Introducción a la Teoría de la Información
Codificación de fuentes

Facultad de Ingeniería, UdelaR

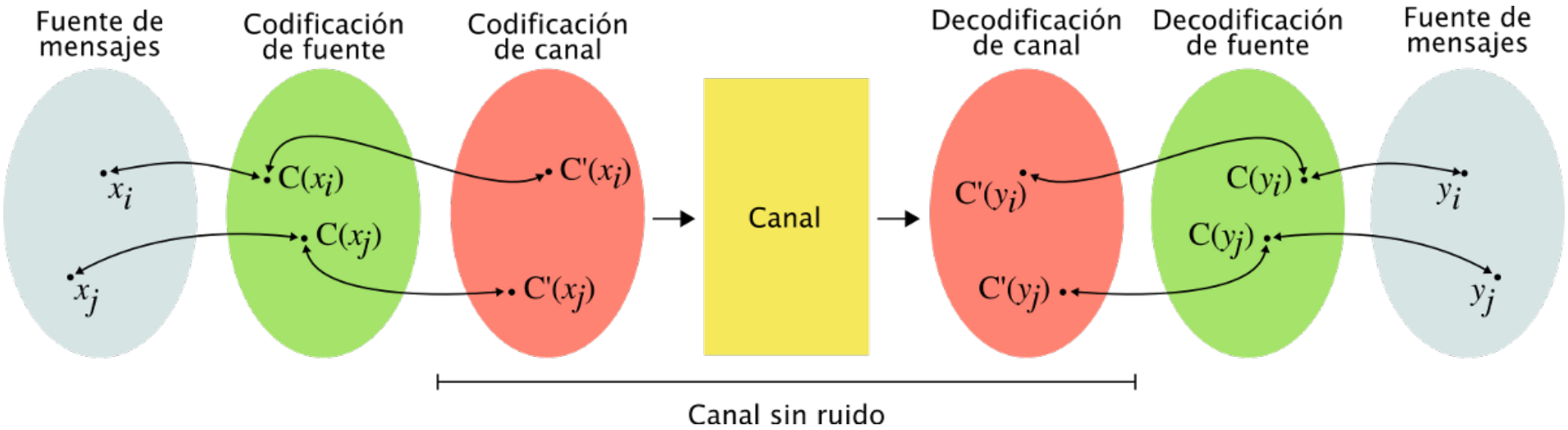
Agenda

1. Codificación de fuente
   - Definiciones
   - Clasificación de códigos
   - Desigualdad de Kraft
   - Códigos óptimos
   - Teorema de Codificación de Fuente
   - Cotas para el largo medio

2. Esquemas de codificación
   - Códigos de Shannon-Fano
   - Códigos de Huffman
   - Códigos de Shannon-Fano-Elias
   - Codificación aritmética
   - Optimalidad competitiva de los códigos

