate H_V to any desired accuracy: compute $a_1, b_1, a_2, b_2, \dots$, until $b_m - a_m$ is smaller than the desired accuracy of approximation.