Coding and Data Compression

Lec.4

Shannon-Fano Coding

Shannon-Fano coding, named after Claude Shannon and Robert Fano, was the first algorithm to construct a set of the best variable-length codes. We start with a set of n symbols with known probabilities (or frequencies) of occurrence. The symbols are first arranged in descending order of their probabilities. The set of symbols is then divided into two subsets that have the same (or almost the same) probabilities. All symbols in one subset get assigned codes that start with a 0, while the codes of the symbols in the other subset start with a 1. Each subset is then recursively divided into two sub subsets of roughly equal probabilities, and the second bit of all the codes is determined in a similar way. When a subset contains just two symbols, their codes are distinguished by adding one more bit to each. The process continues until no more subsets remain. Table 5.1 illustrates the Shannon-Fano algorithm for a seven-symbol alphabet. Notice that the symbols themselves are not shown, only their probabilities.

The Shannon-Fano method is easy to implement but the code it produces is generally not as good as that produced by the Huffman method that is explain in next section.

The algorithm

- Build the tree (top-down)
- Select the biggest frequency that is (A) and put it in the left side of the tree from the first node (ACBDE).
- Select the remainder symbols (CBDE) and put it on the right side from the first node (ACBDE), and so on to obtain just one symbol.
- After that put (0) on the left path and (1) on the right path.
- Obtain the code for all symbols from the root to the node.

This process is best illustrated by an example.

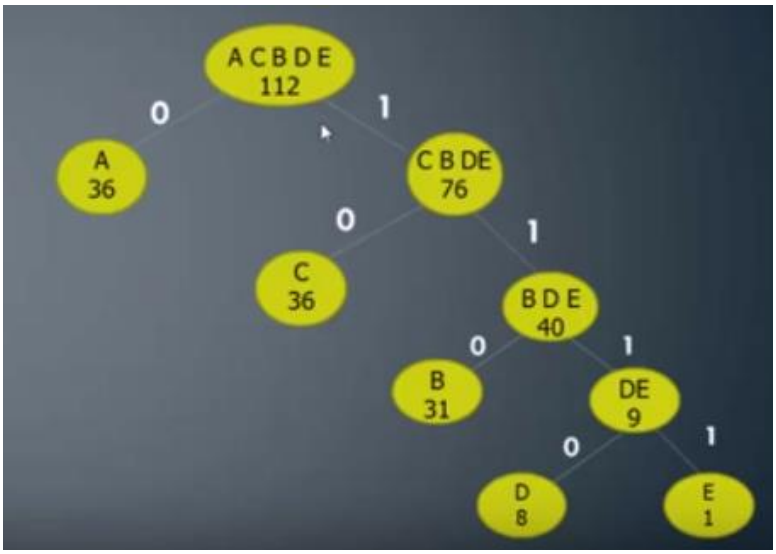
Ex: Find code for this data by using Shannon-Fano Coding,

A	A	C	B	B	D	A	B	B	B	A	A	A	A
A	A	C	B	B	D	A	B	B	B	A	A	A	A
A	A	C	B	B	D	A	D	D	D	A	A	A	A
A	A	C	B	B	D	A	E	C	C	C	C	C	C
A	A	C	C	B	B	A	D	C	C	C	C	C	C
A	C	C	C	B	B	A	B	B	B	C	C	C	A
A	C	C	C	B	B	A	B	B	B	C	C	C	A
A	C	C	C	B	B	A	B	B	B	C	C	C	A

Sol:

Symbols = 112

Arrange the symbols based on the biggest frequency: A=36, C=36, B=31, D=8, E=1.



The code are: A=0, C=10, B=110, D=1110, E=1111

Huffman Coding

Huffman coding is a popular method for data compression. It serves as the basis for several popular programs run on various platforms. Some programs use just the Huffman method, while others use it as one step in a multistep compression process. The Huffman method [Huffman 52] is somewhat similar to the Shannon-Fano method. It generally produces better codes, and like the Shannon-Fano method, it produces the best code when the probabilities of the symbols are

negative powers of 2. The main difference between the two methods is that Shannon-Fano constructs its codes top to bottom (from the leftmost to the rightmost bits), while Huffman constructs a code tree from the bottom up (builds the codes from right to left). Since its development, in 1952, by D. Huffman, this method has been the subject of intensive research into data compression.

The algorithm

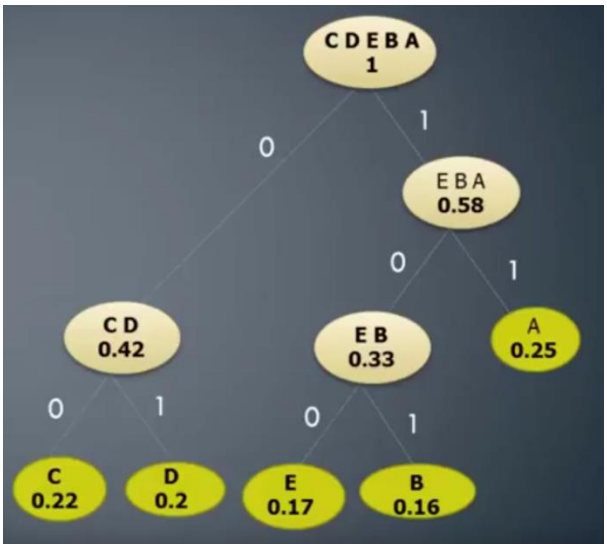
- Build the tree (down-top)

- Find the smallest two Probabilities and make summation between them, and so on.
- After that put (0) on the left path and (1) on the right path.
- Obtain the code for all symbols from the root to the node.

This process is best illustrated by an example.

Ex: If you have the symbols : A,B,C,D,E with Probabilities A=0.25, B=0.16, C=0.22, D=0.2, E=0.17

Sol:



A=11, B=101, C=0.22, D=00, E=100

Average bit length= 2(0.25) 3(0.16) 2(0.22) 2(0.2) 3(0.17)=2.33 bit per symbol

Entropy= $-(0.25\log_2 0.25+0.16\log_2 0.16+0.22\log_2 0.22+0.2+\log_2 0.2+0.17\log_2 0.17=2.3$ bit per symbol