CSC 310S
Spring, 2003

March 10, 2003
DUE: April 2 / 4, 2003

Information Theory
Assignment #3
[Model Answers]

[Note: There are four questions in this assignment, but your basic mark for the assignment (out of 60) will be based only on the three questions that you answer best. If you answer all four, then your fourth best answer will give you "extra credit" in the course (between 1% and 5% added to your final mark).]

1. Lempel-Ziv-Welch Coding [20 marks]

Consider using Lempel-Ziv-Welch for encoding *binary strings*, so that the source alphabet is $\{0, 1\}$. Use knowledge of the relationships among (i) T, the length of text to be encoded, (ii) $c(T)$, the number of codes transmitted, (iii) the number of dictionary entries at the end of the transmission, and (iv) the sum lengths of all the strings in the dictionary entries to answer the following questions. (In all parts, assume that source symbols are coded in 8 bits each, so that the compression ratio is: $\frac{B \times c(T)}{8 \times T}$)

A. If the LZW algorithm is applied to a source stream of length T that consists only of *one* bits:

"11111111111111111111111 ... 1111111111111"

(i). Obtain an *approximate* expression for the value of $c(T)$ in terms of T.

The code dictionary will consist of $c(T)$ entries, the first of which is zero, and the remainder of which are successively longer strings of ones, starting at length one and going up to length $c(T) - 1$. Since the total length of the codewords in the dictionary must be $T + c(T)$, we have the approximate relationship that:

$$T = \frac{c(T)^2}{2} \qquad or \qquad c(T) = \sqrt{2\,T}$$

(ii). If the dictionary codes are B bits long, find an *approximate* expression for R, the ratio of the length of the encoded stream to the length of the original stream.

$$R = \frac{B\ c(T)}{8\ T} = \frac{B}{8}\sqrt{\frac{2}{T}}$$

(iii). Evaluate the expressions obtained in parts (i) and (ii) for the three indicated pairs of values for B and T:

- ($B = 8$, $T = 1{,}000$)
- ($B = 12$, $T = 1{,}000{,}000$)
- ($B = 16$, $T = 1{,}000{,}000{,}000$)

B	T	$c(T)$	R
8	10^3	45	.045
12	10^6	1414	.002
16	10^9	44,700	.0001

B. Perform all the same calculations as in part A, but this time assuming that the source string of length T consists entirely of *strictly* alternating ones and zeroes:

"10101010101010101010101 ... 101010101010"

For each length of entry, both the one starting with 0 and the one starting with 1 will be present. (E.g., for length 4, both "0101" and "1010" will be among the code-words.) so, this time:

$$T = \frac{c(T)^2}{4} \qquad or \qquad c(T) = 2\sqrt{T}$$

$$R = \frac{B\ c(T)}{8\ T} = \frac{B}{4\sqrt{T}}$$

B	T	$c(T)$	R
8	10^3	63	.063
12	10^6	2000	.003
16	10^9	63,245	.00013

C. Finally, consider the case where the source string is a truly random bit string with each bit being independent of previous bits, and being equally likely to be either a "0" or a "1". Produce expressions for rough (but justifiable) upper bounds on $c(T)$ and R for this case, and use them to express bounds on the values of $c(T)$ and R for the following two cases:

- ($B = 8$, $T = 1{,}000$)
- ($B = 16$, $T = 1{,}000{,}000$)

The largest possible value of c(T) will occur if it happens that the shortest possible strings make it into the dictionary. This happens if the dictionary includes all 2^j strings of length j for j from 1 to x, where x satisfies:

$$T = \sum_{j=1}^{x} j\ 2^j \qquad so\ \ that \qquad c(T) = \sum_{j=1}^{x} 2^j$$

Solving this approximately yields:

B	T	$c(T)$	R
8	10^3	180	.18
16	10^6	71,000	.14

2. Information Channels with Noise [20 marks]

Consider an Information Channel (with noise) which is a generalization of the Binary Symmetric Channel (BSC) and the Binary Erasure Channel (BEC): The input alphabet is {0, 1} with input distribution $\{ p_0, p_1 \}$, and the output alphabet is {0, 1, ?}. When a bit (either zero or one) is transmitted, it is correctly received with probability P. When it is not received correctly, it is flipped (from 0 to 1 or 1 to 0) with probability $f(1-P)$, and it is erased (becoming "?") with probability $(1-f)(1-P)$. In order to obtain numerical answers to the questions below, assume that $p_0 = .6$, $P = .7$ and $f = 1/3$.

