Advanced Communication Systems, ELEN90051
Workshop week 3 (=*week of 12 March*): Source Coding

Department of Electrical and Electronic Engineering
The University of Melbourne

updated February 28, 2018

1 Introduction

In this workshop you will focus on the area of Source Coding and practice various **data compression techniques**. There exist four types of data compression techniques, here we will call them type I, II, III and IV:

- (**Type I**) fixed-to-fixed source coding
- (**Type II**) fixed-to-variable source coding
- (**Type III**) variable-to-fixed source coding
- (**Type IV**) variable-to-variable source coding

You will get detailed exposure to several of these different techniques and you will then be able to compare them. The main objective is to enable you to make the theory your own by thinking critically and comparing techniques. For this, you will work on each of the three lossless compression techniques, namely Huffman coding, LZ coding and arithmetic coding. For your group report you are also asked for a comparison of the three compression techniques.

2 Organisation

You are expected to **be prepared before attending the workshop session**. For this, an **individual pre-workshop report answering all questions in section 3 as well as Question 4.1 (a)-(e)**, worth 18 marks, is to be submitted before the start of the workshop session. You are also asked to write a group project report, answering all questions of section 4-6 (except question 4.1(a)-(e) which was done in the pre-workshop report). This report is worth 55 marks and should be submitted before the start of your next workshop of week 4. Please read the document "Rules on workshops & report submission" for more information, see LMS "Workshops" .

3 Uniformly distributed sources (10 marks)

1. (2 marks) What does the expression "uniformly distributed DMS" mean? Explain each of the terms. What is the entropy of a DMS?
2. Consider a DMS X that takes values in an alphabet of five letters T, H, D, E, A, each occurring with probability 0.2. Suppose that X generates the following sequence x of 48 letters (HEAD repeatedly 12 times):

   $$HEADHEADHEADHEADHEADHEADHEADHEADHEADHEADHEADHEAD$$

   (a) (1/2 marks) Compute the entropy $H(X)$.

   (b) (1/2 marks) Encode the above sequence x into a sequence of bits by encoding each letter separately. What is the length of your binary sequence?

   (c) (1 mark) Repeat (b) but now encode two letters at a time (we call this *second extension* encoding).

   (d) (1 mark) Repeat (b) but now encode three letters at a time (we call this *third extension* encoding).

   (e) (1 mark) Recall that there are 4 types of compression, namely Type I, Type II, Type III and Type IV as explained in section 1. What type of data compression did you use in (b)-(d)? Reflect on the outcomes of (d), relating your outcomes to the entropy $H(X)$. Is your encoding optimal?
3. (4 marks) Consider a DMS Y that takes values in an alphabet of four letters T, H, E, A, each occurring with equal probabability. Suppose that Y generates the following sequence of 48 letters (HEAT repeatedly 12 times):

   $$HEATHEATHEATHEATHEATHEATHEATHEATHEATHEATHEATHEAT$$

   Repeat (a)-(e) for this case.

4 Huffman coding

1. Consider a DMS X that takes values in an alphabet of eight letters T, H, E, A, D, R, S, Q, with probabilities 0.25, 0.20, 0.15, 0.12, 0.10, 0.08, 0.05 and 0.05, respectively.

   (a) (1 mark) What is the entropy $H(X)$ of X?

   (b) (3 marks) Design a binary Huffman code **for this source**, encoding one letter at a time (by hand, no MATLAB needed).

   (c) (1 mark) Recall that there are 4 types of compression, namely Type I, Type II, Type III and Type IV as explained in section 1. What type is your Huffman coding?

   (d) (1 mark) Determine the average number R of bits per source letter when using your Huffman code.

(e) (2 marks) Compare R with $H(X)$. Is your Huffman code optimal? If yes, why? If no, does there exist another Huffman-based code that gives a better compression rate?

2. Consider a binary DMS X, with probabilities $1 - 10^{-2}$ and 10^{-2}.

   (a) (1 mark) What is the entropy $H(X)$ of X?

   (b) (2 marks) Derive a second extension binary Huffman code **for this source**. Determine the average number R of bits per source letter. Is your code optimal?

   (c) (11 marks) Generate a binary sequence x_1 of length 1000 in MATLAB that is output by this source (use the specified probabilities to generate the bits). **Write MATLAB programs** to encode and decode your sequence, using your Huffman coder of part (b). Determine the number $R(x_1)$ of bits per source letter and compare with (b). Do you get compression, expansion or neither? Can you think of a more efficient way to compress your binary sequence?

   (d) (10 marks) Repeat (a)-(c) for a binary DMS Y, with probabilities 0.7 and 0.3.

5 Lempel-Ziv coding & arithmetic coding

1. In this question we focus on LZ78, which is explained in the textbook.

   (a) (2 marks) Explain in your own words how LZ78 works. Use any reference material you can find (the textbook, other books, web, scientific papers, survey papers...); make sure that you refer to your references in the text, for this include a References section. How to do this? Read "ieeecitationref.pdf", see LMS-this workshop. Do not use word-for-word citations (paraphrasing in your own words is ok).

   (b) (1 mark) Recall that there are 4 types of compression, namely Type I, Type II, Type III and Type IV as explained in section 1. What type is LZ78?

   (c) (3 marks) By using LZ78, compress the following binary source sequence (00010010 repeatedly):

   000100100001001000010010000100100001001000010010000100100001001000010010000100

   (d) (7 marks) Write a MATLAB program for an LZ78 encoder; also write a MATLAB program for the corresponding decoder. Make sure that you validate your programs, using the sequence from part (c).

2. (11 marks) LZ coding is a "stream coding" method. Another type of stream coding is "arithmetic coding". It is widely used in a range of applications, see the LMS additional material "WangOstermannZhang pages 234-241". Another good description that you may find helpful is in the book by David McKay "Information Theory, Inference, and Learning Algorithms", 2005 edition:
   `http://www.inference.phy.cam.ac.uk/mackay/itila/book.html`

(a) (1 mark) Recall that there are 4 types of compression, namely Type I, Type II, Type III and Type IV as explained in section 1. What type is arithmetic coding?

(b) (5 marks) Explain in your own words how arithmetic coding works. For this, use any reference material you can find (the textbook, other books, web, scientific papers, survey papers...); make sure that you refer to your references in the text, for this include a References section. How to do this? Read "ieeecitationref.pdf", see LMS-this workshop. Do not use word-for-word citations (paraphrasing in your own words is ok).

(c) (5 marks) Consider the Example 8.4 in "WangOstermannZhang pages 234-241". Explain in detail how "a a c b" is encoded by an arithmetic encoder.

6 Reflection and comparison (12 marks)

Explain the differences between LZ, Huffman and arithmetic coding, using any reference material you can find (the textbook, other books, web, scientific papers, survey papers...; make sure that you refer to your references in the text, for this include a References section. How to do this? Read "ieeecitationref.pdf", see LMS-this workshop. Do not use word-for-word citations (paraphrasing in your own words is ok). Particularly pay attention to:

1. In which situation would one type of coding be preferred over another? (6 comparisons)
2. Where is LZ coding used in practice?
3. Where is arithmetic coding used in practice?
4. Where is Huffman coding used in practice?