Information Theory **DCC/ICEx/UFMG**
Prof. Mário S. Alvim **2017/02**

SOLUTION OF HOMEWORK

Stream Codes
(MacKay - Chapter 6)

Necessary reading for this assignment:

- *Information Theory, Inference, and Learning Algorithms* (MacKay):
  - Chapter 6.1: *The guessing game*
  - Chapter 6.2: *Arithmetic codes*
  - Chapter 6.4: *Lempel-Ziv coding*
  - Chapter 6.6: *Summary*

Note: The exercises are labeled according to their level of difficulty: `[Easy]`, `[Medium]` or `[Hard]`. This labeling, however, is subjective: different people may disagree on the perceived level of difficulty of any given exercise. Don't be discouraged when facing a hard exercise, you may find a solution that is simpler than the one the instructor had in mind!

Exercises.

1. The following exercises regard stream codes.

   (a) (MacKay 6.5) `[Medium]`

   Instructor's solution: Given in the textbook.

   (b) (MacKay 6.6) `[Medium]`

   Instructor's solution: Given in the textbook.
   The following may help. To decode the string 00101011101100100100011010101000011 we use the table below that tells us to break the string into 0/01/010/111/0110/0100/1000/1101/01010/00011 (corresponding to the pairs of pointer, bit).

$s(n)$ using 1 bit	0	1	-	-	-	-	-	-	-	-	-
$s(n)$ using 2 bits	00	01	10	11	-	-	-	-	-	-	-
$s(n)$ using 3 bits	000	001	010	011	100	101	110	111	-	-	-
$s(n)$ using 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010
(pointer, bit)	-	$(-,0)$	$(0,1)$	$(01,0)$	$(11,1)$	$(011,0)$	$(010,0)$	$(100,0)$	$(110,1)$	$(0101,0)$	$(0001,1)$
source substrings	λ	0	1	00	001	000	10	0010	101	0000	01

   This gives us the decoding 0100001000100010101000001.

2. **(The entropy of a compressed file)** This exercise regards compression algorithms in general.

   An information-theory student wants to check whether she can beat Shannon's compression limit of $H(X)$ bits per symbol for an optimal code C applied to a source ensemble $X = (x, \mathcal{A}_X, \mathcal{P}_X)$.

   She envisions a lossless compression method in two steps as follows:

Step 1. Apply an optimal lossless code C to the source X, obtaining a compressed binary file Y.

Step 2. Consider the new file Y as a new source ensemble, in which each symbol of Y is a bit. Apply a new optimal lossless code C' to compress Y into a new binary file Z.

Recalling Shannon's Source Coding Theorem, the student makes the following claims about her newly proposed compressing method:

Claim 1: Since code C is optimal for the source X, file Y uses approximately $H(X)$ bits to represent each symbol of X.

Claim 2: Since code C' is optimal for the source Y, file Z uses approximately $H(Y)$ bits to represent each bit of Y (note that each symbol of Y is itself a bit).

Claim 3: File Z represents each symbol of X using approximately $H(X)H(Y)$ bits.

(a) `[Easy]` Discuss whether or not each of the student's three claims are correct.

Instructor's solution: Claims 1 and 2 are direct consequences of Shannon's Source Coding Theorem. [1]

Claim 3 follows immediately from Claims 1 and 2: since each symbol of X is represented in Y using $H(X)$ bits, and each bit of Y (note that each symbol of Y is itself a bit) is represented in Z using $H(Y)$ bits, we have that each symbol of X is represented using $H(X)H(Y)$ bits in Z.

(b) `[Medium]` What can we say about the size of file Y in comparison to the size of file Z? Is Z gonna be smaller, larger, or of equal size to Y? (Hint: Recall that Shannon's Source Coding Theorem must be valid for the compression from X to Z.)

Instructor's solution: Shannon's Source Coding Theorem must be valid for the compression from X to Z, no matter whether the algorithm runs in one step (compressing X directly into X), or in two steps (first compressing X into Y, then compressing Y into Z), or in whatever number of steps. Hence, Z cannot be smaller than Y, otherwise we would have a compression method that beats Shannon limit by producing an output file using less than $H(X)$ bits per symbol of the source file. That means that the size of Z must be approximately the same size of Y.

(c) `[Medium]` Using your answers to the previous items, what would be an accurate estimation for the value of $H(Y)$?

Instructor's solution: From Claim 3 we know that Z uses $H(X)H(Y)$ bits per symbol of X, and from (2b) we know that Z uses $H(X)$ bits per symbol of X. Hence, we must conclude that $H(Y) \approx 1$.

(d) `[Medium]` Using your answers to the previous items, what can the student conclude about the frequency of bits 0 and 1 in any optimally compressed file? How does that relate to the title of this assignment: "*Compression and redundancy*"?

Instructor's solution: From (2c) we know that $H(Y) \approx 1$ bit per bit, which means that bits 0 and 1 must occur approximately with equal frequency in Y.

This is a general characteristic of optimal compressors: they remove all redundancy from the source file, producing a compressed file in which the frequency of 0's and 1's is uniform.

[1] Recall that Shannon's Source Coding Theorem states that any compression algorithm C for an ensemble X must satisfy $L(C, X) \geq H(X)$ (where $L(C, X)$ is the average length per symbol of the code C for ensemble X), and that for a long enough sequence of symbols of X, an optimal code C uses only $L(C, X) \approx H(X)$ bits per symbol of X.