Homework Set 1 EE 1473 - Digital Communication Systems

Tuesday, January 8, 2019 5:00 PM

Josh Eaton

Problem 1.

(a) Find a pair of dice. Toss the dice 100 times, recording the sum of the two numbers showing. Pretend that these numbers represent a sequence of source "messages".

9,6,12,7,6,8,5,6,4,5,7,6,9,6,7,8,6,9,7,7,5,8,11,9,5,7,8,4,6,7,12,5,10,9,8,10,4,7,6,10,5,7,5,8,6,5,8,4,9,8,8,4,7,6,6,5,7,
7,7,4,3,6,7,6,3,6,9,7,2,9,4,10,11,9,3,8,10,3,8,6,11,7,7,8,5,6,10,6,4,6,7,6,11,8,7,4,10,8,6,5

(b) Encode this string of 100 messages using an ordinary binary representation of 4 bits per message. (i.e. 610 → 01102, 1110 → 10112, etc.) What is the total number of bits?

1001011011000111011010000101011001000101011101101001011001111000011010010111011101011000101 1
100101010111110000100011001111100010110101001100010100100011101101010010101110101100001100101
1000010010011000100001000111011001100101011101110111010000110110011101100011011010010111 0010
1001010010101011100100111000101000111000011010110111011110000101011010100110010001100111 0110
1011100001110100101010000110 0101

Total length: 400 bits (4 bits per 'message')

(c) Encode the message string using the code given below. Now what is the total number of bits? What is the average number of bits per message? Compare this with the results of part (b). Which code is more efficient, and why?

1001011011000111011010000101011001000101011101101001011001111000011010010111011101011000101 1
1001010101111100001000110011111000101101010011000101001000111011010100101011101011000011001 01
1000010010011000100001000111011001100101011101110111010000110110011101100011011010010111 0010
1001010010101011100100111000101000111000011010110111011110000101011010100110010001100111 0110
1011100001110100101010000110 0101

Total length: 323 (3.23 bits per 'message')

This method is more efficient, the same information content (same 100 dice sums) is transmitted using fewer binary digits.

(d) Compute the entropy of the source, using the probabilities shown in the table. Compare this to the average number of bits per message for each of the two codes

$$H = -\sum p_j \log_2(p_j) = (1/36)\log_2(1/36) + (2/36)\log_2(2/36) + \cdots + (1/36)\log_2(1/36)$$

encoded representation $<$ $\boxed{H = 3.274}$ $<$ ordinary representation (4)

(f) See if you can decode the original messages from your classmate's bit stream. Is it possible? What property of the code makes it possible or impossible?

```
orig_msg = []
while not data == "":
    if data[0] == '0':
        if data[1] == '0':
            if len(data) > 2 and data[2] == '0':
                if len(data) > 3 and data[3] == '0':
                    if data[4] == '0':
                        orig_msg.append(2)
                        data = data[5:]
                    else:
                        orig_msg.append(12)
                        data = data[5:]
                else:
                    orig_msg.append(4)
                    data = data[4:]
            else:
                orig_msg.append(6)
                data = data[3:]
        else:
            if len(data) >2 and data[2] == '1':
                orig_msg.append(8)
                data = data[3:]
            else:
                orig_msg.append(7)
                data = data[3:]
    else:
        if data[1] == '1':
            if len(data) > 2 and data[2] == '1':
                orig_msg.append(9)
                data = data[3:]
            else:
                orig_msg.append(5)
                data = data[3:]
        else:
            if len(data) >2 and data[2] == '1':
                orig_msg.append(10)
                data = data[3:]
            else:
                if len(data) > 4 and data[4] == '0':
                    orig_msg.append(3)
                    data = data[4:]
                else:
                    orig_msg.append(11)
                    data = data[4:]
```

Here is a simple code snippet demonstrating that this can be decoded by stepping through a binary decision tree at each subsequent bit.

This is possible because of the prefix property of of this code.

- None of the code words overlaps with the prefix of any other in the table.

Problem 2.

Sum	Outcomes	Probability
2	[1,1]	1/36
3	[1,2] [2,1]	2/36

Problem 2.

