Arithmetic coding, lecture example

Coding

Alphabet $\mathcal{A} = \{1, 2, 3\}$. Cumulative distribution function

$$F(0) = 0,\ F(1) = 0.6,\ F(2) = 0.9,\ F(3) = 1$$

Code the sequence 1,3,2,1. Number of symbols to code $n = 4$.

$$\begin{aligned}
l^{(0)} &= 0 \\
u^{(0)} &= 1 \\
\\
l^{(1)} &= 0 + (1-0)\cdot 0 = 0 \\
u^{(1)} &= 0 + (1-0)\cdot 0.6 = 0.6 \\
\\
l^{(2)} &= 0 + (0.6-0)\cdot 0.9 = 0.54 \\
u^{(2)} &= 0 + (0.6-0)\cdot 1 = 0.6 \\
\\
l^{(3)} &= 0.54 + (0.6-0.54)\cdot 0.6 = 0.576 \\
u^{(3)} &= 0.54 + (0.6-0.54)\cdot 0.9 = 0.594 \\
\\
l^{(4)} &= 0.576 + (0.594-0.576)\cdot 0 = 0.576 \\
u^{(4)} &= 0.576 + (0.594-0.576)\cdot 0.6 = 0.5868
\end{aligned}$$

The sequence corresponds to the interval $[0.576 \quad 0.5868)$. The interval size is 0.0108 and thus we will need at least $\lceil -\log_2 0.0108 \rceil = 7$ bits in our codeword, maybe one more. Write the two interval limits as binary numbers:

$$\begin{aligned}
0.576 &= 0.10010011011\ldots \\
0.5868 &= 0.10010110001\ldots
\end{aligned}$$

The smallest seven bit number inside the interval is 0.1001010, and all numbers starting with these bits are also inside the interval (ie smaller than the upper interval limit). Thus, seven bits are enough. The codeword is **1001010**.

Decoding

The decoder has to know the number of symbols coded ($n = 4$) and the cumulative distribution function:

$$F(0) = 0,\ F(1) = 0.6,\ F(2) = 0.9,\ F(3) = 1$$

Decode the first codeword in the bitstream starting 1001010001011 . . .

Read one bit at a time. The bits read sofar $b_1 b_2 \ldots b_k$ specify an interval $[0.b_1 b_2 \ldots b_k \quad 0.b_1 b_2 \ldots b_k + 1/2^k)$ As soon as this interval is wholly inside the subinterval for a particular symbol sequence, we can decode one symbol. For the first symbol, the intervals corresponding to symbols 1, 2 and 3 are $[0 \quad 0.6)$, $[0.6 \quad 0.9)$ and $[0.9 \quad 1)$ respectively.

bits	binary interval	decimal interval
1	[0.1 1)	[0.5 1)
10	[0.10 0.11)	[0.5 0.75)
100	[0.100 0.101)	[0.5 0.625)
1001	[0.1001 0.1010)	[0.5625 0.625)
10010	[0.10010 0.10011)	[0.5625 0.59375)

Since this interval is wholly inside the interval for symbol 1, we can decode the first symbol as 1. We then split this symbol interval into three parts acording to F. This is gives us the intervals corresponding to symbols 1,2 and 3 as $[0 \quad 0.36)$, $[0.36 \quad 0.54)$ and $[0.54 \quad 0.6)$ respectively. We again check our bit interval and see that we are wholly inside the interval for 3. Thus we decode the second symbol as 3. We again split the symbolinterval into three parts acording to F. This is gives us the intervals corresponding to symbols 1,2 and 3 as $[0.54 \quad 0.576)$, $[0.576 \quad 0.594)$ and $[0.594 \quad 0.6)$ respectively. Checking our bit interval again, we see it covers more than one symbol interval. We must thus read more bits.

bits	binary interval	decimal interval
100101	[0.100101 0.10011)	[0.578125 0.59375)

This interval is wholly inside the interval for symbol 2, so we can decode the third symbol as 2. Split this symbol interval into three parts acording to F. This is gives us the intervals corresponding to symbols 1,2 and 3 as $[0.576 \quad 0.5868)$, $[0.5868 \quad 0.5922)$ and $[0.5922 \quad 0.594)$ respectively. Checking our bit interval again, we see it covers more than one symbol interval. We must thus read more bits.

bits	binary interval	decimal interval
1001010	[0.1001010 0.1001011)	[0.578125 0.5859375)

This interval is wholly inside the interval for symbol 1, so we can decode the fourth symbol as 1. Since we have now decoded 4 symbols, we are finished. The following bits in the bitstream belong to the next codeword.

Our decoded symbol sequence is 1,3,2,1 which is exactly the symbol sequence that that we coded.