A. What is the entropy of the input distribution, $H(A)$?

$$H(A) = H(p_0, p_1) = H(.6, .4) = .9710$$

B. Show the *channel matrix*, P_{ij}, for the information channel, and calculate the entropy of the symbol received given the symbol transmitted, $H(B \mid A)$.

$$P_{ij} = \begin{bmatrix} .7 & .1 & .2 \\ .1 & .7 & .2 \end{bmatrix}$$

$$H(B \mid A) = H(P, f(1-P), (1-f)(1-P)) = 1.1568$$

C. Show the joint probability matrix, R_{ij}, and calculate the entropy of the joint distribution, $H(A, B)$.

$$R_{ij} = \begin{bmatrix} .42 & .06 & .12 \\ .04 & .28 & .08 \end{bmatrix}$$

$$H(A, B) = H(.42, .06, .12, .04, .28, .08) = 2.1277$$

D. What is the output distribution, $(q_0, q_1, q_?)$, and what is its entropy, $H(B)$?

$$(q_0, q_1, q_?) = \begin{bmatrix} .46 & .34 & .20 \end{bmatrix}$$

$$H(B) = H(.46,.34,.20) = 1.5089$$

E. Show the backward probability matrix, Q_{ij}, and calculate the entropy of the backward probabilities, $H(A \mid B)$.

$$Q_{ij} = \begin{bmatrix} .913 & .177 & .6 \\ .087 & .824 & .4 \end{bmatrix}$$

$$H(A \mid B) = .46\, H(.913,.087) + .34\, H(.176,.824) + .20\, H(.6,.4) = .61884$$

F. What is the mutual information between the symbols sent and the symbols received, $I(A, B)$?

$$I(A, B) = H(A) - H(A \mid B) = H(B) - H(B \mid A) = .35216$$

G. By considering other values for p_0 (and p_1) but keeping $P = .7$ and $f = 1/3$ fixed, give an estimate of the *capacity*, C, of the information channel, and indicate the approximate value of p_0 for which the channel capacity is attained. (It may be helpful to try to sketch a graph of $I(A, B)$ versus p_0 when $P = .7$ and $f = 1/3$.)

$$C = \max_{p_0} I(A, B) = .3651 \quad \text{when} \quad p_0 = p_1 = .5$$

3. Error Correcting Codes [20 marks]

Investigate three *Error Correcting Codes [n,k]*, characterized by the parameters k (where 2^k is the number of codewords), and n (the length of each codeword), for pairs [4,1], [5,2], and [6,3].

A. For each of the three *generator matrices* below composed of k codewords of length $k+3$, generate a full code of 2^k codewords.

[4,1] code	[5,2] code	[6,3] code
1111	10101 01110	010101 100110 111000

The full codes are:

[4,1] code	[5,2] code	[6,3] code
0000 1111	00000 10101 01110 11011	000000 010101 100110 110011 111000 101101 011110 001011

B. How do these three codes compare to the [7,4] Hamming code with respect to rate of transmission?

The rates (shown in this table) are all lower than for the Hamming [7,4] code.

[4,1] code	[5,2] code	[6,3] code	[7,4] Hamming
$1/4 = .250$	$2/5 = .400$	$3/6 = .500$	$4/7 = .571$

C. For each of the three codes, give the distribution of pairwise distances between all code word pairs. (This distribution consists of $2^{k-1}(2^k - 1)$ values.)

[4,1] code	[5,2] code	[6,3] code	[7,4] Hamming
$p(4) = 1$	$p(3) = .667$ $p(4) = .333$	$p(3) = .571$ $p(4) = .429$	$p(3) = .467$ $p(4) = .467$ $p(7) = .066$

D. Based on the answer to part C, comment on how these three codes compare to the [7,4] Hamming code with respect to ability to correct and/or detect various numbers of erroneous bits.

As in the Hamming [7,4} code, all these code have minimum distance of at least $d = 3$, so any single bit errors in a codeword can be *corrected*. In the first code, the minimum distance is $d = 4$, so in addition, two bit errors in a codeword can be *detected* (but not corrected).

E. In the [7,4] Hamming code, the *syndrome* can be used to identify the specific bit out of the seven that was most likely transmitted erroneously. Specify an appropriate parity check matrix, H, that will yield the possibility of syndrome decoding for the [6,3] code given above.

$$H = \begin{bmatrix} 000111 \\ 011001 \\ 101010 \end{bmatrix}$$

Note: In a code with minimum distance d, using *maximum likelihood* decoding, consider four cases for the number of single bit errors in a codeword, e:

Case 1: $e = 0$
In this case, no errors occur, so none are either dectected or corrected.

Case 2: $1 \le e \le \frac{d-1}{2}$
In this case, the errors are both detected and corrected by maximum likelihood decoding.

Case 3: $e = \frac{d}{2}$
In this case, the errors are detected, but it is at best a 50/50 guess which of two or more codewords was actually sent, so that the errors cannot be confidently corrected.

Case 4: $\frac{d-1}{2} \le e \le d - 1$
In this case, it can be detected that some number of bit errors occurred, but maximum likelihood decoding would *not* lead to the correct codeword being selected, and the message would be erroneously received.

There is general agreement that only Case 2 above describes the *correction* of errors. However, there is some disagreement over what constitutes *detection* (without *correction*) of errors. I (and several sources I have read) would say the only Case 3 involves the detection (without correction) of errors. That is, whenever the codeword received has the same Hamming distance to two or more codewords, then it is know that $d/2$ errors have occurred, but it is not possible to know what correction to take.

Other sources (including at least Jones and Jones) choose to regard *detection* of errors as covering all of cases 2, 3, and 4, ignoring the distinction of whether or not the errors can be automatically corrected.