Sum	Outcomes	Probability
2	[1,1]	1/36
3	[1,2] [2,1]	2/36
4	[2,2] [1,3] [3,1]	3/36
5	[4,1] [1,4] [3,2] [2,3]	4/36
6	[3,3] [4,2] [2,4] [5,1] [1,5]	5/36
7	[3,4] [4,3] [5,2] [2,5] [6,1] [1,6]	6/36
8	[4,4]	5/36
9		4/36
10		3/36
11		2/36
12		1/36

(a) Consider rolling two fair (and statistically independent) dice. What is the probability that the sum of the two dice is a prime number?

Primes ≤ 12 : $\{2,3,5,7,11\}$

$P(2) + P(3) + P(5) + P(7) + P(11)$

$1/36 + 2/36 + 4/36 + 6/36 + 2/36$

$\boxed{P(\text{prime}) = 15/36 = 0.4167}$

(b) What is the probability that the sum is a factor of 42?

Factors of $42 \leq 12$: $\{2,3,6,7\} = A$

$P(2) + P(3) + P(6) + P(7)$

$1/36 + 2/36 + 5/36 + 6/36$

$\boxed{P(A) = 14/36 = 0.3889}$

(c) What is the conditional probability that the sum is a prime number, given that the sum is a factor of 42?

$$P(A|B) = \frac{P(AB)}{P(B)}$$

$A \rightarrow$ prime $\{2,3,5,7,11\}$

$B \rightarrow$ Factor of 42 $\{2,3,6,7\}$

$A \cap B \rightarrow \{2,3,7\}$

$P(AB) = P(2) + P(3) + P(7) = 1/36 + 2/36 + 6/36 = 9/36$

$P(B) = 14/36$

$$P(A|B) = \frac{9/36}{14/36} = \boxed{\frac{9}{14} = 0.6429}$$

(d) What is the conditional probability that the sum is a prime number, given that the sum is not a factor of 42?

$A \rightarrow$ prime

$B \rightarrow$ not a factor of 42 $\quad P(B) = 1 - P(\bar{B}) = 22/36$

$A \cap B = \{5, 11\} = 6/36$

$$P(A|B) = \frac{6/36}{22/36} = \boxed{\frac{6}{22} = 0.2728}$$

(e) What is the probability that the sum is a factor of 42, given that the sum is not a prime number?

$A \rightarrow$ Factor of 42

$B \rightarrow$ not prime $\quad P(B) = 1 - P(\bar{B}) = 21/36$

$A \cap B = \{6\} = 5/36$

$$\boxed{P(A|B) = \frac{5}{21} = 0.2381}$$

$$\boxed{P(A|B) = \frac{5}{21} = 0.2381}$$

Problem 3.

$$Pr(error) = p \qquad Pr(correct) = 1 - p$$

Pr{k errors among n bits}: $\binom{n}{k}p^k(1-p)^{n-k}$; $\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$

(a) Determine

i. The probability that a byte is received correctly, without any bit errors.

$$k = 0,\ n = 8$$

$$\frac{8!}{0!\,(8)!}p^0(1-p)^8 = \boxed{(1-p)^8}$$

ii. The probability that a byte is received incorrectly, i.e. that one or more bit errors occur.

$$Pr(error) = 1 - P_r(correct)$$

$$\Rightarrow \boxed{1-(1-p)^8}$$

(b) Now suppose that a single parity bit is appended to each byte before transmission, using even parity
Determine:

i. The probability that a byte is received correctly, without any bit errors, including the parity bit. Compare to the probability of a correct byte without parity.

$$k = 0,\ n = 9$$

$$P_i = \boxed{(1-p)^9}$$

ii. The probability that a byte is received incorrectly, and this is detected by the parity system. (Hint: what will happen if there are 3, 5, 7, or 9 bit errors among the 9 bits?)

The parity system will detect any odd number of errors:

$$P_i = P(1) + P(3) + P(5) + P(7) + P(9)$$

$$\Rightarrow \frac{9!}{1!\,(9-1)!}p^1(1-p)^{9-1} + \dots + \frac{9!}{9!\,(9-9)!}p^9(1-p)^{9-9} =$$

$$P_{ii} = \boxed{9p(1-p)^8 + 84p^3(1-p)^6 + 126p^5(1-p)^4 + 36p^7(1-p)^2 + p^9}$$

iii. The probability that a byte is received incorrectly, and this is not detected by the parity system. (Hint: what will happen if there are 2, 4, 6, or 8 bit errors among the 9 bits?)

iii. The probability that a byte is received incorrectly, and this is not detected by the parity system. (Hint: what will happen if there are 2, 4, 6, or 8 bit errors among the 9 bits?)

