Prof. Dr. Rudolf Mathar, Dr.-Ing. Gholamreza Alirezaei, Emilio Balda, Vimal Radhakrishnan

Tutorial 7
- Proposed Solution -

Monday, January 21, 2019

Solution of Problem 1

AEP and source coding

A discrete memoryless source emits a sequence of statistically independent binary digits with probabilities $p(1) = 0.005$ and $p(0) = 0.995$. The digits are taken 100 at a time and a binary codeword is provided for every sequence of 100 digits containing three or fewer ones.

a) Assuming that all codewords are the same length, find the minimum length required to provide codewords for all sequences with three or fewer ones.

The number of 100-bit binary sequences with three or fewer ones is

$$\binom{100}{0} + \binom{100}{1} + \binom{100}{2} + \binom{100}{3} = 1 + 100 + 4950 + 161700 = 166751\,.$$

The required codeword length is $\lceil \log_2 166751 \rceil = 18$. Note that $H(0.005) = 0.0454$, so 18 is quite a bit larger than the 4.5 bits of entropy.

b) Calculate the probability of observing a source sequence for which no codeword has been assigned.

$$\sum_{i=0}^{3} \binom{100}{i} (0.005)^i (0.995)^{100-i} = 0.60577 + 0.30441 + 0.7572 + 0.01243 = 0.99833\,.$$

Thus the probability that the sequence that is generated cannot be encoded is $1 - 0.99833 = 0.00167$.

c) Use Chebyshev's inequality to bound the probability of observing a source sequence for which no codeword has been assigned. Compare this bound with the actual probability computed in part **b)**.

In the case of a random variable S_n that is the sum of n i.i.d. random variables $X_1, X_2, \ldots, X_n$, Chebyshev's inequality states that

$$P(|S_n - n\mu| \geq \epsilon) \leq \frac{n\sigma^2}{\epsilon^2}\,,$$

where μ and σ^2 are the mean and variance of X_i (thus $n\mu$ and $n\sigma^2$ are the mean and variance of S_n). In this problem $n = 100$, $\mu = 0.005$, and $\sigma^2 = 0.005 \times 0.995$. Note that $S_{100} \geq 4$ iff $|S_{100} - 100(0.005)| \geq 3.5$, so we could choose $\epsilon = 3.5$. Then

$$P(S_{100} \geq 4) \leq \frac{100(0.005)(0.995)}{(3.5)^2} \approx 0.04061\,.$$

This bound is much larger than the actual probability 0.00167.

Solution of Problem 2

Shannon codes and Huffman codes

Consider a random variable X which takes on four values with probabilities $(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12})$.

a) Construct a Huffman code for this random variable.

Applying the Huffman algorithm gives us the following table

Code	Symbol	Probability			
0	2	1/3	1/3	1/3	1
11	2	1/3	1/3	2/3	
101	3	1/4	1/3		
100	4	1/12			

which gives codeword lengths of $1, 2, 3, 3$ for the different codewords.

b) Show that there exist two different sets of optimal lengths for the codewords, namely, show that codeword length assignments $(1, 2, 3, 3)$ and $(2, 2, 2, 2)$ are both optimal.

Both set of lengths $1, 2, 3, 3$ and $2, 2, 2, 2$ satisfy the Kraft inequality, and they both achieve the same expected length (2 bits) for the above distribution. Therefore they are both optimal.

c) Conclude that there are optimal codes with codeword lengths for some symbols that exceed the Shannon code length $\lceil \log \frac{1}{p(x)} \rceil$.

The symbol with probability $1/4$ has an Huffman code of length 3, which is greater than $\lceil \log \frac{1}{p} \rceil$. Thus the Huffman code for a particular symbol may be longer than the Shannon code for that symbol. But on the average, the Huffman code cannot be longer than the Shannon code.

Solution of Problem 3

Twenty Questions

Player A chooses some object in the universe, and player B attempts to identify the object with a series of yes-no questions. Suppose that player B is clever enough to use the code achieving the minimal expected length with respect to player A's distribution. We observe that player B requires an average of 38.5 questions to determine the object. Find a rough lower bound to the number of objects in the universe.

Solution:

$$37.5 = L^* - 1 < H(X) \leq log|\mathcal{X}|$$

and hence number of objects in the universe $> 2^{37.5} = 1.94 \times 10^{11}$.