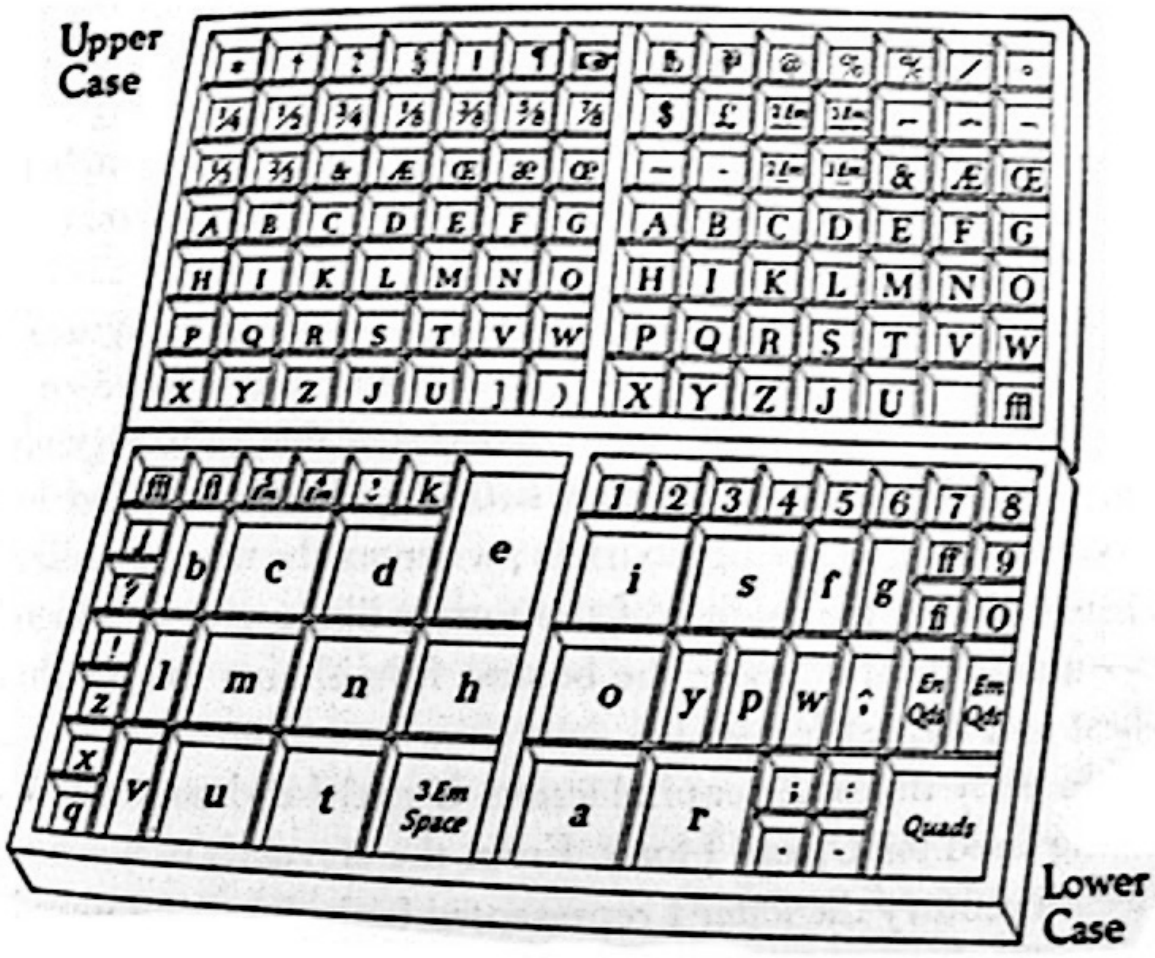
Codificación de fuente

Fuente generadora de mensajes de un alfabeto fuente $\mathcal{X} = \{x_1, \ldots, x_m\}$ con probabilidades $p_X(x_i)$. A cada uno de los mensajes se le asignará una palabra de código $C(x_i)$.



Cómo asignar las palabras de códigos de forma *óptima* y sistemática?

Código Morse



Codificación de fuente

Sea $\mathcal{D} = \{0, 1, \ldots, D-1\}$ un alfabeto código D-ario.

$$\mathcal{D}^* = \cup_{k=1}^{k=\infty} d^k$$

con $d^k = d_1 d_2 \ldots d_k$ con $d_i \in D$, es el conjunto de posibles palabras que pueden formarse con los elementos de $\mathcal{D}$.

Definición (Código fuente)

Un *código fuente* C para una variable aleatoria X es un mapeo de $\mathcal{X}$ en $\mathcal{D}^*$, i.e.,

$$C : \mathcal{X} \to \mathcal{D}^*$$

- Cada mensaje x_i tendrá asignado una palabra de código $C(x_i)$ de largo $l(x_i) = l_i$.
- D generalmente es chico ($D = 2$)

Ejemplo

C es un código de fuente para $\mathcal{X} = \{x_1, x_2\}$ con alfabeto $\mathcal{D} = \mathcal{B} = \{0, 1\}$, $C(x_1) = 00$ y $C(x_2) = 11$

Definición (Largo medio de un código)

El *largo medio* $L(C)$ de un código $C(x)$ para una variable aleatoria X con distribución de probabilidad $p(x)$ se define como

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$$

donde $l(x)$ es el largo de la palabra de código asignada a x.

Ejemplo

- $p_X(\mathcal{X}) = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$ con $C(X) = \{0, 10, 110, 111\}$, $L(C) = H(X) = 1.75$ bits
- $p_X(X) = \left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$ con $C(X) = \{0, 10, 11\}$, $L(C) = 1.66 > H(X) = 1.58$ bits

Clasificación de códigos

Definición (Código no singular)

Un código es *no singular* si cada elemento de $\mathcal{X}$ se mapea en una palabra de código diferente de $\mathcal{D}^*$

$$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)$$

Definición (Extensión de un código)

La *extensión* C^* de un código C es un mapeo de una secuencia de símbolos de $\mathcal{X}$ en un secuencia de $\mathcal{D}$ definida por

$$C(x_1 x_2 \dots x_n) = C(x_1)C(x_2)\dots C(x_n),$$

donde $C(x_1)C(x_2)\dots C(x_n)$ es la concatenación de las palabras de código.

Clasificación de códigos

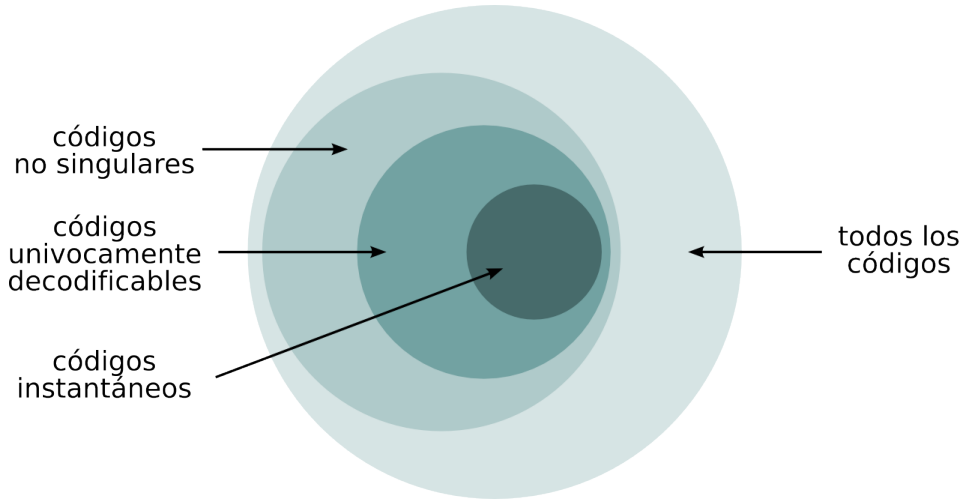
Definición (Código unívocamente decodificable)

Un código es *unívocamente decodificable* si su extensión es no singular.

O sea, no hay ambigüedades al momento de decodificar una secuencia codificada.

Definición (Código instantáneo)

Un código es *instantáneo* o *de prefijo* si ninguna palabra de código es prefijo de otra palabra de código.



Clasificación de códigos

Ejemplo

$\mathcal{X}$	C_1	C_2	C_3	C_4	C_5
x_1	0	0	10	0	0
x_2	0	010	00	10	10
x_3	0	01	11	110	110
x_4	0	10	110	1110	111

- C_1 es singular, no sirve para mucho
- C_2 no es unívocamente decodificable (UD), la secuencia 010 puede decodificarse como x_2, x_1x_4 o x_3x_1
- C_3 es UD pero no es instantáneo.
  Si se recibe $0010...11...111..110..1101..1100..$ se decodifica $x_2x_1x_?x_3..x_?x_4..x_3x_2$
- C_4 es instantáneo, es un código de puntuación (de coma), el 0 marca el final de la palabra , ó*es eficiente*?
  Si se recibe 01101110 se decodifica $x_1x_3x_4$
- C_5 es instantáneo.

