ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Handout 16 — Information Theory and Coding

Midterm solutions — Oct. 31, 2017

PROBLEM 1.

(a) We already know that

$$H(X) + H(Y) \geq H(XY),$$
$$H(Y) + H(Z) \geq H(YZ),$$

and

$$H(Z) + H(X) \geq H(ZX).$$

Adding these inequalities together and diving by two gives

$$H(X) + H(Y) + H(Z) \geq \frac{1}{2}\big[H(XY) + H(YZ) + H(ZX)\big].$$

(b) The difference between the left and right sides, i.e.,

$$H(XY) + H(YZ) - H(XYZ) - H(Y),$$

equals

$$H(X|Y) - H(X|YZ) = I(X;Z|Y),$$

which is always positive.

(c) Using (b) with (YZX) and (ZXY) in the role of (XYZ) gives the inequalities

$$H(YZ) + H(ZX) \geq H(XYZ) + H(Z)$$

and

$$H(ZX) + H(XY) \geq H(XYZ) + H(X).$$

Adding the inequality in (b) to these two gives

$$2\big[H(XY) + H(YZ) + H(ZX)\big] \geq 3H(XYZ) + H(X) + H(Y) + H(Z).$$

(d) Since $H(X) + H(Y) + H(Z) \geq H(XYZ)$, (c) yields

$$2\big[H(XY) + H(YZ) + H(ZX)\big] \geq 4H(XYZ).$$

(e) Let $\{(x_i, y_i, z_i) : i = 1, \dots, n\}$ be the xyz-coordinates of the n points. Let X, Y and Z be random variables with $\Pr\big((X, Y, Z) = (x_i, y_i, z_i)\big) = 1/n$ for every $1 \leq i \leq n$. Then, $H(XYZ) = \log_2 n$. Furthermore, the random pair (XY) takes values in the projection of the n points to the xy plane and similarly for (YZ) and (ZX). Thus $H(XY) \leq \log_2 n_{xy}$, $H(YZ) \leq \log_2 n_{yz}$, and $H(ZX) \leq \log_2 n_{zx}$. Part (d) now yields

$$\log_2[n_{xy}n_{yz}n_{zx}] \geq H(XY) + H(YZ) + H(ZX) \geq 2H(XYZ) = 2\log_2 n,$$

which implies that $n_{xy}n_{yz}n_{zx} \geq n^2$.

The relationship between $H(XYZ)$ and $H(XY)$, $H(YZ)$ and $H(ZX)$ is a special case of Han's inequality, which, for a collection of n random variables relates the sum of the $\binom{n}{k}$ joint entropies of k out of n random variables to the sum of the $\binom{n}{k+1}$ entropies of $k+1$ out of n random variables.

The combinatorial fact about the projections of points in 3D is known as Shearer's lemma.

Problem 2. Observe first that

$$\sum_u Q(u) = \sum_{j\geq 0} \sum_{u=2^j}^{2^{j+1}-1} Q(u) = \sum_{j\geq 0} 2^j(1-2p)p^j = 1,$$

so Q is indeed a distribution.

(a)
$$\begin{aligned} H(V) &= \sum_{j\geq 0} 2^j(1-2p)p^j \Big[\log_2 \frac{1}{1-2p} + j\log_2\frac{1}{p}\Big] \\ &= \log_2\frac{1}{1-2p} - (1-2p)(\log_2 p)\sum_j j(2p)^j \\ &= -\log_2(1-2p) - \frac{2p}{1-2p}\log_2 p. \end{aligned}$$

(b) $L(V) = E[\lfloor \log_2 V\rfloor] = \sum_{j\geq 0} j2^j(1-2p)p^j = \dfrac{2p}{1-2p}.$

(c) With $L = L(V)$, by (b), we have $2p = L/(L+1)$ and $1-2p = 1/(L+1)$. Thus by (a) we have

$$H(V) = \log_2(L+1) + L\log_2\frac{2(L+1)}{L} = L + L\log_2(1+1/L) + \log_2(1+L),$$

which is (c1). On the other hand, we already know that $\ln(1+x) \leq x$, so

$$\log_2(1+x) = \ln(1+x)\log_2 e \leq x\log_2 e.$$

Using this with $x = 1/L$ yields (c2), i.e., $H(V) \leq L + \log_2(1+L) + \log_2 e$.

(d) With $P(i) = \Pr(U = i)$, we have

$$H(U) = \sum_i P(i)\log_2[1/P(i)] = \sum_i P(i)\log_2[1/Q(i)] - \sum_i P(i)\log_2[P(i)/Q(i)].$$

As the second sum is a divergence, it is non-negative, consequently (d1) follows. Since $\log_2\dfrac{1}{Q(i)}$ is of the form $a\lfloor\log_2 i\rfloor + b$, the sums $\sum_i P(i)\log_2\dfrac{1}{Q(i)}$ and $\sum_i Q(i)\log_2\dfrac{1}{Q(i)}$ equal $aL(U)+b$ and $aL(V)+b$ respectively. Now (d2) follows because $L(U) = L(V)$.

