ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 16 Information Theory and Coding
Midterm solutions Oct. 30, 2018

PROBLEM 1.

(a) Recall that $\mathcal{C}$ is uniquely decodable means that $\mathcal{C}^*$ is injective, i.e., for any $u^n \neq v^m$ we have $\mathcal{C}^n(u^n) \neq \mathcal{C}^m(v^m)$. In particular, whenever $u^n \neq v^n$ we have $\mathcal{C}^n(u^n) \neq \mathcal{C}^n(v^n)$. The last statement is the definition of $\mathcal{C}^n$ being injective.

(b) Since we are supposed to show that $u_1 \neq v_1$, we may assume that $|\mathcal{U}| \geq 2$.

If $\mathcal{C}$ is not uniquely decodable, then there are $u^n \neq v^m$ such that $\mathcal{C}^n(u^n) = \mathcal{C}^m(v^m)$. Among all such (u^n, v^m) choose one for which $n+m$ is smallest, and assume (without loss of generality) that $m \leq n$. If $m \geq 1$ we are done, since in this case we must have $u_1 \neq v_1$ (because, if not, we can replace u^n by $\tilde{u}^{n-1} = u_2 \dots u_n$ and v^m by $\tilde{v}^{m-1} = v_2 \dots v_m$, contradicting $m+n$ being smallest).

Otherwise, $m = 0$ and $v^m = \lambda$ (the null string) with $\mathcal{C}(v^m) = \lambda$. Since $u^n \neq v^m = \lambda$ and $\mathcal{C}(u^n) = \lambda$, we have a letter $a = u_1 \in \mathcal{U}$ such that $\mathcal{C}(a) = \lambda$. Take now any letter $b \in \mathcal{U}$ with $b \neq a$, and note that $\mathcal{C}^2(ab) = \mathcal{C}^1(b)$, i.e., there are two source sequences that differ in their first letter and have the same representation.

(c) $\mathcal{C}$ is not uniquely decodable means that there is $u^n \neq v^m$ such that $\mathcal{C}^n(u^n) = \mathcal{C}^m(v^m)$. If $n = m$ then we are done: this would by definition mean that be $\mathcal{C}^n$ is not injective. If $n \neq m$, we could attempt the following reasoning: observe $\mathcal{C}^*(u^n v^m) = \mathcal{C}^*(v^m u^n)$ and conclude that $\mathcal{C}^{m+n}$ is not injective. However this reasoning fails because we can't be sure that $u^n v^m \neq v^m u^n$ just because $u^n \neq v^m$. (E.g., suppose $u^n = a$ and $v^m = aa$). This is the reason the problem has "part (b)":

As $\mathcal{C}$ is not uniquely decodable, we can find u^n and v^m as in part (b). Now observe that (i) $u^n v^m \neq v^m u^n$ (as they differ in their first letter), (ii) $u^n v^m$ and $v^m u^n$ have the same length $k = n + m$, and $\mathcal{C}^k(u^n v^m) = \mathcal{C}^k(v^m u^n)$, i.e., $\mathcal{C}^k$ is not singular.

Moral of the problem: it is clear that the statement "$\mathcal{C}^*$ is injective" is a stronger statement than "for every n, $\mathcal{C}^n$ is injective" — since the first ensures that $u^n \neq v^m$ are assigned different codewords not only when $n = m$ but also for $n \neq m$ — so part (a) is unsurprising. The statement "$\mathcal{C}^n$ is injective for each n" only means that different source sequences of *same length* get different representations; it is not immediately clear that this will also imply that source sequences of *different lengths* also get different representations. Part (c) shows this is indeed the case: that injectiveness of $\mathcal{C}^n$ for every n implies the injectiveness of $\mathcal{C}^*$.

PROBLEM 2.

(a) Note that $H(X^n) = \sum_{i=1}^n H(X_i|X^{i-1})$. Since X^{i-1} is part of Y_i, and since conditioning reduces entropy $H(X_i|Y_i) \leq H(X_i|X^{i-1})$, and the inequality follows.

(b) Since $H(X^n) = H(Y_i) + H(X_i|Y_i)$, we have $nH(X^n) = \sum_i H(Y_i) + \sum_i H(X_i|Y_i)$. Thus (b) and (a) are equivalent statements.

(c) Since (a) and (b) are equivalent statements, we need only consider the condition for equality in (a). Accordingly suppose equality in (a) holds. It then follows that $H(X_i|X^{i-1}) = H(X_i|Y_i)$ for each i, and in particular that $H(X_1) = H(X_1|Y_1)$.

Observe now that both $H(X^n)$ and $\sum_i H(X_i|Y_i)$ remain unchanged if we permute $X_1, \dots, X_n$. So, equality in (a) will not only imply that $H(X_1) = H(X_1|Y_1)$ but also $H(X_2) = H(X_2|Y_2)$, ..., $H(X_n) = H(X_n|Y_n)$. (One can also see this by expanding $H(X^n)$ by the chain rule in different orders). We see that each X_i is independent of Y_i and thus, that $\{X_i\}$ are independent (but not necessarily identically distributed) random variables.

It is easy to check that the independence of $\{X_i\}$ is sufficient for equality in (a) to hold (via $H(X_i|Y_i) = H(X_i|X^{i-1}) = H(X_i)$). Thus we have shown that the independence of the random variables $X_1, \dots, X_n$ is necessary and sufficient condition for equality to hold.