Desigualdad de Kraft

Teorema (Desigualdad de Kraft)

Para todo código instantáneo sobre un alfabeto de tamaño D y largos de palabra $l(x_1), l(x_2), \ldots, l(x_m)$ se debe cumplir

$$K = \sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1$$

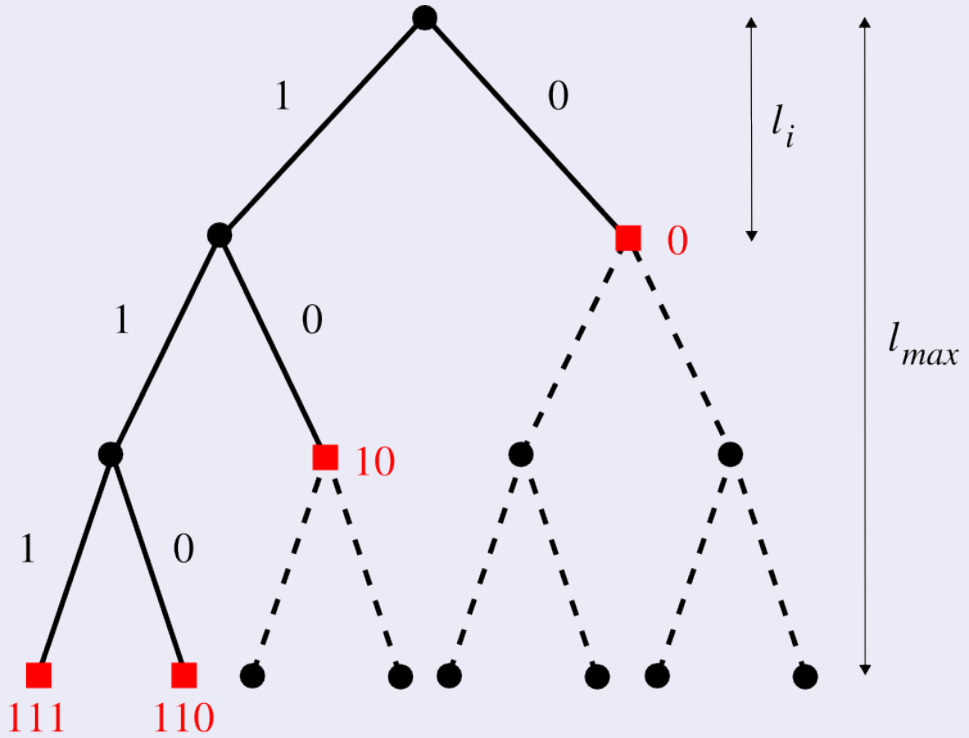
Igualmente dado un conjunto de largos de código que cumple la desigualdad, existe un código instantáneo con esos largos.

- Las longitudes de las palabras no pueden ser todas "cortas", si hay una muy corta debe haber otras más largas.
- No especifica cómo asignar los largos.

Desigualdad de Kraft

Demostración

Considerar un árbol D-ario (ramas, hojas y camino).



$$\sum_i D^{l_{\text{máx}}-l_i} \leq D^{l_{\text{máx}}}$$

Desigualdad de Kraft extendida

Teorema (Desigualdad de Kraft extendida)

Para todo código instantáneo contable infinito sobre un alfabeto de tamaño D los largos de palabra deben cumplir

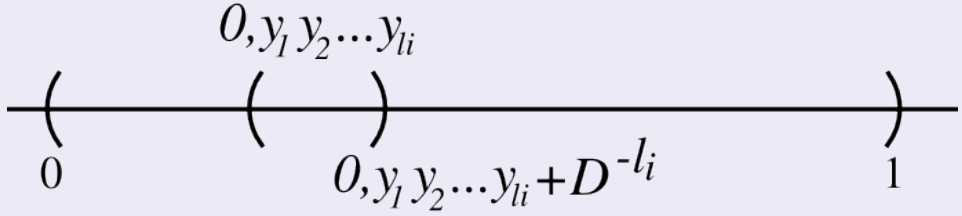
$$\sum_{i=1}^{+\infty} D^{-l_i} \leq 1$$

Igualmente dado un conjunto de largos de código que cumple la desigualdad, existe un código instantáneo con esos largos.

Desigualdad de Kraft extendida

Demostración

Sea $y_1 y_2 \dots y_{l_i}$ la i-ésima palabra, y sea $0.y_1 y_2 \dots y_{l_i} = \sum_{j=1}^{l_i} y_j D^{-j}$ el número real representado.
El intervalo $[0.y_1 y_2 \dots y_{l_i}, 0.y_1 y_2 \dots y_{l_i} + D^{-l_i}) \in [0, 1]$ contiene todos los reales cuya representación D-aria empieza con $0.y_1 y_2 \dots y_{l_i}$



Todos estos intervalos son disjuntos por la condición de prefijo.
Su suma debe ser menor que la longitud del intervalo

$$\sum_{i=1}^{+\infty} D^{-l_i} \leq 1$$

Desigualdad de Kraft para códigos UD I

La clase de los códigos UD es más grande que la de los instantáneos, sin embargo no presentan ninguna ventaja respecto a la longitud de las palabras de código.

Teorema (McMillan)

El largo de las palabras de código l_i para un código C UD deben cumplir la desigualdad de Kraft.

$$\sum D^{-l_i} \leq 1$$

Igualmente dado un conjunto de largos de código que cumple la desigualdad, existe un código UD con esos largos.