Solution of Problem 4

Bad Wine

One is given 6 bottles of wine. It is known that precisely one bottle has gone bad (tastes terrible). From inspection of the bottles it is determined that the probability p_i that the i-th bottle is bad is given by $(p_1, p_2, ..., p_6) = (\frac{8}{23}, \frac{6}{23}, \frac{4}{23}, \frac{2}{23}, \frac{2}{23}, \frac{1}{23})$. Tasting will determine the bad wine. Suppose you taste the wines one at a time. Choose the order of tasting to minimize the expected number of tastings required to determine the bad bottle. Remember, if the first 5 wines pass the test you don't have to taste the last.

a) What is the expected number of tastings required? If we taste one bottle at a time, to minimize the expected number of tastings the order of tasting should be from the most likely wine to be bad to the least. The expected number of tastings required is

$$\begin{aligned}\sum_{i=1}^{6} p_i l_i &= 1 \times \frac{8}{23} + 2 \times \frac{6}{23} + 3 \times \frac{4}{23} + 4 \times \frac{2}{23} + 5 \times \frac{2}{23} + 5 \times \frac{1}{23} \\ &= \frac{55}{23} \\ &= 2.39\end{aligned}$$

b) Which bottle should be tasted first? The one with probability $\frac{8}{23}$.

Now you get smart. For the first sample, you mix some of the wines in a fresh glass and sample the mixture. You proceed, mixing and tasting, stopping when the bad bottle has been determined.

c) What is the minimum expected number of tastings required to determine the bad wine?

The idea is to use Huffman coding. With Huffman coding, we get codeword lengths as $(2, 2, 2, 3, 4, 4)$. The expected number of tastings required is

$$\begin{aligned}\sum_{i=1}^{6} p_i l_i &= 2 \times \frac{8}{23} + 2 \times \frac{6}{23} + 2 \times \frac{4}{23} + 3 \times \frac{2}{23} + 4 \times \frac{2}{23} + 4 \times \frac{1}{23} \\ &= \frac{55}{23} \\ &= 2.35\end{aligned}$$

d) What mixture should be tasted first?

The mixture of the first and second bottles.

Solution of Problem 5

Horse Race

Three horses run a race. A gambler offers 3-for-1 odds on each of the horses. These are fair odds under the assumption that all horses are equally likely to win the race. The true win probabilities are known to be

$$\mathbf{p} = (\frac{1}{2}, \frac{1}{4}, \frac{1}{4}).$$

Let $\mathbf{b} = (b_1, b_2, b_3)$, $b_i \geq 0$, $\sum b_i = 1$, be the amount invested on each of the horses. The expected log wealth is thus

$$W(\mathbf{b}) = \sum_{i=1}^{3} p_i \log 3b_i \,.$$

a) Maximize this over $\mathbf{b}$ to find $\mathbf{b}^*$ and W^*. Thus the wealth achieved in repeated horse races should grow to infinity like 2^{nW^*} with probability one.

The doubling rate

$$\begin{aligned} W(\mathbf{b}) &= \sum_i p_i \log 3b_i \\ &= \sum_i p_i \log 3b_i + \sum_i p_i \log p_i - \sum_i p_i \log \frac{p_i}{b_i} \\ &= \log 3 - H(\mathbf{p}) - D(\mathbf{p}\|\mathbf{b}) \\ &\leq \log 3 - H(\mathbf{p}) = 0.085 \end{aligned}$$

with equality iff $D(\mathbf{p}\|\mathbf{b}) = 0$, that is $\mathbf{p} = \mathbf{b}$. Hence $\mathbf{b}^* = \mathbf{p} = (\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$ and $W^* = 0.085$.

By the strong law of large numbers,

$$\begin{aligned} S_n &= \prod_j 3b(X_j) \\ &= 2^{n(\frac{1}{n}\sum_j \log 3b(X_j))} \\ &\to 2^{n(E \log 3b(X))} \\ &= 2^{nW(\mathbf{b})} \end{aligned}$$

When $\mathbf{b} = \mathbf{b}^*$, $W(\mathbf{b}) = W^*$ and $S_n = 2^{nW^*} = 2^{0.085n} = (1.06)^n$.

b) Show that if instead we put all of our money on horse 1, the most likely winner, we will eventually go broke with probability one.

If we put all the money on the first horse, then the probability that we do not go broke in n races is $(\frac{1}{2})^n$. Since this probability goes to zero with n the probability of the set of outcomes where we do not ever go broke is zero, and we will go broke with probability 1.

Alternatively, if $\mathbf{b} = (1, 0, 0)$, then $W(\mathbf{b}) = -\infty$ and

$$S_n \to 2^{nW} = 0 \quad \text{w.p.1}$$

by the strong law of large numbers.