Práctica 3. Codificación Aritmética (expresión en base 10)

Para realizar una codificación aritmética debemos de partir de una fuente de información base. Con dicha fuente de información se construye una partición del intervalo de la recta real

$$[0,1]$$

de forma que a cada símbolo de la fuente se le asocia un subintervalo de la partición, de forma que la amplitud de cada subintervalo es la probabilidad de aparición del símbolo de la fuente.

Ejemplo de la partición.

Supongamos que tenemos una fuente de información con dos símbolos equiprobables

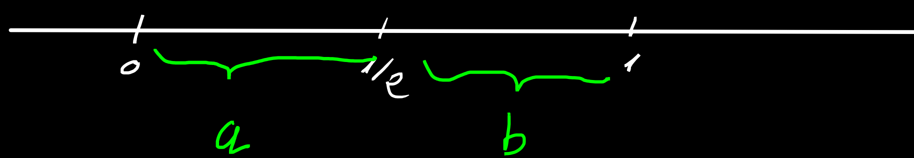
$$A = \{a, b\} \qquad p_1 = 1/2 \qquad p_2 = 1/2$$

La partición sería

		intervalo
a	$1/2$	$[0, 1/2)$
b	$1/2$	$[1/2, 1)$

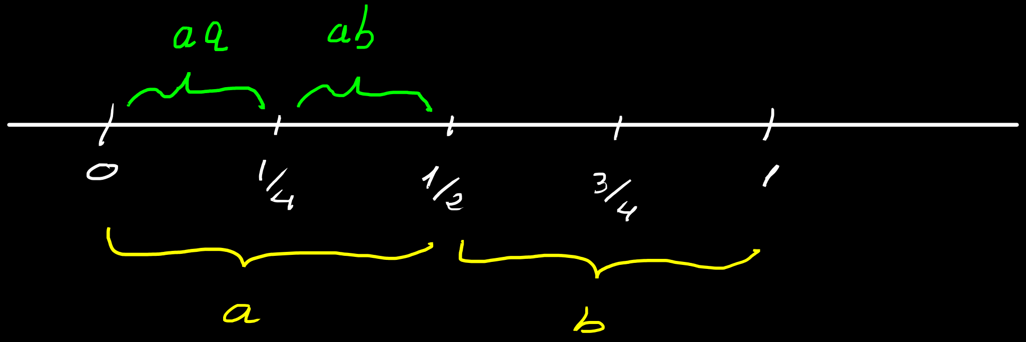
Gráficamente

$0 \leftrightarrow p_1 \quad p_1+p_2 \quad \dots \quad \sum p_j$



Partición del intervalo $[0,1)$

Repitiendo esta idea podemos asociar intervalos a todas las secuencias escritas en el alfabeto $\{a, b\}$. Por ejemplo, para la secuencia ab, nos situamos en el intervalo que codifica a, el $[0, 1/2)$, y de manera proporcional hacemos una partición según la probabilidad de emisión de cada símbolo.



$$ab \Downarrow [1/4, 1/2)$$

Idea de la Codificación Aritmética (expresión en base 10)

Usando esta idea, dado cualquier mensaje escrito en el alfabeto de la fuente F, se realizan dos pasos:

(1) Asignar un subintervalo del intervalo $[0,1) \longrightarrow [L,H)$

(2) Dar la expresión decimal "mas corta" de un número que esté dentro del intervalo $[L,H)$

Datos previos

a_1	p_1	$\longrightarrow [0, p_1)$	notación	$[L_1, H_1)$
a_2	p_2	$\longrightarrow [p_1, p_1+p_2)$		$[L_2, H_2)$
$\vdots$	$\vdots$			
a_m	p_m	$\longrightarrow [\sum_{i=1}^{m-1} p_i, 1)$		$[L_m, H_m)$

Partición Base

Proceso de Codificación (sobre un ejemplo)

alf	frec	intervalo
P	1	$[0, 1/7)$
A	2	$[1/7, 3/7)$
S	1	$[3/7, 4/7)$
E	2	$[4/7, 6/7)$
L	1	$[6/7, 1)$
	7	

Mensaje a codificar

"ASA"