Desigualdad de Kraft para códigos UD II

Demostración

Sea C^k la extensión de orden k de C. C^k es no singular. Por ser UD no hay más de D^n secuencias de largo n en C^k.

$$l(x^k) = l(x_1 x_2 \dots x_k) = \sum_{i=1}^{k} l(x_i)$$

Consideremos

$$\begin{aligned}
\left(\sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k &= \sum_{x_1 \in \mathcal{X}} \sum_{x_2 \in \mathcal{X}} \cdots \sum_{x_k \in \mathcal{X}} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)} \\
&= \sum_{x_1, x_2, \dots, x_k \in \mathcal{X}^k} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)} \\
&= \sum_{x^k \in \mathcal{X}^k} D^{-l(x^k)}
\end{aligned}$$

(cont.)

Desigualdad de Kraft para códigos UD III

Demostración

Reescribimos esta sumatoria en término de la longitud de las palabras

$$\sum_{x^k \in \mathcal{X}^k} D^{-l(x^k)} = \sum_{m=1}^{kl_{\text{máx}}} a(m) D^{-m}$$

donde $a(m)$ es el número de secuencias x^k con palabra de largo m, además $a(m) \leq D^m$.

$$\left(\sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k = \sum_{m=1}^{kl_{\text{máx}}} a(m) D^{-m} \leq \sum_{m=1}^{kl_{\text{máx}}} D^m D^{-m} = kl_{\text{máx}}$$

Entonces

$$\sum_j D^{-l_j} \leq (kl_{\text{máx}})^{\frac{1}{k}} \xrightarrow{k \uparrow \infty} 1$$

□

Desigualdad de Kraft para códigos UD IV

Corolario

Un código UD para un alfabeto fuente infinito también cumple la desigualdad de Kraft.

Ejemplo

Con los códigos analizados, y la fuente $p_X(\mathcal{X}) = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$ de $H(X) = 1.75$bits

$\mathcal{X}$	C_1	C_2	C_3	C_4	C_5
x_1	0	0	10	0	0
x_2	0	010	00	10	10
x_3	0	01	11	110	110
x_4	0	10	110	1110	111
K	2	1.125	0.875	0.9375	1
L	1	1.75	2.25	1.875	1.75
H/L	1.75	1	0.778	0.933	1

Códigos instantáneos óptimos (o cómo buscarlos)

Se puede plantear como un problema de optimización

$$\min_{l_1,l_2,\ldots,l_m} L = \sum_i p_i l_i \text{ restringido a (Kraft) } \sum_i D^{-l_i} = 1$$

$$J = \sum_i p_i l_i + \lambda \left(\sum_i D^{-l_i} - 1 \right) \quad \Rightarrow \quad D^{-l_i} = \frac{p_i}{\lambda \ln D}$$

Usando la restricción se llega a $\lambda = 1/\ln D$ y $p_i = D^{-l_i}$, dando los largos óptimos

$$l_i^* = -\log_D p_i$$

Con estos largos óptimos el largo medio queda

$$L^* = \sum_i p_i l_i^* = -\sum_i p_i \log_D p_i = H_D(X)$$

Estos largos óptimos no tienen que ser enteros, en la práctica sí.
¿Es un mínimo global?

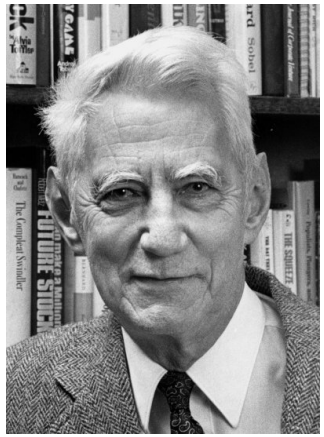
Primer Teorema de Shannon

Teorema (Teorema de Codificación de Fuente)

El largo medio de un código C instantáneo, D-ario para una variable aleatoria X es mayor o igual a la entropía de X

$$L(C) \geq H_D(X)$$

y la igualdad se cumple sii $p_i = D^{-l_i}$



- La igualdad se da sii $p_i = q_i = D^{-l_i}$ $(K = 1)$; distribución D-ádica.
- El procedimiento para encontrar el código óptimo sería hallar la distribución D-ádica más cercana a la distribución de X (en el sentido de la entropía relativa). Pero esto no siempre es fácil.
- Es una cota para la longitud media para la descripción de una fuente, "no se puede comprimir más allá de la entropía".

Teorema de Codificación de Fuente

Demostración

Sea $K = \sum_i D^{-l_i} \leq 1$ y $q_i = D^{-l_i}/K$ una distribución de probabilidad

$$\begin{aligned} D(p||q) &= \sum_i p_i \log_D \frac{p_i}{q_i} = \sum_i p_i \log_D p_i - \sum_i p_i \log_D q_i \\ &= -H_D(X) - \sum_i p_i \log_D \frac{D^{-l_i}}{K} \\ &= -H_D(X) - \sum_i p_i \log_D D^{-l_i} + \sum_i p_i \log_D K \\ &= -H_D(X) + \sum_i p_i l_i + \log_D K \sum_i p_i \\ &= -H_D(X) + L(C) + \log_D K \geq 0 \end{aligned}$$

La igualdad se da si $D(p||q) = 0$ y $K = 1$. □

Cotas para el largo medio

Qué pasa si $l_i^* = -\log_D p_i$ no es entero? Se toma el menor entero mayor

$$l_i = \left\lceil \log_D \frac{1}{p_i} \right\rceil \geq l_i^*$$

- Estos largos cumplen con la cota de Kraft

$$\sum_i D^{-\left\lceil \log_D \frac{1}{p_i} \right\rceil} \leq \sum_i D^{-log_D \frac{1}{p_i}} = \sum_i p_i = 1$$

- $\log_D \frac{1}{p_i} \leq l_i < \log_D \frac{1}{p_i} + 1$

$$H_D(X) \leq L < H_D(X) + 1$$

- El largo óptimo L^* también cumple con estas cotas

$$H_D(X) \leq L^* \leq L < H_D(X) + 1$$

Cotas para el largo medio

- Por lo menos 1 bit de exceso, ¿cómo "repartirlo"?