(e) Since $P(1) \geq P(2) \geq \dots$, the optimal non-singular code $\mathcal{C}$ will assign the ith shortest binary string to the letter i. As the ith shortest binary string has length $\lfloor\log_2 i\rfloor$, we see that length $\mathcal{C}(u) = \lfloor \log_2 u\rfloor$.

(f) One-to-one transformations of a random variable changes neither the entropy nor L^*. We can thus assume without loss of generality that the distribution P of the random variable U satisfies $P(1) \geq P(2) \geq \dots$. By (e) $L^* = \sum_u P(u)\lfloor\log_2 u\rfloor$, and by (d) $H(U) \leq H(V)$ for the random variable V with distribution Q — where p is chosen so that $L^* = L(V) = 2p/(1-2p)$. By (c) we know that $H(V) \leq L^* + \log_2(1+L^*) + \log_2 e$, hence $H(U) \leq L^* + \log_2(1+L^*) + \log_2 e$.

What we have shown is that even if we relax the unique decodability requirement on a code to non-singularity, the expected codeword length L^* is not too small compared to the entropy: even if $L^* \leq H$, by (f) we would have $H \leq L^* + \log_2(1+L^*) + \log_2 e \leq L^* + \log_2(1+H) + \log_2 e$ so that $L^* \geq H - \log_2(1+H) - \log_2 e$.

Problem 3.

(a) Even though the decoder can not find out the value of u_{i+l+1} from the description of the word w, it can determine u_{i+l+1} once it receives the index of the next word w': either

  (i) w' is not the newly added word, in which case u_{i+1+l} is the first letter of w', or

  (ii) w' is the newly added word, unknown to the decoder. But w is a prefix of w', so u_{i+l+1} is the first letter of w.

(b) As the algorithm always looks for the longest word w in the dictionary that matches the start of the as-yet-unprocessed segment of the input, wu_{i+l+1} is not in the dictionary before its addition to the dictionary in step 4. Thus the words added to the dictionary are distinct. For each occurrence of a word w in the parsing a word of the form wu with $u \in \mathcal{U}$ is added to the dictionary. Since these are distinct, w cannot appear more than $|\mathcal{U}|$ times in the parsing.

(c) There are $|\mathcal{U}|^i$ words of length i, and by (b), each can appear at most $\mathcal{U}$ times in the list $w_1, \dots, w_m$. As the algorithm never parses the null string, at most $F(k) = |\mathcal{U}| \sum_{i=1}^{k-1} |\mathcal{U}|^i$ words in the list are of length $k-1$ or less, and each of the remaining words in the list has length k or more. Thus $n \geq k[m - F(k)]$.

(d) By (c) $\frac{m(u^n)}{n} \leq \frac{1}{k} + \frac{F(k)}{n}$. Thus, $\limsup_{n\to\infty} m(u^n)/n \leq 1/k$. As k is arbitrary, the conclusion follows.

(e) The dictionary size s is initially $|\mathcal{U}|$, and increases by 1 at each parsing. Thus, after the m'th parse the algorithm has output m binary strings, each of length at most $\lceil \log_2(|\mathcal{U}| + m) \rceil$.

(f) Let $u^n = w_1 \dots w_m$ be the parsing of the input by the algorithm. Let z_i be the state the IL machine is in at just before it reads w_i, and z_{i+1} be the state just after it has read w_i. Let t_i be the binary string output by the IL machine while it reads w_i. No string t can occur in $t_1, \dots, t_m$ more than $k = s^2|\mathcal{U}|$ times. If it did, there will be a state-pair (z, z') that occurs among (z_i, z_{i+1}) more than $|\mathcal{U}|$ times with $t_i = t$. By (b) there will be $w_i \neq w_j$ with $t_i = t_j = t$ and $(z_i, z_{i+1}) = (z_j, z_{j+1})$. But this contradicts the IL property of the machine. Thus the output of the IL machine $t_1 \dots t_m$ is a concatenation of m binary strings with no string occurring more than $k = s^2|\mathcal{U}|$ times, so their total length is at least $L(m, k)$.

This version of Lempel-Ziv is known as the Lempel-Ziv-Welch (LZW) algorithm and is the one commonly implemented. In (f) we have shown that no IL machine with fewer than s states can output fewer than $L(m(u^n), s^2|\mathcal{U}|)$ bits when it processes u^n. In the notes on LZ we had shown that $L(m, k) \geq m \log(m/(8k))$. With $m = m(u^n)$ and $k = s^2|\mathcal{U}|$, by (e), the number of bits per letter LZW emits is at most

$$\frac{m}{n}\lceil \log(|\mathcal{U}| + m) \rceil - \frac{m}{n} \log \frac{m}{8k} \leq \frac{m}{n} \log \frac{16k(|\mathcal{U}| + m)}{m}$$

more than that of an IL machine. As n gets large, the argument of the log approaches $16k$, m/n approaches 0, and so the right hand side above approaches 0. Thus, LZW competes well against the class of finite state IL machines, and we see that it (just as LZ) is an asymptotically optimal method to compress an infinite sequence $u_1u_2\dots$.