"A" $\longrightarrow$ $[1/7, 3/7)$

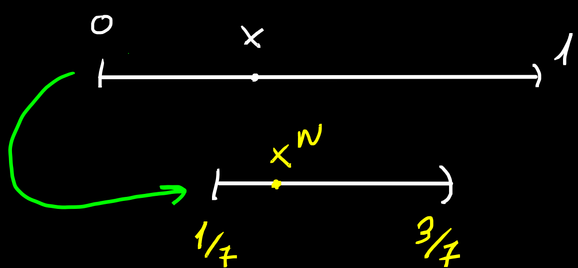
Partición Base

Partición del intervalo $[0, 1)$

"A" $\longrightarrow$ $[1/7, 3/7)$

"S" $\longrightarrow$ $[3/7, 4/7)$ sería el subintervalo para la partición $[0,1)$

* hay que actualizar los datos para que sea un subintervalo para la partición del $[1/7, 3/7)$



Mantener la proporción

$$\frac{x-0}{1-0} = \frac{x^N - 1/7}{3/7 - 1/7}$$

luego

$$x^N = 1/7 + (3/7 - 1/7)\, x$$

$$[L, H) \qquad a_j \longrightarrow [L_j, H_j) \longrightarrow [L^N, H^N)$$

- $[L, H)$: intervalo inicial
- a_j: símbolo a codificar
- $[L_j, H_j)$: subintervalo del $[0,1]$
- $[L^N, H^N)$: hemos actualizado los datos para que sea un subintervalo de $[L, H)$

$$L^N = L + (H-L)\, L_j$$

$$H^N = L + (H-L)\, H_j$$

Es decir, al actualizar los datos de $[3/7, 4/7)$ a un subintervalo de $[1/7, 3/7)$ se tiene

$$L^N = 1/7 + (3/7 - 1/7) \cdot 3/7 = 13/49$$

$$H^N = 1/7 + (3/7 - 1/7) \cdot 4/7 = 15/49$$

"A" $\longrightarrow [1/7, 3/7)$

"S" $\longrightarrow [3/7, 4/7) \rightsquigarrow [13/49, 15/49)$

El intervalo base actual es $\left[\frac{13}{49}, \frac{15}{49}\right)$

Toca añadir en la codificación el símbolo A.

"A" $\longrightarrow \left[\frac{1}{7}, \frac{3}{7}\right)$ actualizamos los datos al intervalo $\left[\frac{13}{49}, \frac{15}{49}\right)$

$$L^N = \frac{13}{49} + \left(\frac{15}{49} - \frac{13}{49}\right) \cdot \frac{1}{7} = \frac{93}{343}$$

$$H^N = \frac{13}{49} + \left(\frac{15}{49} - \frac{13}{49}\right) \cdot \frac{3}{7} = \frac{97}{343}$$

Intervalo actualizado $\left[\frac{93}{343}, \frac{97}{343}\right)$

Proceso de codificación

"A" $\longrightarrow [1/7, 3/7)$

"S" $\longrightarrow [3/7, 4/7) \rightsquigarrow [13/49, 15/49)$

"A" $\longrightarrow [1/7, 3/7) \rightsquigarrow [93/343, 97/343) = [L, H)$

(2) Dar la expresión decimal "mas corta" de un número que esté dentro del intervalo $[L, H)$

$0.271\ldots \qquad 0.282\ldots$

Codificación 0.28

Criterios para la elección de cifras decimales

$L = 0.\underbrace{l_1 l_2 \cdots}_{\text{cifras en base 10 de } L}$ $\qquad$ $H = 0.\underbrace{h_1 h_2 \cdots}_{\text{cifras en base 10 de } H}$

Sea r la posición hasta la cual las cifras decimales de L y H coinciden

para $j \leq r$ se tiene $l_j = h_j$ y además $l_{r+1} < h_{r+1}$

1. Si H tiene más de $r+1$ cifras decimales, el algoritmo devuelve

$$0.h_1 h_2 \ldots h_{r+1} \quad \text{FIN}$$

2. En caso contrario

2.1 Si $l_{r+1} + 1 \neq h_{r+1}$ entonces el algoritmo devuelve

$$0.l_1 l_2 \ldots l_r \underbrace{l_{r+1} + 1} \quad \text{FIN}$$

2.1 Si $l_{r+1} + 1 = h_{r+1}$ entonces se cogen las $r+1$ cifras decimales de L, y las siguientes hasta llegar a una distinta de 9. A esta cifra se le suma uno y el algoritmo termina.

