Arhitectura Sistemelor de Calcul October, 2021

Seminar 0x01

Cristian Rusu, Bogdan Macovei

1 Seminar objectives

In this seminar, we will solve problems involving:

- the concept of information and Shannon entropy;
- variable-length data encoding;
- error detection and correction.

2 Exercises

1. We have a deck of playing cards (52 cards). We take the cards out of the box for the first time. How much information do we have about the cards at this point? We shuffle the cards randomly (by the way, how can we do this efficiently using an algorithm?). How much information do we now have in the deck of cards? (Also use Stirling's approximation to calculate the result.)
2. An urn contains 5 red disks and 3 blue disks. We are told that someone draws a disk from the urn and that it is blue. Tasks:
   (a) how much information do we gain from this observation?
   (b) what was the entropy of the urn before the draw? What is the entropy of the urn after the draw?
   (c) continue calculating the entropy, assuming that we draw all the blue disks one by one;
   (d) repeat the previous task for the red disks (starting with the original urn, assume that you draw each red disk in turn and calculate the entropy at each step);
3. You are given 12 disks. Eleven of them have the same weight, and one is either lighter or heavier than all the others. You have a balance scale. Find the different disk using the minimum number of weighings.
4. Lecture 0x01 contains a slide about A. Turing. Calculate the entropy of the text on the slide (at the character/symbol level, without grouping): first with and then without diacritics. Compare the two calculated values.
5. Let x be a natural number represented using N bits. How much information have we gained if:
   (a) we are told that x has exactly two "1" values in its binary representation;
   (b) we are told that x has exactly $N/2$ "1" values in its binary representation;
   (c) we are told that x has a contiguous sequence of $N/4$ "1" bits in its binary representation (the remaining bits are "0");
   (d) we are told that x has its MSB set to "1";
   (e) we are told that x is odd;
   (f) we are told that x is a power of 2;
   (g) we are told that x has the first $N/2$ bits of its binary representation set to "0";
   (h) we are told that x is a prime number (only a rough estimate is possible here);

(i) we are told that x has an even number of bits set to "1" in its binary representation;

(j) we are told that $x = 42$.

6. A random variable X has the distribution given in the following table:

event/symbol x	A	B	C	D	E
Probability$(X = x)$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{12}$

Tasks:

(a) check that the given probabilities are valid (the distribution is consistent);

(b) calculate the entropy $H(X)$;

(c) construct the Huffman code for X. Check the efficiency of this code and compare it with the entropy;

(d) encode the message ABBACED using the Huffman code you have found;

(e) group the events/symbols in pairs (we will call this variable X^2). Tasks:

i. how many new events/symbols are there now? (What is the size of the variable X^2?);

ii. calculate $H(X^2)$. How does it relate to $H(X)$?

iii. construct the Huffman code for X^2. Check the efficiency of this code and compare it with $H(X)$.

7. Consider the following message block:

D_{00}	D_{01}	D_{02}	$P_{0\ell}$
D_{10}	D_{11}	D_{12}	$P_{1\ell}$
D_{20}	D_{21}	D_{22}	$P_{2\ell}$
P_{c0}	P_{c1}	P_{c2}	$P_{c\ell}$

The entries D_{ij} are data (so we have 9 bits), and the entries P_{ij} are parity bits (so we have 7 redundant bits): for each row, for each column, and for the entire message. The parity bits are "1" if there is an even number of "1" values in the row/column/message. Tasks:

(a) suppose that bit D_{01} changes. How many parity bits change?

(b) what is the minimum Hamming distance of the code above?

(c) how many errors can we detect? How many errors can we correct?

(d) are the following messages valid or corrupted (if corrupted, in what way)?

1	0	1	1
0	1	1	1
1	1	0	1
1	1	1	1

1	0	1	1
1	1	0	1
0	1	1	1
1	0	1	1

0	1	0	1
0	0	1	0
1	1	0	1
1	1	0	0

0	1	0	0
1	0	1	1
0	1	1	1
0	1	1	1

1	0	1	1
0	0	1	1
1	1	1	1
1	1	1	1

8. The ISBN-10 coding system for books uses 9 digits plus a check digit to uniquely identify each book. The check digit has two purposes: 1) it detects whether one of the 10 digits has been written incorrectly, and 2) it detects whether two consecutive digits have been swapped. These two errors are the most common when encoding this type of information for books. The encoding is as follows:

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, p, \tag{1}$$

where $x_1, \ldots, x_9 \in \{0, \ldots, 9\}$ are given and p (which would be x_{10}) is calculated so that:

$$\sum_{i=1}^{9} (11 - i)x_i + p = 0 \bmod 11. \tag{2}$$

Tasks:

(a) calculate the check digit p for the ISBN-10 code: 0-306-40615p;

(b) show that any ISBN code can detect any single-digit error;

(c) show that any ISBN code can detect any swap of two consecutive digits in the code;

(d) what is the minimum distance of this code?