The inequality (b) (or equivalently (a)) is a special case ($A_{n-1}/(n-1) \geq A_n/n$) of Han's equality, which say that if $X_1, \dots, X_n$ are random variables, and we compute the average A_k of all $H((X_i : i \in S))$ with $S \subset \{1, \dots, n\}$ of size k, then $A_1 \geq A_2/2 \geq \dots A_n/n$.

PROBLEM 3.

(a) The calculation $\Pr(X_2 = 4) = (1-p)/2 \neq (1-p) = \Pr(X_1 = 4)$ shows that the process is not stationary.

(b) With $h_2(\beta) = -\beta\log(\beta) - (1-\beta)\log(1-\beta)$, we have $H(X_{n+1}|X_n = 1) = H(X_{n+1}|X_n = 2) = h_2(\alpha)$ and $H(X_{n+1}|X_n = 3) = H(X_{n+1}|X_n = 4) = h_2(1/2) = 1$. The answer is independent of n.

(c) Since the process is Markov, $H(X_n|X^{n-1}) = H(X_n|X_{n-1})$ for $n > 1$. By part (b) we get $a_1 = h_2(p)$, $a_n = H(X_n|X_{n-1}) = h_2(\alpha)[Pr(X_n = 1) + Pr(X_n = 2)] + h_2(1/2)[Pr(X_n = 3) + Pr(X_n = 4)] = h_2(\alpha)p + (1-p)$ for $n > 1$.

(d) Using the chain rule $b_n = (a_1 + \cdots + a_n)/n$, and by (c) $b_n = a_1/n + (n-1)a_2/n$,

(e) As $\lim_n b_n = a_2$, H exists and equals a_2. Note that $b_n \neq a_n$.

Moral: we know that entropy rate is well defined for stationary sources; here we have seen an instance of a non-stationary source for which the entropy rate is still defined.

PROBLEM 4.

(a) Since $L = j$ means that $2^j \leq U < 2^{j+1}$, we see that conditioned on $\{L = j\}$, U can take 2^j values. Thus $H(U|L = j) \leq j$.

(b) As $H(U|L) = \sum_j H(U|L = j)\Pr(L = j)$, by part (a) we find $H(U|L) \leq \sum_j j\Pr(L = j) = E[L]$.

(c) Note that $H(U) \leq H(UL) = H(L) + H(U|L)$. (Indeed, since L is a function of U, we even have $H(U) = H(UL)$.) The conclusion now follows by part (b).

(d) Note that for any i, $1 \geq \Pr(U \leq i)$. But $\Pr(U \leq i) = \sum_{j=1}^{i} \Pr(U = j) \geq i\Pr(U = i)$ since each term in the sum is at least $\Pr(U = i)$.

(e) By (d) we get $\log_2 u \leq -\log_2 \Pr(U = u)$ for $u = 1, 2, \ldots$. Multiplying both sides by $\Pr(U = u)$ and summing over u we get $E[\log_2 U] \leq H(U)$. As $L = \lfloor \log_2 U \rfloor \leq \log_2 U$, we conclude that $E[L] \leq E[\log_2 U] \leq H(U)$.

(f) Since $f(n, \mu) = (n+1)\log(\mu+1) - n \log \mu$, we see that $E[f(G, \mu)] = (E[G]+1)\log(\mu+1) - E[G]\log \mu$ and $E[f(N, \mu)] = (E[N] + 1)\log(\mu + 1) - E[N]\log \mu$. Since $E[G] = E[N]$, we get $E[f(G, \mu)] = E[f(N, \mu)]$. Remembering that $f(n, \mu) = -\log p_G(n)$, we see that $H(G) = E[f(G, \mu)]$. Thus

$$H(G) - H(N) = E[f(G, \mu)] - H(N) = E[f(N, \mu)] - H(N) = \sum_n p_N(n) \log \frac{p_N(n)}{p_G(n)}$$

so, we see that $H(G) - H(N) = D(p_N \| p_G) \geq 0$ and consequently $H(N) \leq H(G)$. Moreover $H(G) = E[f(G, \mu)] = (\mu + 1)\log(\mu + 1) - \mu \log \mu = g(\mu)$.

(g) By (c) we have $E[L] \geq H(U) - H(L)$. By (f) we have $H(L) \leq g(E[L])$. As g is increasing (by computing $g'(\mu) = \log(1 + \mu) - \log \mu > 0$), by part (e) we further find $g(E[L]) \leq g(H(U))$. Thus, $E[L] \geq H(U) - g(H(U))$.

Moral of the problem: Consider designing an injective code for a random variable U. By labelling the values of U as 1, 2, ..., with 1 denoting the most probable value of U, 2 the next probable, etc., we can assume without loss of generality that U is as in (d). The injective code with shortest expected length will assign the binary strings $\lambda, 0, 1, 00, 01, 10, 11, 000, \ldots$ to the values $1, 2, 3, 4, \ldots$ of U in that order. Note that in this assignment the binary string assigned to the letter u has length exactly $\lfloor \log_2 u \rfloor$. Thus (g) gives a lower bound to the expected codeword length of the best code (and thus any injective code) in terms of the entropy. As $g(x)$ is a function that is $O(\log x)$, we conclude that relaxing the requirement of unique decodability to injectivity does not yield a substantive improvement on expected codeword length.