For the sake of marking this question, both interpretations were allowed.

4. Encoding of Real Numbers [20 marks]

A sequence of n real numbers, $x_1, x_2, \cdots, x_n$, is composed of *independent* selections from the distribution $f(x)$ defined below. Consider encodings of the sequence of numbers in which each real number is encoded as a bit string of length b, and the i^{th} bit string (i.e., $i-1$ in binary) corresponds to real value r_i. With *distortion* defined as:

$$distortion = \frac{1}{n} \sum_{j=1}^{n} \min_{i} \; | x_j - r_i |$$

devise coding schemes by selecting the r_i values to minimize distortion for each of the cases below. The function $f(x)$ is a "trapezoidal" distribution of real numbers:

$$f(x) = \begin{cases} \dfrac{\beta+x}{\beta-\alpha} \left[\dfrac{1}{\beta+\alpha} \right] & -\beta \le x \le -\alpha \\ \left[\dfrac{1}{\beta+\alpha} \right] & -\alpha \le x \le +\alpha \\ \dfrac{\beta-x}{\beta-\alpha} \left[\dfrac{1}{\beta+\alpha} \right] & +\alpha \le x \le +\beta \end{cases}$$

Let $\beta = 60$ always, and let $\alpha = 0$ or 30 or 60 to form "isoceles triangular" or "trapezoidal" or "uniform" distributions, respectively.

A. Give an expression for the *differential entropy* of this distribution, and evaluate the expression for the case of $\alpha = \beta = 60$ (i.e., the uniform distribution).

$$h(x) = -\int_{x=-\beta}^{\beta} f(x) \log_2(f(x)) dx$$

For the case of $\alpha = \beta = 60$, this becomes:

$$h(x) = -\int_{x=-60}^{60} \frac{1}{120} \log_2\left(\frac{1}{120}\right) dx = \log_2(120) = 6.90689$$

B. For each of the three values of α, find boundaries (b_0, b_1, b_2) and values (r_1, r_2) so that distortion is minimized when a *one-bit* code is used for the real numbers. What is the distortion with your chosen values?

By exploiting the symmetry of the distribution about the y-axis, we can simply seek on representitive between 0 and 60, and then also take its negative.

α	b_0	r_1	b_1	r_2	b_2	D
0	-60	-17.6	0	+17.6	+60	11.71574
30	-60	-22.5	0	+22.5	+60	12.08333
60	-60	-30	0	+30	+60	15.00000

Analytically, the results for this part are:

α	r_i	D
0	$\pm 30\ (2-\sqrt{2})$	$20\ (2-\sqrt{2})$
30	$\pm 45 / 2$	$145 / 12$
60	± 30	15

C. For each of the three values of α, find boundaries $(b_0, b_1, b_2, b_3, b_4)$ and values (r_1, r_2, r_3, r_4) so that distortion is minimized when a *two-bit* code is used for the real numbers. What is the distortion with your chosen values?

Again, exploit symmetry, and find two representatives in (0, 60), and then also take their negatives.

α	r_1	r_2	r_3	r_4	D
0	-32.125	-9.375	+9.375	+32.125	6.21485
30	-34.125	-11.375	+11.375	+34.125	6.44252
60	-45	-15	+15	+45	7.50000

α	b_0	b_1	b_2	b_3	b_4
0	-60	-21.75	0	+21.75	+60
30	-60	-22.75	0	+22.75	+60
60	-60	-30	0	+30	+60

D. Assess the benefits of using *vector quantization* (with vectors of length two) rather than scalar quantization for input values from distribution f(x) with an encoding rate of one bit per input value as in part (B). Each *pair* of values from the input steam, (x_k, x_{k+1}), is encoded in two bits by associating the pair with one of four *representatives*, $((r_{x1}, r_{y1}), (r_{x2}, r_{y2}), (r_{x3}, r_{y3}), (r_{x4}, r_{y4})$. Then the contribution to distortion of this pair is:

$$\frac{1}{2} \min_{i=1,2,3,4} \left(| x_k - r_{xi} | + | x_{k+1} - r_{yi} | \right)$$

For any of the three values of α, can distortion be made lower than in part (B)? Discuss briefly to support your answer.

Again, by exploiting symmetry across the diagonal as well as vertically and horizontally, the combinatorics of the possible representative placements can be reduced substantially.

α	r_1	r_2	r_3	r_4	D
0	(-27.6, 0)	(0, -27.6)	(0, +27.6)	(+27.6, 0)	11.55042
30	(-22.5, -22.5)	(-22.5, +22.5)	(+22.5, -22.5)	(+22.5, +22.5)	12.08333
60	(-30, -30)	(-30, +30)	(+30, -30)	(+30, +30)	15.00000

By comparing the results of Parts B and D, it is apparent that, when α is 30 or 60, there is no gain in using vector rather than scalar quantization. However, when α is zero, the use of vector rather than scalar quantization reduces distortion (by at least 1%).

[Note: To produce answers for parts (B), (C) and (D), you may use any combination of analysis, heuristic search or algorithmic search. Just indicate the method(s) you use.]