Consideremos un sistema que genera símbolos independientes. Y consideremos la secuencia $(x_1, x_2, \ldots, x_n) = x^n$ un símbolo de la fuente extendida $\mathcal{X}^n$

$$L_n = \frac{1}{n} \sum p(x_1, x_2, \ldots, x_n) l(x_1, x_2, \ldots, x_n) = \frac{1}{n} El(X_1, X_2, \ldots, X_n)$$

Aplicando la cota a esta fuente

$$H(X_1, X_2, \ldots, X_n) \leq El(X_1, X_2, \ldots, X_n) < H(X_1, X_2, \ldots, X_n) + 1$$

Al ser símbolos independientes: $H(X_1, X_2, \ldots, X_n) = nH(X)$.

$$H(X) \leq L_n < H(X) + \frac{1}{n}$$

Cotas para el largo medio

Si la secuencia proviene de un proceso estocástico estacionario

$$H(X_1, X_2, \ldots, X_n) \leq El(X_1, X_2, \ldots, X_n) < H(X_1, X_2, \ldots, X_n) + 1$$

$$\frac{H(X_1, X_2, \ldots, X_n)}{n} \leq L_n < \frac{H(X_1, X_2, \ldots, X_n)}{n} + \frac{1}{n}$$

recordando que $\lim_{n\uparrow\infty} \frac{H(X_1, X_2, \ldots, X_n)}{n} = H(\mathcal{X})$ es la tasa de entropía del proceso X_i

Teorema

El largo medio mínimo por símbolo para un proceso cumple

$$\frac{H(X_1, X_2, \ldots, X_n)}{n} \leq L_n^* < \frac{H(X_1, X_2, \ldots, X_n)}{n} + \frac{1}{n}$$

y si $(X_1, X_2, \ldots, X_n)$ *es estacionario con tasa de entropía* $H(\mathcal{X})$

$$L_n^* \longrightarrow H(\mathcal{X})$$

La mala distribución

¿Qué pasa si utilizamos una distribución $q(x) \neq p(x)$?
Sea $p(x)$ la distribución real y $q(x)$ la distribución usada para elegir el código C de largos $l(x) = \left\lceil \log \frac{1}{q(x)} \right\rceil$

Teorema

El largo medio con $p(x)$ para el código asignado con $l(x) = \left\lceil \log \frac{1}{q(x)} \right\rceil$ cumple

$$H(X) + D(p||q) \leq E_p l(X) < H(X) + D(p||q) + 1$$

$D(p||q)$ es el incremento en la complejidad de la descripción debido a «información incorrecta».

La mala distribución

Demostración

La clave $x \leq \lceil x \rceil < x + 1$

$$\begin{aligned}
E_p l(X) &= \sum_x p(x) \left\lceil \log \frac{1}{q(x)} \right\rceil \\
&< \sum_x p(x) \left(\log \frac{1}{q(x)} + 1 \right) \\
&= \sum_x p(x) \log \left(\frac{p(x)}{q(x)} \frac{1}{p(x)} \right) + 1 \\
&= \sum_x p(x) \log \frac{p(x)}{q(x)} + \sum_x p(x) \log \frac{1}{p(x)} + 1 \\
&= D(p||q) + H(X) + 1
\end{aligned}$$

□

Códigos de Shannon-Fano

Es un procedimiento suboptimal para construir un código, que alcanza una cota de

$$L(C) \leq H(X) + 2$$

- Ordenar las probabilidades en forma decreciente.
- Seleccionar k tal que $|\sum_{i=1}^{k} p_i - \sum_{i=k+1}^{m} p_i|$ sea mínima.
- Asignar un bit (diferente) a cada uno de los subconjuntos (cercanos a equiprobable) en que se divide la fuente.
- Repetir el procedimiento para todos los subconjuntos.

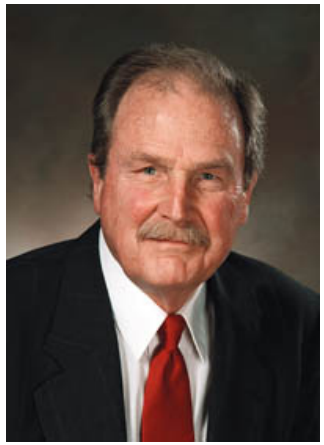
Ejemplo

p_i	1	2	3	c_i
0.25	0	0		00
0.25	0	1		01
0.2	1	0		10
0.15	1	1	0	110
0.15	1	1	1	111

Códigos de Huffman

Un código instantáneo óptimo (mínima $\sum p_i l_i$) para una distribución dada puede ser construido con un (simple) algoritmo propuesto por David A. Huffman en 1952.

Es un procedimiento recursivo que en cada paso agrupa los D símbolos menos probables para formar un nuevo símbolo.



Ejemplo

Para una fuente con $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5\}$, con probabilidades $p = \{0.25, 0.25, 0.2, 0.15, 0.15\}$ hallar el código de Huffman con $D = 2$ y $D = 3$.

