Introduction to Information Theory, Data Compression, Coding

Mehdi Ibm Brahim, Laura Minkova

April 15, 2018

This is the augmented transcript of a lecture given by Luc Devroye on the 13th of March 2018 for a Data Structures and Algorithms class (COMP 252) at McGill University. The subject was Information Theory, Data Compression, and Coding.

Data Compression: The efficient encoding of information.

In many compression methods, input symbols are mapped to codewords (bit sequences). The set of codewords is called a code. If all codewords are of equal length, then we have a **fixed-length code**. Otherwise, we have a **variable-length code**. The most important codes are **prefix codes**, i.e., codes in which no codeword is the prefix of another codeword.

If codewords are mapped to binary trees (a 0 corresponding to a left edge, and a 1 to a right edge), then one can associate each symbol in a prefix code with a unique leaf. It is noteworthy that the compressed (a coded) sequence can be decoded to yield the input by repeatedly going down the tree until leaves are reached.

Claude E. Shannon (1916-2001) is a highly recognized American mathematician and computer scientist. He studied electrical engineering and mathematics at the University of Michigan before going on to complete a masters and postdoctorate degree at MIT. The computer science and engineering community increasingly began to notice his brilliant mind after the publication of his master's thesis "A Symbolic Analysis of Relay and Switching Circuits", written in 1936. His most notable and well known publication "A Mathematical Thoery of Communication", was published a few years later, in 1948. Although he worked in a field in which no Nobel Prize existed, he was granted numerous prestigious prizes throughout his career. He passed away at the age of 84 after a long fight with alzheimer disease.

Information Theory

Information Theory[1] is the study of information and how it can be processed and communicated. Not long after beginning work at the Bell Laboratories, Claude E. Shannon published his paper "A Mathematical Theory of Communication"[2], in 1948, in the Bell Systems Technical Journal.[3] This paper quickly gained wide-spread recognition as being the ground work for what is now known as modern day information theory.

The main premise of the paper was an investigation into solving communication problems, discussing them both in a theorectical and real life sense. The greatest difference between the two is that in real life, often times there is noise that can interfere with the mode of transmission of information, which he called the channel. For the purpose of this course, we consider a communication system in which no noise is present.

[1] Charles E. Leiserson, Thomas H. Cormen, and Ronald L. Rivest. *Introduction to Algorithms*. Cambridge, MA, 2009

[2] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 1948

[3] Inrene Woo Adel Magra, Emma Goune. Information theory. `http://luc.devroye.org/Magra-Goune-Woo--Shannon+InformationTheory-LectureNotes-McGillUniversity-20` pdf, March 2017. Accessed on 2018-03-20

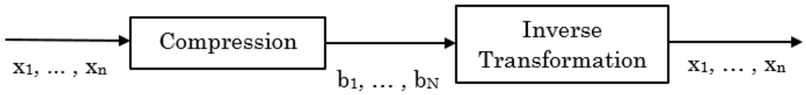


Figure 1: Noiseless communication system

Shannon's greatest concern was the "how" and not the "what" of information transmission. He did note, however, that in the case of data compression how well you compress (and how easily) depends on the input you are considering. That being said, he pays no attention to the actual meaning of the input, stating "...these semantic aspects of communication are irrelevant to the engineering problem."[4]

[4] Shannon [1948]

The compression ratio, C, is defined by

$$C = \frac{\text{number of symbols in output}}{\text{number of symbols in input}}$$

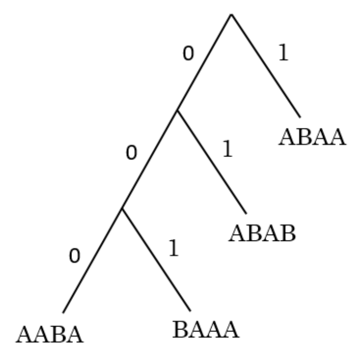
In order to determine the expected length of the output sequence, Shannon considered every possible input. He assumed that every input sequence that may have to be compressed has a given probability p_i, where the p_i's sum to one. If the i^{th} input, was given some encoding of length l_i bits, then the expected length of the output bit sequence is $\Sigma_i p_i li$.

Things to note:

1. Input in a communication system is not limited to words, characters etc. It can be anything!
2. Output is always binary.

A binary tree proved to be very useful in representing the encoding of information. The internal nodes of this tree would have no value, however each leaf would represent a possible input. Every left edge represents by a 0, and every right edge a 1.

Example:



Corresponding Translation Table:

Input	Code
AABA	000
BAAA	001
ABAB	01
ABAA	1

Entropy (Symbol $\mathcal{E}$)

In information theory, **entropy** is a quantity that measures the amount of information in a random variable. Thus entropy provides a theoretical (sometimes inachievable) limit for the efficiency of any possible encoding.[5]

The binary entropy is defined as follows

$$\mathcal{E} = \Sigma_i p_i \log_2 \frac{1}{p_i} \geq 0,$$

where p_i's are the probabilities of the input sequences.

