ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 13 Information Theory and Coding
Homework 6 Oct. 22, 2019

PROBLEM 1. Suppose $L : \mathbb{R}^K \to \mathbb{R}^N$ is a linear function and $g : \mathbb{R}^N \to \mathbb{R}$ is a concave function. Show that $f : \mathbb{R}^K \to \mathbb{R}$ defined as $f(x) = g(L(x))$ is concave.

PROBLEM 2. From the notes on Lempel-Ziv algorithm, we know that the total length n of c distinct binary strings satisfies

$$n > c \log_2(c/8)$$

, where c is the maximum possible number of distinct parsings. The same technique, when applied to the non-binary strings yields

$$n > c \log_K(c/K^3)$$

where K is the size of the alphabet the letters of the string belong to. This inequality lower bounds n in terms of c. We will now show that n can also be upper bounded in terms of c.

(a) Show that, if $n \geq \frac{1}{2}m(m-1)$, then $c \geq m$.

(b) Find a sequence for which the bound in (a) is met with equality.

(c) Show now that $n < \frac{1}{2}c(c+1)$.

PROBLEM 3. Let the alphabet be $\mathcal{X} = \{a, b\}$. Consider the infinite sequence $X_1^\infty = ababababababab\ldots$

(a) What is the compressibility of $\rho(X_1^\infty)$ using finite-state machines (FSM) as defined in class? Justify your answer.

(b) Design a specific FSM, call it M, with at most 4 states and as low a $\rho_{\text{M}}(X_1^\infty)$ as possible. What compressibility do you get?

(c) Using only the result in point (a) but no specific calculations, what is the compressibility of X_1^∞ under the Lempel–Ziv algorithm, i.e., what is $\rho_{\text{LZ}}(X_1^\infty)$?

(d) Re-derive your result from point (c) but this time by means of an explicit computation.

PROBLEM 4. We are given a memoryless stationary binary symmetric channel BSC(ϵ). Namely, if $X_1, \ldots, X_n \in \{0, 1\}$ are the input of this channel and $Y_1, \ldots, Y_n \in \{0, 1\}$ are the output, we have:

$$P(Y_i | X_i, X^{i-1}, Y^{i-1}) = P(Y_i | X_i) = \begin{cases} 1 - \epsilon & \text{if } Y_i = X_i, \\ \epsilon & \text{otherwise.} \end{cases}$$

Let W be a random variable that is uniform in $\{0, 1\}$ and consider a communication system with feedback which transmits the value of W to the receiver as follows:

- At time $t = 1$, the transmitter sends $X_1 = W$ through the channel.

- At time $t = i + 1 \le n$, the transmitter gets the value of Y_i from the feedback and sends $X_{i+1} = Y_i$ through the channel.

(a) Give the capacity C of the channel in terms of ϵ, and show that $C = 0$ when $\epsilon = \frac{1}{2}$.

(b) Show that if $\epsilon = \frac{1}{2}$, $I(X^n; Y^n) = n - 1$. This means that $I(X^n; Y^n) \le nC$ does not hold for this system.

(c) Show that although $I(X^n; Y^n) > nC$ when $\epsilon = \frac{1}{2}$, we still have $I(W; Y^n) \le nC$.

Note that since W is the useful information that is being transmitted, it is the value of $I(W; Y^n)$ that we are interested in when we want to compute the amount of information that is shared with the receiver.

PROBLEM 5. Consider the following variation on the Lempel-Ziv algorithm to encode an infinite sequence $u_1 u_2 \dots$ from an alphabet $\mathcal{U}$.

1. Set the dictionary $\mathcal{D} = \mathcal{U}$. Denote the dictionary entries as $d(0), \dots, d(s-1)$, with $s = |\mathcal{U}|$ being the size of the dictionary. Set $i = 0$ (the number of input letters read so far).
2. Find the largest l such that $w = u_{i+1} \dots u_{i+l}$ is in $\mathcal{D}$.
3. With $0 \le j < s$ denoting the index of w in $\mathcal{D}$, output the $\lceil \log_2 s \rceil$ bit binary representation of j.
4. Add the word $w u_{i+l+1}$ to $\mathcal{D}$, i.e., set $d(s) = w u_{i+l+1}$, and increment s by 1. Increment i by l. Goto step 2.

For example, with $\mathcal{U} = \{\mathtt{a}, \mathtt{b}\}$, the input string `abbbbaaab...` will lead to the execution steps

$\mathcal{D}$ at 2	w	output at 3	added-word at 4
a b	a	0	ab
a b ab	b	01	bb
a b ab bb	bb	11	bbb
a b ab bb bbb	b	001	ba
a b ab bb bbb ba	a	000	aa
a b ab bb bbb ba aa	aa	110	aab

(a) (5 pts) Can the decoder reconstruct the input sequence $u_1 u_2 \dots$ from the output of the algorithm? If so, how? (The crucial difficulty is that the description of w in step 3 does not determine the word added to the dictionary in step 4.)

(b) (5 pts) The algorithm parses the sequence $u_1 u_2 \dots$ into a sequence of words $w_1 w_2 \dots$, (the w's found in step 2). Show that a word w can appear at most $|\mathcal{U}|$ times in the parsing.

(c) (5 pts) Suppose $u^n = u_1 \dots u_n$ is parsed into $m(u^n)$ words $w_1 \dots w_m$ by the algorithm. Show that for any $k \ge 1$

$$n \ge k[m(u^n) - F(k)],$$

where $F(k) = |\mathcal{U}| \sum_{i=1}^{k-1} |\mathcal{U}|^i$.

(d) (5 pts) Show that $\lim_{n\to\infty} m(u^n)/n = 0$.

(e) (5 pts) Show that after reading u^n the algorithm outputs fewer than $m(u^n)\lceil \log_2[|\mathcal{U}| + m(u^n)]\rceil$ bits.

Let $L(m, k)$ denote the minimum possible total length of a collection of m binary strings where no string appears more than k times.

(f) (5 pts) Show that if u^n is fed to an information lossless finite state machine with s states, then the machine outputs at least $L(m(u^n), s^2|\mathcal{U}|)$ bits.