

Handin 1: Source Coding

Information Theory

Willard Thór Rafnsson

due 26th January 2009

1 Short

Exercise 1 (Unique Decodability) We have seen two definitions of unique decodability.

Definition 1 (Unique Decodability)

A code \mathcal{C} is *uniquely decodable* iff $\mathcal{C}^* : S^* \rightarrow T^*$ is an injection. \square

Definition 2 (Unique Decodability (Simplified))

Assuming $w_i \neq w_j$ for $i \neq j$, a code \mathcal{C} is *uniquely decodable* iff any code word sequence \mathbf{w} is uniquely factorisable: $v_1 \cdots v_j = w_1 \cdots w_k \implies k = j \wedge v_i = w_i, \forall i. \square$

Prove that they are equivalent.

Exercise 2 (Algorithm Simulation)

- a) Compute \mathcal{C}_∞ for the codes $\{02, 11, 110, 21, 22, 20\}$, $\{02, 10, 101, 21, 22, 20\}$, and $\{02, 10, 110, 101, 22, 20\}$. Which of these codes are uniquely decodable?
- b) Consider word length $\mathbf{l} = 1, 2, 3, 3$, and $\mathbf{l}' = 1, 1, 2, 2, 3, 3, 3, 3$. Is there a binary, ternary, or quaternary code for \mathbf{l} and \mathbf{l}' ? If there is, construct the one with the least arity r .
- c) Let $P = (\frac{1}{2}, \frac{1}{3}, \frac{1}{12}, \frac{1}{12})$. Construct a (binary) Huffman code for P . How many such codes are there (is the one you constructed unique)?

Exercise 3 (Bad Codes) Prove for each of these binary source codes whether it can or cannot be a (binary) Huffman code, for any probability distribution.

- a) 0, 10, 111, 101
- b) 00, 010, 011, 10, 110
- c) 1, 000, 001, 010, 011

2 Long

Exercise 4 (Equality in Kraft) Let \mathcal{C} be an r -ary code with word-lengths l_1, \dots, l_q . Show that any two of the following conditions imply the third:

1. \mathcal{C} is instantaneous;
2. \mathcal{C} is *exhaustive*; a code is exhaustive if every sufficiently long sequence of code-symbols begins with a code-word¹.
3. $\sum r^{-l_i} = 1$.

Show that no one of these conditions imply any other.

Exercise 5 (Huffman block) Consider this slight modification to the Huffman Code generation algorithm:

Instead of renumbering P at the start of each recursive step, assume that P is a decreasing sequence², and before each recursive call, move the amalgamated p' as far left in P' as possible s.t. P' is a decreasing sequence.

When does this implementation produce a blockcode? Prove your claim.

3 Extra

Exercise 6 (20 questions) 2 dice are cast. Alice records its value, v . Device a scheme, using Information Theory, where Bob can guess v by asking a minimum number of yes/no questions (such as, “is v less than 6?”, or “is v one of the numbers 2, 5, 11?”)

Exercise 7 (Transferring a file efficiently) Consider a text file \mathcal{T} , written in English, to be sent across a binary channel. Assume that you are capable of sending an encoding scheme (a source code \mathcal{C}) across a different channel optimised for sending code descriptions.

1. Why is it not a good idea to make a unary code which maps $\mathcal{T} \mapsto 0$?
2. Device a means of sending \mathcal{T} efficiently.

¹That is, every infinite sequence of code symbols can be decomposed into code-words

²From left-to-right, that is, $p_1 \geq p_2 \geq \dots \geq p_q$.