$P_r(\text{even errors}) = 1 - P_r(\text{odd}) - P_r(\text{zero})$

$\Rightarrow \boxed{1 - P_i - P_{ii}}$

(c) Next, suppose that each 8-bit byte is divided into two words of 4 bits each, and each word is encoded using the (7,4) Hamming code.

Determine:

i. The probability that an encoded byte is decoded correctly.

14 bits/Byte decoded correctly $\Leftrightarrow k \in \{0,1\}$

$P(0) + P(1)$

$\Rightarrow \frac{14!}{14!}(1-p)^{14} + \frac{14!}{13!}p^1(1-p)^{13} = \boxed{(1-p)^{14} + 14p(1-p)^{13}} = P_c$

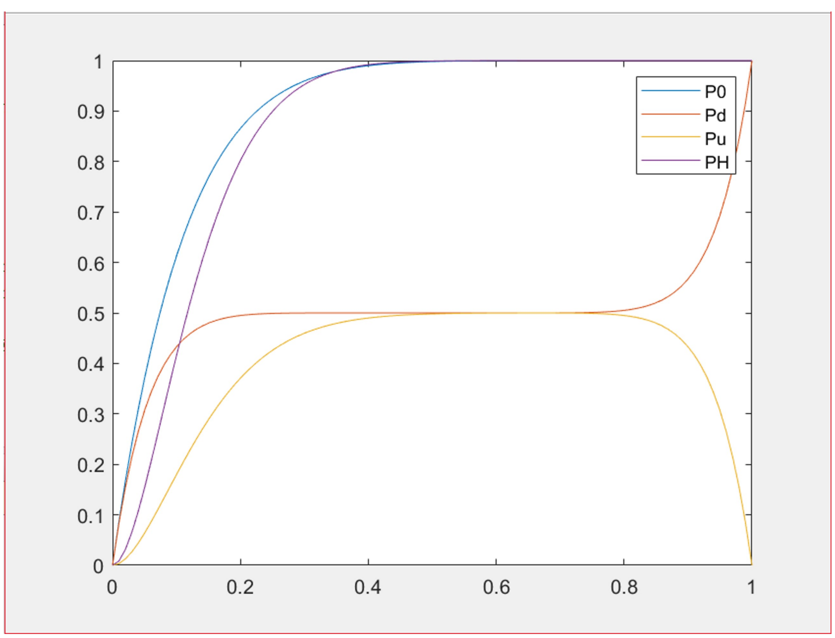
ii. The probability that an encoded byte is decoded incorrectly.

incorrectly = 1 − correctly

$\boxed{1 - \left((1-p)^{14} + 14p(1-p)^{13}\right)}$

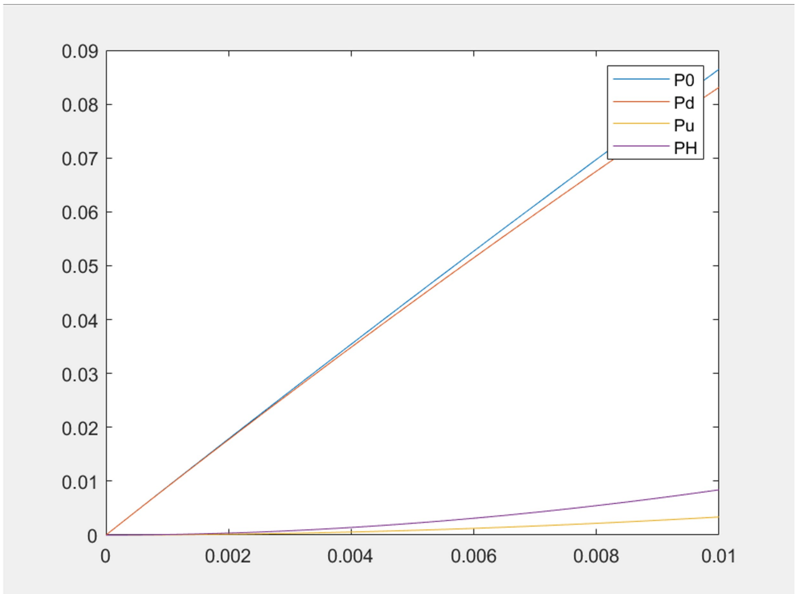
(d) Use MATLAB to plot the following four probabilities, all on the same pair of axes, for 0 ≤ p ≤ 1:

i. From part (a): P0 = the probability of a byte error with no parity.
ii. From part (b): Pd = the probability of a detected byte error with 1 parity bit per byte.
iii. From part (b): Pu = the probability of an undetected byte error with 1 parity bit per byte.
iv. From part (c): PH = the probability of a byte error with the Hamming code.