[5] George Markowsky. Information theory. `https://www.britannica.com/science/information-theory`, June 2017. Accessed on 2018-03-11

Shannon faced three problems:

1. Find a binary tree that minimizes $\Sigma_i p_i li$ (solved by his student, David Huffman).

2. Prove $\mathcal{E} \leq \min \Sigma_i p_i li$, where "min" refers to the minimum over all binary trees. (Thus, the expected length of the output, regardless of the comparison method, is at least $\mathcal{E}$.)

3. Prove $\Sigma_i p_i l_i \leq \mathcal{E}+1$, for some binary tree. (This reassures us, since we can come close to the lowerbound, $\mathcal{E}$.)

We will first prove (2) $\mathcal{E} \leq min \Sigma_i p_i li$.

Proof. Recall Kraft's inequality, which is valid for all binary trees:

$$\sum \frac{1}{2^{\ell_i}} \leq 1$$

By Taylor's series expansion, $\log_e x \leq x - 1$. Now observe that:

$$\sum_i p_i \ell_i = \sum_i p_i \log_2 2^{\ell_i} \tag{1}$$

$$= \sum_i p_i \log_2 (2^{\ell_i} p_i \frac{1}{p_i}) \tag{2}$$

$$= \sum_i p_i \log_2 \frac{1}{p_i} + \sum_i p_i \log_2 (p_i 2^{\ell_i}) \tag{3}$$

$$= \mathcal{E} - (\log_2 e) \sum_i p_i \log_e \left(\frac{1}{p_i 2^{\ell_i}} \right) \tag{4}$$

$$\geq \mathcal{E} - (\log_2 e) \sum_i p_i \left(\frac{1}{p_i 2^{\ell_i}} - 1 \right) \tag{5}$$

$$= \mathcal{E}. \tag{6}$$

□

We have shown that $\sum_i p_i l_i \geq \mathcal{E}$. We must now exhibit a compression method with $\mathcal{E} + 1 \geq \sum_i p_i \ell_i$.

Proof. We take $\ell_i = \lceil \log_2 (\frac{1}{p_i}) \rceil$ so we have,

$$\sum_i \frac{1}{2^{\ell_i}} \leq \sum_i \frac{1}{2^{\log_2 \frac{1}{p_i}}} \leq \sum_i p_i \leq 1.$$

So, Kraft's inequality holds, By ordering the lengths l_i from small to large, and assigning the l_i's to leaves in a binary tree from left to right, one can find a code with the given l_i's. This code is called the Shannon-Fano code.

Now,

$$\sum_i p_i \ell_i \leq \sum_i p_i (1 + \log_2 \frac{1}{p_i}) = 1 + \mathcal{E}.$$

□

Huffman Tree

A Huffman tree is a binary tree that minimizes $\Sigma_i p_i li$ where p_i is the weight of leaf i and l_i is the distance from leaf i to the root. It has the following properties:

1. Two inputs with smallest p_i value are furthest from the root.
2. Every internal node has 2 children.
3. Two inputs with smallest p_i value can safely be made siblings.

It is important to note that Huffman trees are not unique!

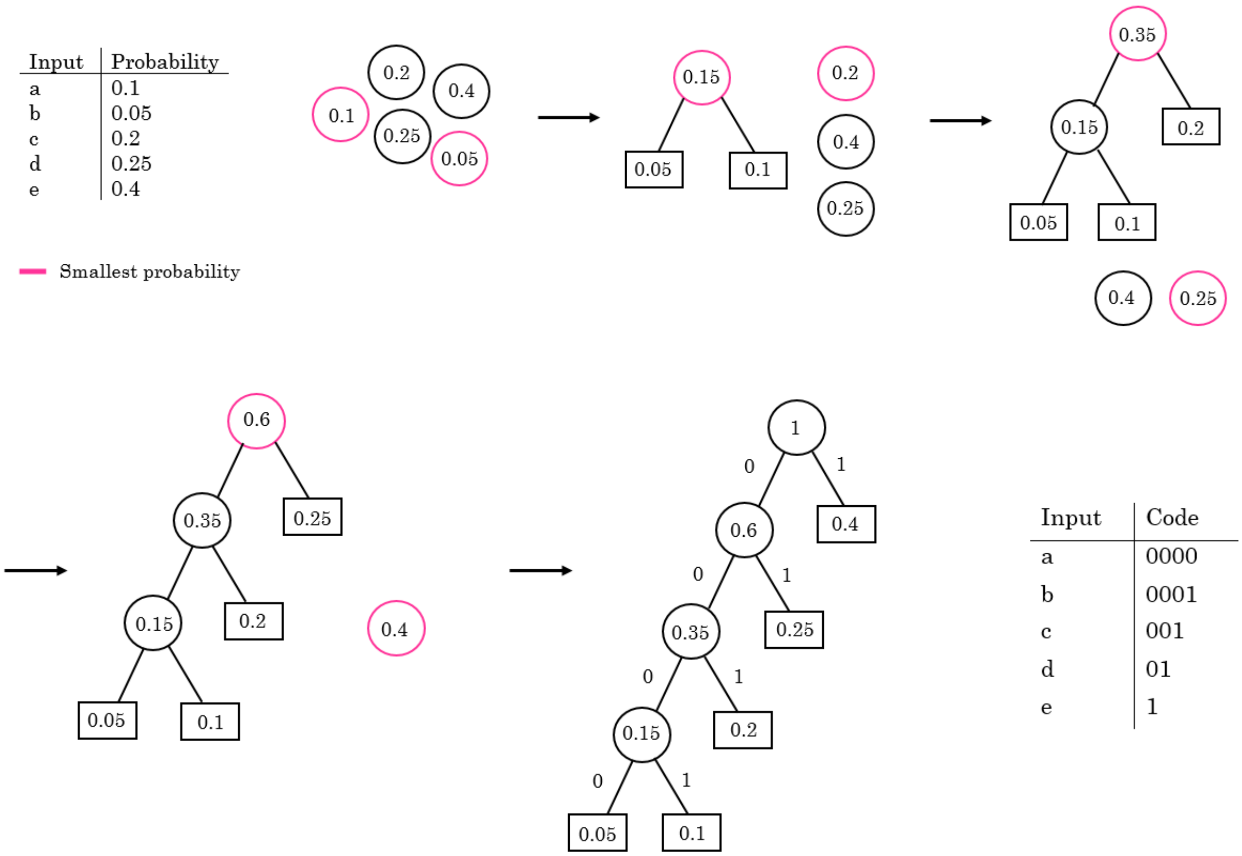
The Hu-Tucker algorithm is a greedy algorithm designed to output the Huffman tree given a set of inputs and their p_i's. It has time complexity $O(n \log n)$.

