

Codificação Aritmética

- Apresenta taxas de compressão melhores que o código de Huffman;
- No código de Huffman um símbolo cuja probabilidade, $p(s_i)$, esteja próximo de 1, ou seja, $\log_2 \frac{1}{p(s_i)}$ está próximo de zero, atribuir um bit é muito penalizador.

(notar que no mínimo, o código de Huffman, atribui um bit ao símbolo mais provável !)

- A codificação aritmética trata a mensagem como uma unidade

- Algoritmo

```
low = 0.0
high = 1.0
range = high - low
While (not end)
    s = read simbol
    high = low + range * highrange(s)
    low = low + range * lowrange(s)
    range = high - low
end
write code;           % low <= code < high
```

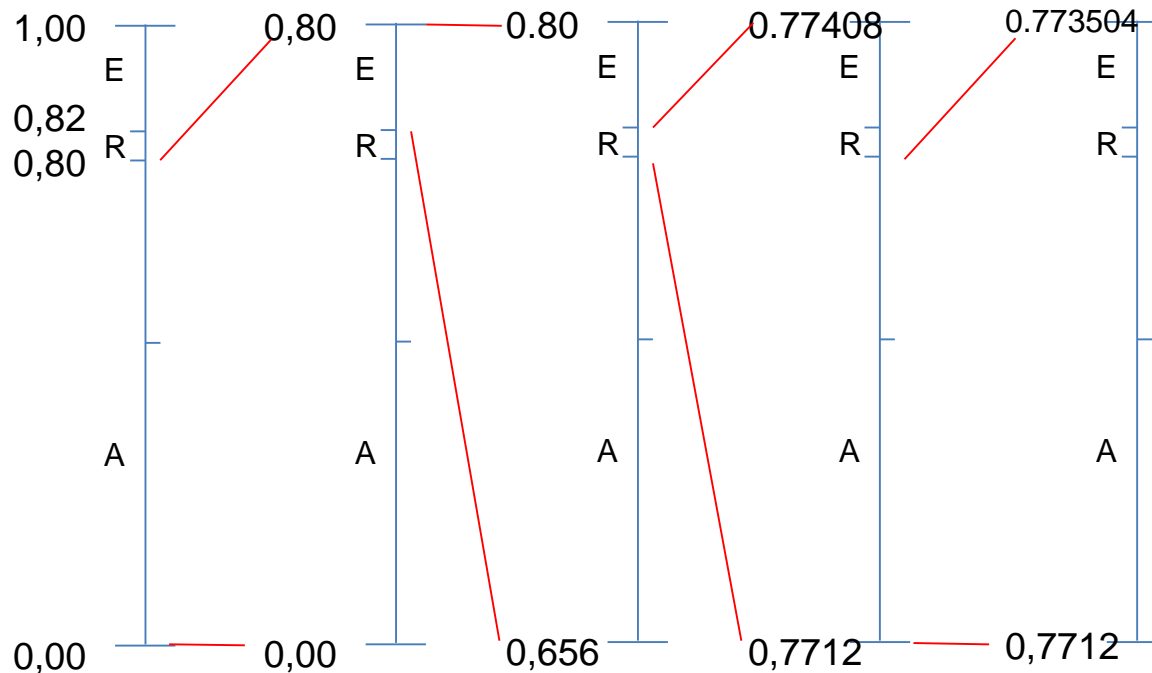
Codificação Aritmética decimal

Exemplo: símbolos possíveis:

Símbolo	prob.	interval
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
R	0.80-0.82	0.7712	0.77408	0.00288
A	0.00-0.80	0.7712	0.773504	0.002304

Mensagem: "AERA"



Valor entre
0,7712 e 0,773504

Codificação Aritmética decimal

- Valor entre 0,7712 e 0,773504

- 1 = 0.5

- 1 1 = 0.75

- 1 1 1 = 0.875

- 1 1 0 1 = 0.8125

- 1 1 0 0 1 = 0.78125

- 1 1 0 0 0 1 = 0.765625

- 1 1 0 0 0 1 1 = 0.7734375

Descodificação Aritmética decimal

■ Algoritmo

value = read code

While (not end)

find simbol such that: $\text{lowrange}(s) < \text{code} < \text{highrange}(s)$

write s

low = lowrange(s)

high = highrange(s)

range = high – low

value = (value – low) / range

end

■ Code: 0,7734375

■ Mensagem: A

■ Code: 0,966797

■ Mensagem: AE

■ Code: 0,815538

■ Mensagem: AER

■ Code: 0,77691

■ Mensagem: AERA



Codificação Aritmética

- A compressão é melhor que o código de Huffman;
- No pior caso, o código mais pequeno é no máximo

$$L \leq \log_2 \frac{1}{range} = \log_2 \frac{1}{\prod_i p_i}$$

- Notar que o código de Huffman pode ultrapassar este limite.
- Desvantagens:
 - Precisa de saber à priori as probabilidades de cada símbolo;
 - Necessidade de trabalhar com números com muita precisão;
 - O tempo de compressão/descompressão pode ser elevado, devido à complexidade do cálculo.
- Há implementações práticas que escalam os intervalos para trabalhar com inteiros.
- É usado na codificação JPEG.