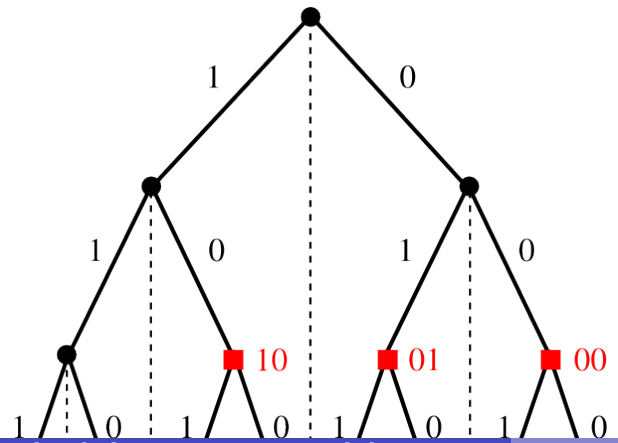
Códigos de Huffman

- Si $D \geq 3$ puede ser que no haya suficientes símbolos para combinar en cada iteración. En este caso se agregan símbolos "falsos" con probabilidad cero.
- En cada iteración se reducen en $(D-1)$ el número de símbolos. el número inicial de símbolos debe ser $1 + k(D-1)$ donde k es el número de niveles del árbol. Deben agregarse tantos símbolos falsos como sea necesario, pero cada uno de estos será una palabra del código que no se usará.

Comentarios sobre los códigos de Huffman

- Códigos de Huffman y número de preguntas.
  Para hallar el número óptimo de preguntas (con respuesta sí/no) para determinar un objeto (conociendo las probabilidades de los objetos) ¿cuál es la secuencia de preguntas más eficientes? El número medio de preguntas EQ siguiendo el esquema de Huffman cumple
  $$H(X) \leq EQ < H(X) + 1$$

- Códigos de Huffman y códigos alfabéticos.
  Las "preguntas" que responden los códigos de Huffman son del estilo "¿Es $X \in A$?". ¿Cómo usar Huffman para responder "óEs $X > a$?"
  Supongamos que los símbolos están ordenados por probabilidad $p_1 \geq p_2 \geq \cdots \geq p_m$, Huffman asigna largos $l_1 \leq l_2 \leq \cdots \leq l_m$. Luego creamos otro código con esos largos (óptimo) asignando las palabras "en orden" recorriendo los nodos libres del árbol.



$$\begin{aligned} x_1 : 01 &\rightarrow 00 \\ x_2 : 10 &\rightarrow 01 \\ x_3 : 11 &\rightarrow 10 \\ x_4 : 000 &\rightarrow 110 \\ x_5 : 001 &\rightarrow 111 \end{aligned}$$

Optimalidad de los códigos de Huffman

Probaremos el siguiente lema para un código instantáneo óptimo cualquiera. Asumiremos que las probabilidades están ordenadas $p_1 \geq p_2 \geq \cdots \geq p_m$.

Lema

Para toda distribución existe un código instantáneo óptimo (mínima $\sum p_i l_i$) que cumple las siguientes propiedades:

1. *Si $p_j > p_k$, entonces $l_j \leq l_k$*
2. *Las dos palabras de código más largas tienen el mismo largo*
3. *Las dos palabras de código más largas difieren en el último bit y corresponden a los dos símbolos menos probables.*

Optimalidad de los códigos de Huffman

Demostración

[Demostración de 1]

Sean C_m un código óptimo y C'_m un código igual que C_m pero con las palabras j y k intercambiadas, entonces

$$\begin{aligned} L(C'_m) - L(C_m) &= \sum p_i l'_i - \sum p_i l_i = p_j l_k + p_k l_j - p_j l_j - p_k l_k \\ &= (p_j - p_k)(l_k - l_j) \geq 0 \end{aligned}$$

Como $(p_j - p_k) > 0$ entonces $(l_k - l_j) \geq 0$ □

Optimalidad de los códigos de Huffman

Demostración

[Demostración de 2]

Supongamos que $C(x_\alpha)$ y $C(x_\beta)$ son las dos palabras mós largas, con largos l_α, tal que l_β y $l_\beta = l_\alpha + 1$.

Como es un código de prefijo, podemos quitarle el bit extra a $C(x_\beta)$ y tener un código más corto y que conserva la propiedad de prefijo.

Entonces las dos palabras más largas tienen igual largo y por la propiedad 1 son las menos probables. □

Optimalidad de los códigos de Huffman

Demostración

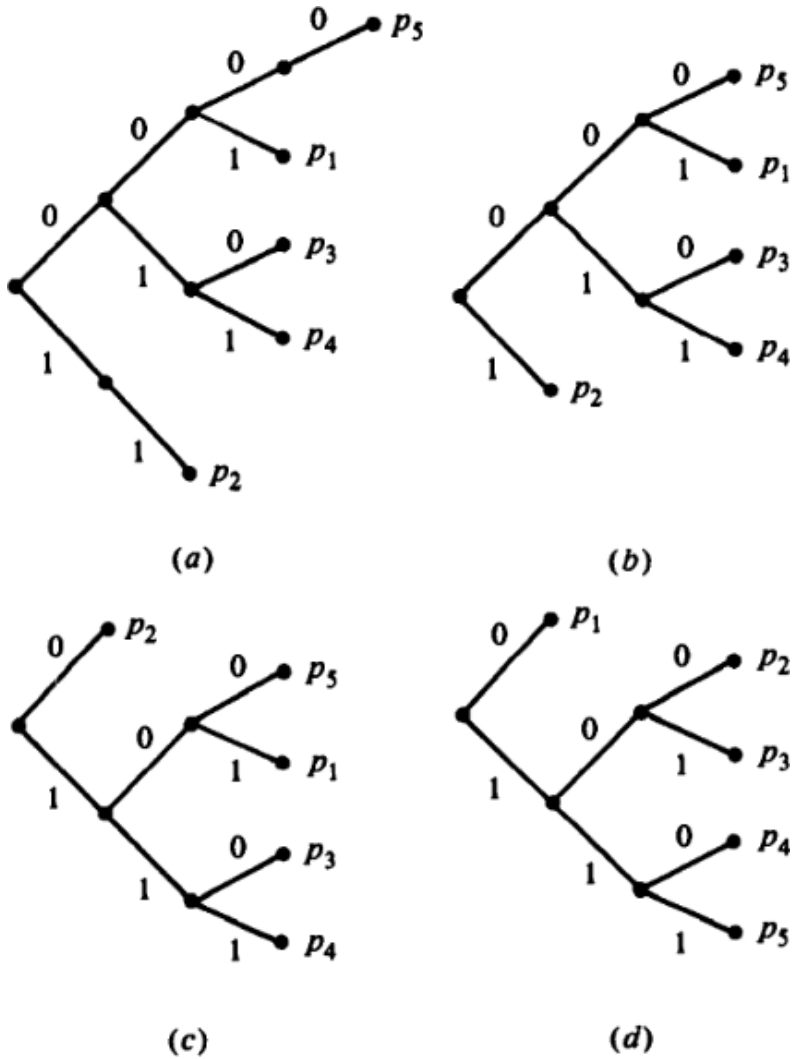
[Demostración de 3]