$$\underbrace{\rule{4cm}{0.4pt}}_{\substack{r+1 \\ \text{cifras decimales} \\ \text{de } L}} \; 9\,9 \ldots 9 \; \underbrace{\text{cifra} \neq 9}_{\text{cifra}+1} \quad \text{FIN}$$

Nota.- El proceso mejora teniendo en cuenta que si en un paso se llega a la representación en base 10 de L, ésta se puede tomar como codificación.

Proceso de Decodificación (mismo ejemplo)

alf	frec	intervalo
P	1	$[0, 1/7)$
A	2	$[1/7, 3/7)$
S	1	$[3/7, 4/7)$
E	2	$[4/7, 6/7)$
L	1	$[6/7, 1)$
	7	

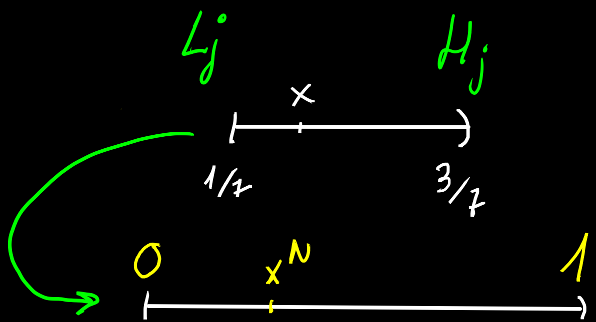
codificación $\underline{num = 0.28}$

- localizamos intervalo

$$1/7 < 28/100 < 3/7$$

- Extraemos símbolo ⇝ "A"
- Actualizamos el dato num.

Mantener la proporción



$$\frac{x - 1/7}{3/7 - 1/7} = \frac{x^N - 0}{1 - 0} \quad \text{luego}$$

$$x^N = \frac{x - 1/7}{3/7 - 1/7} \Rightarrow num^N = \frac{num - L_j}{H_j - L_j}$$

$$num^N = \frac{28/100 - 1/7}{3/7 - 1/7} = 12/25 = 0.48$$

$num = 28/100 \rightsquigarrow$ "A" $\rightsquigarrow$ actualizamos $num^N = 12/25 = 0.48$

$num = 48/100$

- localizamos intervalo

  $3/7 < 12/25 < 4/7$

- Extraemos símbolo $\rightsquigarrow$ "S"

- Actualizamos el dato num $\Rightarrow$ $num^N = \frac{48/100 - 3/7}{4/7 - 3/7} = 9/25$

  $= 0.36$

$num = 28/100 \rightsquigarrow$ "A" $\rightsquigarrow$ actualizamos $num^N = 48/100 = 0.48$

$num = 48/100 \rightsquigarrow$ "S" $\rightsquigarrow$ actualizamos $num^N = 9/25 = 0.36$

$num = 36/100$

- localizamos intervalo

  $1/7 < 9/25 < 3/7$

- Extraemos símbolo $\rightsquigarrow$ "A"

FIN

"ASA"

$\longrightarrow$

Criterios de parada

→ se conoce la longitud del mensaje

→ hacer la codificación para mensajes de longitud previamente establecida (codificación por bloques)

Adaptación al caso binario (cómo obtener las cifras en base 2)

En base 10

$$0.23 = 2 \cdot \frac{1}{10} + 3 \cdot \frac{1}{10^2}$$

$$0.23 \xrightarrow[\text{por } 10]{} 2.3 \xrightarrow{2.3-2} 0.3 \xrightarrow[\text{por } 10]{} 3 \xrightarrow{3-3} 0$$

FIN

2 (parte entera)

3 (parte entera)

En binario

$$0.101_{2} = 1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2^3} = 0.625$$

$$0.625 \xrightarrow[\text{por } 2]{} 1.25 \longrightarrow 0.25 \xrightarrow[\text{por } 2]{} 0.5 \longrightarrow 0.5 \xrightarrow[\text{por } 2]{} 1 \longrightarrow 0$$

BASE 10

1   0   1

FIN

cifras en binario