(e) what happens if we change the rule for the check digit so that $\sum_{i=1}^{9} x_i + p = 0 \bmod 11$? Can we detect/correct errors equally well?

(f) investigate the ISBN-13 code yourselves[1].

9. Consider the use of letters in English. According to Wikipedia[2], the 26 letters ("a" to "z") occur with the following frequencies: 8.20%, 1.50%, 2.80%, 4.30%, 13%, 2.20%, 2%, 6.10%, 7%, 0.15%, 0.77%, 4%, 2.40%, 6.70%, 7.50%, 1.90%, 0.10%, 6%, 6.30%, 9.10%, 2.80%, 0.98%, 2.40%, 0.15%, 2% and 0.07%.

   Tasks:

   (a) how many bits would we use to encode the given alphabet using fixed-length binary strings?

   (b) calculate the entropy of this alphabet;

   (c) what would the entropy be if all the letters had the same frequency? How does it relate to the entropy calculated for the given alphabet?

   (d) you may remove one letter from the alphabet above. Choose the letter to remove so as to minimize/maximize the entropy of the new alphabet (you may write a program to check this, but you must understand how to update all the probabilities when you remove a letter);

   (e) you can find the Morse code in Appendix 1. Assume that a dot occupies one bit and a dash occupies two bits of memory. Tasks:

   - what relationship do you observe between the probabilities of the letters and the lengths of their Morse code representations?
   - decode the message: .- ... -.-. . – .. ... - —
   - are messages encoded in Morse code uniquely decodable? Can you decode the message above without the spaces after each letter?
   - calculate the average length for encoding the alphabet above in Morse code and compare it with the entropy calculated earlier.

10. We have an alphabet with two symbols: symbol "A" is encoded using a single bit and has probability of occurrence p, and symbol "B" is encoded using two bits and has probability of occurrence $1 - p$ (of course).

    Tasks:

    (a) calculate the average length of messages using the alphabet above;

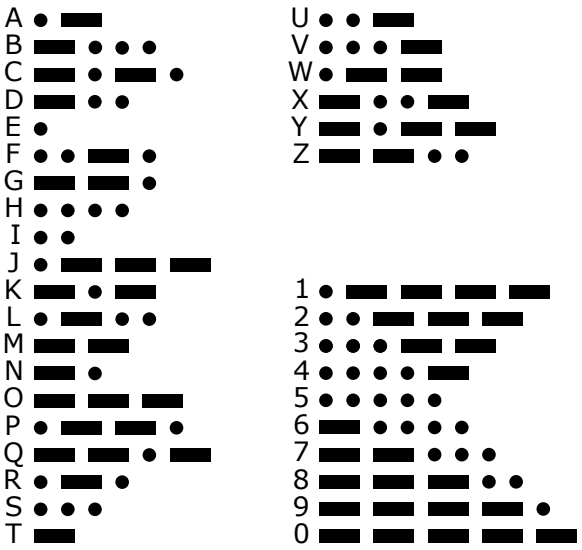
    (b) we define the entropy rate as the entropy divided by the average message length. Find p for which the entropy rate is minimized/maximized.

[1] `https://en.wikipedia.org/wiki/International_Standard_Book_Number`
[2] `https://en.wikipedia.org/wiki/Letter_frequency`

11. Suppose we have an alphabet with N symbols. In the lecture, we discussed the fact that entropy is maximized under "maximum uncertainty", meaning that all symbols in the alphabet have the same probability of occurrence (that is, $p_i = 1/N$ for $i = 1, \ldots, N$ – this is called the uniform distribution). Prove this statement mathematically.

12. We have an alphabet with N symbols x_i, each with probability of occurrence p_i. Assume that the symbols are sorted in decreasing order of their probabilities of occurrence: that is, $p_1 \geq p_2 \geq \ldots \geq p_N$. These symbols are encoded as follows: symbol x_i is encoded with the binary string $00 \cdots 001$ (the bit 0 repeated $i - 1$ times, followed by the bit 1). Are the values p_i relevant in this case? What is the average length of a message encoded this way? Compare this value with the entropy. What is the main advantage of this encoding?

Appendix 1



Morse code (source: Wikipedia).