Si hay una palabra de código de longitud máxima que no esté en el mismo nivel del árbol que otra, se le puede quitar el último bit (igual que en 2), lo cual contradice que sea un código óptimo. Entonces toda palabra de largo código máximo tiene una «hermana» en el mismo nivel del árbol.

Se pueden intercambiar las palabras de código de estos dos símbolos y el largo medio no cambia. Por lo tanto estas dos palabras coinciden en todos los bits excepto en el último. □

Optimalidad de los códigos de Huffman

Si $p_1 \geq p_2 \geq \cdots \geq p_m$ existe un código óptimo con largos tal que $l_1 \leq l_2 \leq \cdots \leq l_{m-1} = l_m$ y $C(x_{m-1})$ y $C(x_m)$ difieren en el último bit.



Optimalidad de los códigos de Huffman

Dado C_m óptimo para m símbolos construiremos un código C_{m-1} para $m-1$ símbolos

- Agrupar las dos palabras de mayor longitud (menor probabilidad) en un nuevo símbolo de probabilidad $p_m + p_{m-1}$
- El resto de las palabras queda igual

El nuevo código queda así

$p(x)$	Palabra	Largo
p_1	$w'_1 = w_1$	$l'_1 = l_1$
p_2	$w'_2 = w_2$	$l'_2 = l_2$
$\vdots$	$\vdots$	$\vdots$
p_{m-2}	$w'_{m-2} = w_{m-2}$	$l'_{m-2} = l_{m-2}$
$p_{m-1} + p_m$	w'_{m-1}	$l'_{m-1} = l_{m-1} - 1$ $= l_m - 1$

además $w_{m-1} = w'_{m-1}0$ y $w_m = w'_{m-1}1$

Optimalidad de los códigos de Huffman

La relación entre los largos medios de los códigos queda

$$
\begin{aligned}
L(C_m) &= \sum_{i=1}^{m} p_i l_i \\
&= \sum_{i=1}^{m-2} p_i l'_i + p_{m-1}(l'_{m-1}+1) + p_m(l'_{m-1}+1) \\
&= \sum_{i=1}^{m-2} p_i l'_i + (p_{m-1}+p_m) l'_{m-1} + (p_{m-1}+p_m) \\
&= \sum_{i=1}^{m-1} p_i l'_i + (p_{m-1}+p_m) = L(C_{m-1}) + (p_{m-1}+p_m)
\end{aligned}
$$

Minimizar $L(C_m)$ es igual que minimizar $L(C_{m-1})$. Así se llega a un código de dos símbolos, el cual tiene una solución obvia ($C_2 = \{0, 1\}$).

Optimalidad de los códigos de Huffman

- A partir de un código óptimo C^*_{m-1} para la fuente de probabilidades $\mathbf{p}'$ encontramos un código óptimo C^*_m para la fuente de probabilidades $\mathbf{p}$. Probamos que

$$L(\mathbf{p}) = L^*(\mathbf{p}') + (p_{m-1} + p_m).$$

- Ahora

$$L(\mathbf{p}') = \sum_{i=1}^{m-2} p_i l'_i + (p_{m-1} + p_m)(l'_{m-1})$$

$$= \sum_{i=1}^{m-2} p_i l_i + p_{m-1}(l_{m-1} + 1) + p_m(l_m + 1)$$

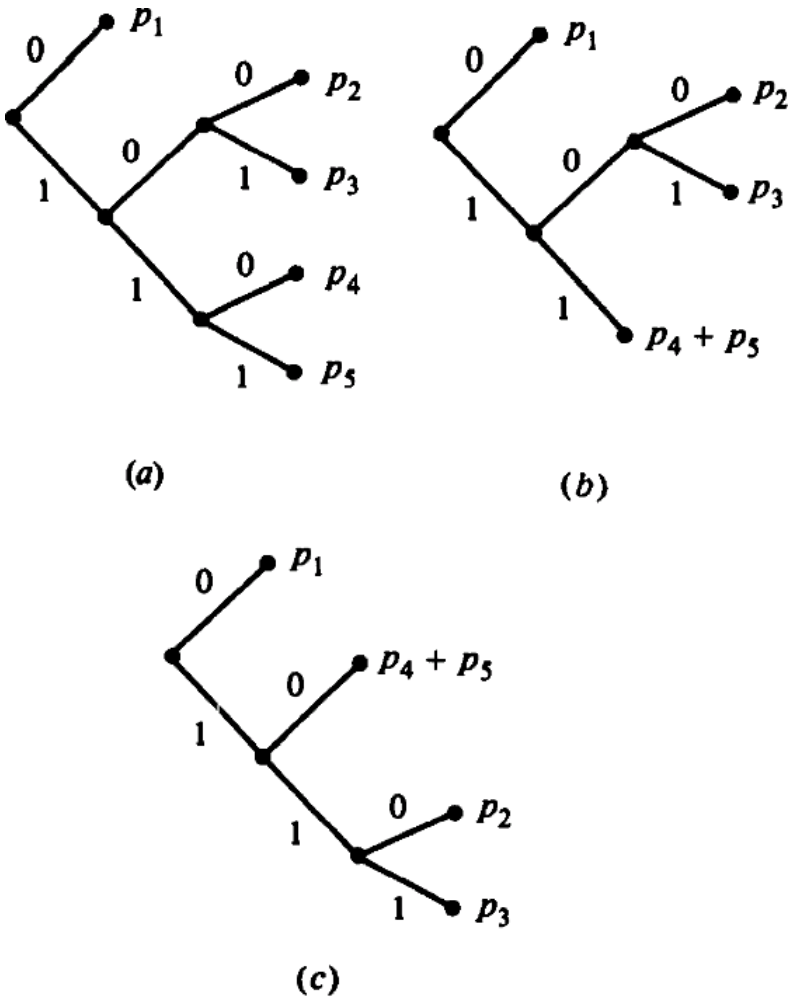
$$= \sum_{i=1}^{m} p_i l_i - (p_{m-1} + p_m) = L^*(\mathbf{p}) - (p_{m-1} + p_m)$$