(e) Repeat part (d) for 0 ≤ p ≤ 0.01.

(e) Repeat part (d) for 0 ≤ p ≤ 0.01.



Using your plot, compare these four probabilities, and what they indicate about the utility of the parity strategies. Explain why these results are expected based on the expressions from parts (a) - (c).

These plots show that the parity and coding strategies that add redundancy will increase the probability of error detection and correction. Even at higher BER, the parity has a 50/50 shot of detecting errors. The second plot shows that the probability of a byte error with these schemes is very low (<1%) for very low BER, while the probability of error without any parity increases much more rapidly as BER increases.

Problem 4.

Event	Probability
$S_3 = \{\text{Sleeps for 3 hours}\}$	0.3
$S_5 = \{\text{Sleeps for 5 hours}\}$	0.4
$S_8 = \{\text{Sleeps for 8 hours}\}$	0.2
$S_{12} = \{\text{Sleeps for 12 hours}\}$	0.1

$$\Pr(Score > Mean) = \frac{-3n^2 + 57n - 138}{200},$$

(a) One day, Ima takes the exam and scores above the mean. What is the most likely number of hours for Ima to have slept the night before?

$$\Pr(S_3) = 0.3\left(\frac{-27 + 57(3) - 138}{200}\right) = 0.009$$

$$\boxed{\Pr(S_5) = 0.144}$$

$$\Pr(S_8) = 0.126$$

$$\Pr(S_{12}) = 0.057$$

5 hours is most likely

(b) Ima has a log book, and on each day when the exam score is above the mean, Ima records the number of hours of sleep from the previous night in the book. If you were to compute the average of all of the numbers in the log book, what value would you expect to obtain?

$$\langle S \rangle = \frac{.009(3) + 0.144(5) + 0.126(8) + 0.057(12)}{.009 + 0.199 + 0.126 + 0.057}$$

$$\boxed{\Rightarrow 7.26 \text{ hours}}$$

5. A natural gas extraction site uses electronic sensors to determine the presence of a toxic chemical in the atmosphere. Each sensor takes a measurement once every 500 milliseconds, and each measurement can take 201 different values. There are 4816 sensors distributed over the site, and the measurements for all sensors are communicated as binary data to a central computer.

(a) How many bits are required to represent each measurement from one sensor? Assume that a fixed-length code is used.

$\boxed{8 \text{ bits}}$ $\quad \lceil \log_2(201) \rceil = 8$

(b) What is the information content of each measurement from one sensor if each of the 201 possible values is equally likely to occur?

$H = \log_2(N)$ if all equally likely $\quad N = 201$

$\boxed{H = 7.651}$

(c) The measurement data from all sensors are combined into a single bit stream for transmission to an off-site monitoring station. These data are then divided into 7-bit blocks, and one parity bit is appended to each block to allow for detection of bit errors that may occur during transmission. Determine the bit rate of the combined bit stream with parity bits.

$[(4816)(8)]\ 8/7$ bits / measurement $\quad 500\text{ ms} = 2\text{ Hz}$

$$R = \frac{(4816)(64)2}{7} = \boxed{88064 \text{ bits/sec}}$$

(d) The data from part (c) are to be communicated over a noisy channel, for which the bandwidth could be 100 Hz, 100 kHz or 100 MHz. For each bandwidth, determine the minimum SNR in dB such that it is theoretically possible to communicate the data through the channel without error. Interpret your result: what is the best choice of channel bandwidth?

Shannon

$$C = B_w \log_2(1 + SNR)$$

$$\Rightarrow SNR = 2^{C/B} - 1$$

Pretty much impossible @ 100 Hz

$\boxed{\text{100 kHz BW is the best choice}}$

@ 100 Hz

$SNR = 2^{880} - 1$ w/w

$10 \log_{10}(2^{880} - 1) = \boxed{2651 \text{ dB}}$

@ 100 kHz

$SNR = 2^{.88064} - 1 \Rightarrow \boxed{-0.75 \text{ dB}}$

@ 100 MHz

$SNR = 2^{.00088064} - 1 \Rightarrow \boxed{-32.1 \text{ dB}}$