Setup:
Let PQ be a binary heap holding pairs (i, p_i) with the smallest key p_i near the root . Assuming that there are n leaves, we can reserve $n - 1$ interval nodes in an array of total size $2n - 1$. Let us use left[i] and right[i] to denote the children of node i. Node 1 is the root.

HUFFMANTREE

```
MAKENULL(PQ)
for i = n to 2n − 1 do
    left[i]=right[i]= nil;
    INSERT((i, p_i), PQ);
for i = n down to 1 do
    (a, p_a) = DELETEMIN(PQ);
    (b, p_b) = DELETEMIN(PQ);
    left[i]= a;
    right[i]= b;
    INSERT((i, (p_a + p_b)), PQ);
```

Example: How to construct a Huffman tree



Huffman tree

Examples

We will now show different methods of coding and see how they compare with Shannon's lower bound.
Suppose our input is $x_1, x_2, ..., x_n$ where x_i are uniformly random elements of $\{1,2,3\}$. There are, therefore, 3^n equally likely input sequences of length n. Note that $\mathcal{E} = \log_2 3^n = n \log_2 3 \approx 1.57n$.

1) (Fixed width length).
We use two bits per input symbol using the fixed width code:

$$1 \to 01, 2 \to 10, 3 \to 11.$$

So the length of the output is $2n$ which is not optimal. There is room for a smaller expected output length.

2) (Huffman code).
Consider the Huffman code where symbols are coded symbol by symbol using a Huffman tree prefix code:

$$1 \to 0, 2 \to 10, 3 \to 11.$$

The expected output length is $\frac{5}{3}n$, since

$$\sum p_i l_i = \frac{1}{3}(1) + \frac{1}{3}(2) + \frac{1}{3}(2) = \frac{5}{3}.$$

Thus, the expected output length is $\frac{5}{3}n$, which is considerably larger than $\mathcal{E} \approx 1.57n$.

3)
Let's now make groups of fixed length d. Each group of d is an input symbol coded by a Huffman code.

The expected output length in number of bits will be $\frac{n}{d}$ times the expected length of the Huffman tree code for one group, which we know is $\leq 1 + \log_2 3^d$. So the overall expected length is

$$\leq \frac{n}{d} \cdot \lceil \log_2 3^d \rceil \leq \frac{n}{d}(1 + d \log_2 3) = n(\log_2 3 + \frac{1}{d}).$$

Finally, by choosing d large enough, we can get arbitrarily close to $\mathcal{E}$. We cannot take d too large though, because computing the Huffman code would require too much space as the Huffman tree has 3^d leaves.

References

[1] Charles E. Leiserson, Thomas H. Cormen, and Ronald L. Rivest. *Introduction to Algorithms*. Cambridge, MA, 2009.

[2] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 1948.

[3] Inrene Woo Adel Magra, Emma Goune. Information theory. `http://luc.devroye.org/Magra-Goune-Woo--Shannon+InformationTheory-LectureNotes-McGillUniversity-2017.pdf`, March 2017. Accessed on 2018-03-20.

[4] George Markowsky. Information theory. `https://www.britannica.com/science/information-theory`, June 2017. Accessed on 2018-03-11.