# Information Theory and Coding
## Variable length code

م. فؤاد حمادي

2020 - 2019

## 2- Variable length code

The codes we have looked above have all fixed lengths and they are called ***block code*** from the fact that the message, are of fixed block lengths in the stream of symbols being sent.

We now examine variable length code in more detail. The ***advantage*** of a code where the message symbols are of variable length is that some time the code is more efficient in the sense that to represent the same information we can use ***fewer digits on the average***. To accomplish this we need to know something about the statistics of the message being sent. If every symbol is as likely as every other one, then the fixed length code are about as efficient as any code can be. But if some symbols are more probable than others, then we can take advantage of this feature to make the ***most frequent symbols correspond to the shorter encodings*** and the ***rare symbols correspond to the longer encodings***. This is exactly the idea behind variable length coding.

However, variable length code bring with them a fundamental problem, at the receiving end, how do you recognize each symbol of the code? In, for example, a binary system how do you recognize the end of one code word and the beginning of the next ?

If the probabilities of the frequencies of occurrence of the individual symbols are sufficiently different , then variable – length encoding can be significantly more efficient than fixed – length encoding

$$L = \sum_{i=1}^{N} P_i l_i \qquad \textbf{bit/symbol}$$

**Note**: for each symbol increase the probability decrease the code length

| **Symbol** | **probability** | **code length** |
|---|---|---|
| X 1 | p 1 | $I_1$ |
| X 2 | p 2 | $I_2$ |
| . | . | . |
| . | . | . |
| X m | p m | Im |

## 1. Source coding for special source

Source coding can achieve 100 % efficiency when r* -level code is used with source having a probability in the form $P(x\ i)= r^{-Ii}$ ( where Ii is an integer ) for all xi .

**Ex1:** design a binary code for the following source :

| **X i** | **P(xi )** | **P(xi)** | **I i** | **Code** |
|---|---|---|---|---|
| $x_1$ | 1/4 | $2^{-2}$ | 2 | 00 |
| $x_2$ | 1/4 | $2^{-2}$ | 2 | 11 |
| $x_3$ | 1/8 | $2^{-3}$ | 3 | 010 |
| $x_4$ | 1/8 | $2^{-3}$ | 3 | 111 |
| $x_5$ | 1/16 | $2^{-4}$ | 4 | 0000 |
| $x_6$ | 1/16 | $2^{-4}$ | 4 | 0101 |
| $x_7$ | 1/16 | $2^{-4}$ | 4 | 1001 |
| $x_8$ | 1/16 | $2^{-4}$ | 4 | 1111 |

**check**

$L = \sum I_i\, p(x_i)$

$=2/4 +2/4+3/8+3/8+4/16+4/16+4/16+4/16$

$=1/2+1/2+3/4+1=2.75$ bit/symbol

$H(x)= -\sum p(x_i) \log p(x_i)$

$= 2.75$ bit/symbol

$\xi$ code $= H(x)/L * 100\% \quad = 275/275 *100\% = 100\%$

**Ex2:** A source produce 7 symbols $x_1, x_2, \ldots, x_7$, with probabilities 0.0625, 0.25, 0.125, 0.25, 0.125, 0.0625, 0.125. design a binary code for the above source, then determine the code efficiency.

| **X i** | **P(x i)** | **P(xi)** | **l i** | **Code** |
|---|---|---|---|---|
| x 1 | 1/16 | $2^{-4}$ | 4 | 1000 |
| x 2 | 1/4 | $2^{-2}$ | 2 | 00 |
| x 3 | 1/8 | $2^{-3}$ | 3 | 010 |
| x 4 | 1/4 | $2^{-2}$ | 2 | 11 |
| x 5 | 1/8 | $2^{-3}$ | 3 | 101 |
| x 6 | 1/16 | $2^{-4}$ | 4 | 1001 |
| x 7 | 1/8 | $2^{-3}$ | 3 | 110 |

**check**

$L = \sum l_i \, p(x_i)$

$=4/16+2/4+3/8+2/4+3/8+4/16+3/8$

$= 0.25 + 0.5 + 0.375 + 0.5 + 0.375 + 0.25 + 0.375 = 2.625$ Bit/symbol

$H(x) = -\sum P(xi) \log P(xi)$

$= - (1/16*-4 +1/4*-2 +1/8*-3 + 1/4*-2 +1/8*-3 +1/16 *-4 +1/8 * -3)$

$= 2.625$ Bit/symbol

$\xi$ code= $H(x) / L * 100\% = 2.625 / 2.625 * 100\% = 100\%$

## Shannon – Fano method

To encode a message using Shannon-Fano method, you can follow the below steps :

1. Sort the symbols in descending order according to their probabilities.

2. Divide the list of symbols into two parts : upper and lower, so that the summation of the probabilities of the upper part is equal as possible to the summation of the lower part symbols.

3. Assign "0" code to each of the upper part symbols, and "1" code to each of the lower part symbols.

4. Divide each of the upper and lower part into upper and lower subdivision as in step (2) above, and assign the code "0" and "1" as in step (3) above.

5. Continue in step(4) until each subdivision contains only one symbols.

**Ex1:** A source produce 5 independent symbols ( $x_1, x_2, x_3, x_4, x_5$ ) with its corresponding probabilities 0.1, 0.3, 0.15, 0.25, 0.2 . design a binary code for the above source source symbol using Shannon – fanon method.

| symbols | P i | code | Ii |
|---|---|---|---|
| x 2 | 0.3 | 0 0 | 2 |
| x 4 | 0.25 | 0 1 | 2 |
| x 5 | 0.2 | 1 0 | 2 |
| x 3 | 0.15 | 1 1 0 | 3 |
| x 1 | 0.1 | 1 1 1 | 3 |

$$L = \sum I_i\, p(x_i)$$

$$= 2*0.3 + 2*0.25 + 2*0.2 + 3*0.15 + 3*0.1 \quad = 2.25 \text{ Bit\textbackslash symbol}$$

$$H = -\sum P(xi) \log P(xi)$$

$$= -( 0.3 \log 0.3 + 0.25 \log 0.25 + 0.2 \log 0.2 + 0.15 \log 0.15 + 0.1 \log 0.1) \quad = 2.228 \text{ Bit/symbol}$$

$$\xi \text{ code} = H(x) / L \; * 100\,\% = 2.228 / 2.25 * 100\,\% = 99\,\%$$