- Tenemos que

$$\underbrace{(L(\mathbf{p}) - L^*(\mathbf{p}))}_{\geq 0} + \underbrace{\left(L(\mathbf{p}') - L^*(\mathbf{p}')\right)}_{\geq 0} = 0$$

$$\Rightarrow L(\mathbf{p}) = L^*(\mathbf{p}), L(\mathbf{p}') = L^*(\mathbf{p}')$$

Optimalidad de los códigos de Huffman



Al mantener la optimalidad en cada paso de la recursión el código obtenido es óptimo. Todo esto demuestra el siguiente

Teorema

Los códigos de Huffman son óptimos.

- Si C^* es un código de Huffman y C' otro código cualquiera para la misma fuente, entonces $L(C^*) \leq L(C')$.
- también es válido para un código D-ario.

Códigos de Huffman: desventajas

- largo entero $\rightarrow$ codificar en bloques $\rightarrow$ aumento de la complejidad
- variación de la distribución de probabilidad de la fuente (ej.: texto, imágenes)
  - adaptación de las probabilidades.
  - estimación a priori adaptada al mensaje.
  - implican "desperdicio" de bits, pero se usa.

Codigos de Shannon-Fano-Elias

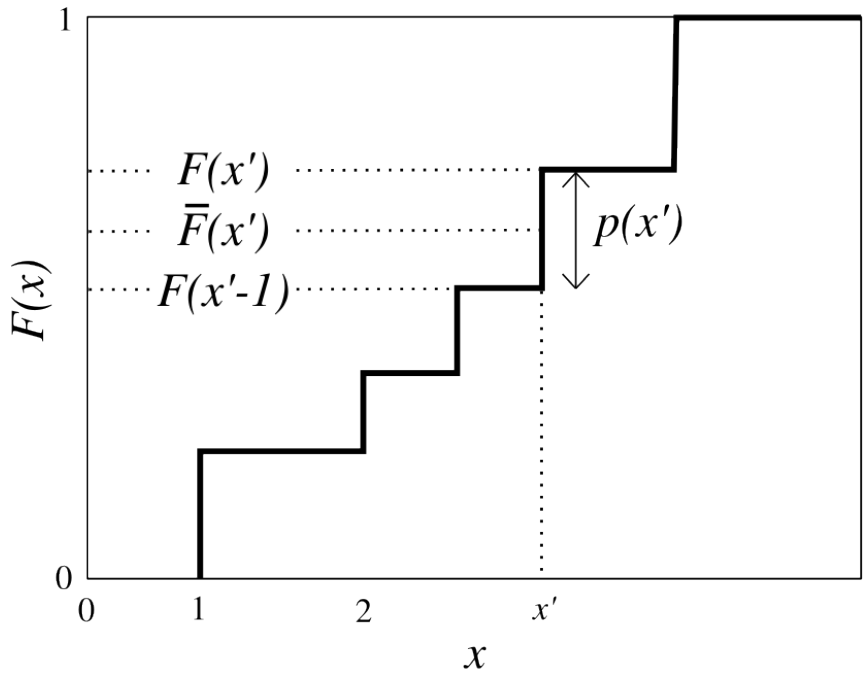
Es un procedimiento constructivo que utiliza la función de distribución acumulativa para asignar palabras de código.
Supongamos que $\mathcal{X} = \{x_1, x_2, \ldots, x_m\}$ y $p(x_i) > 0 \, \forall i = 1 \ldots m$. La función de distribución acumulativa es

$$F(x) = \sum_{a \leq x} p(a)$$

Usaremos una variante

$$\bar{F}(x) = \sum_{a < x} p(a) + \frac{1}{2} p(x)$$

- $\bar{F}(a) \neq \bar{F}(b)$ si $a \neq b$
- $\bar{F}(x)$ es un código para x



Codigos de Shannon-Fano-Elias

¿Con qué precisión debemos representar $\bar{F}(x)$ para que sea un código válido?

- Sea $\lfloor \bar{F}(x) \rfloor_{l(x)}$ el truncamiento de $\bar{F}(x)$ con $l(x)$ bits,

$$\bar{F}(x) - \lfloor \bar{F}(x) \rfloor_{l(x)} < \frac{1}{2^{l(x)}}$$

- Si $l(x) = \lceil -\log p(x) \rceil + 1$

$$\frac{1}{2^{l(x)}} < \frac{p(x)}{2} = \bar{F}(x) - F(x-1)$$

- Entonces $\lfloor \bar{F}(x) \rfloor_{l(x)}$ pertenece siempre al intervalo correspondiente a x.
- también es un código instantáneo; la palabra $w_1 w_2 \dots w_l$ representa el intervalo

$$[0.w_1 w_2 \dots w_l, 0.w_1 w_2 \dots w_l + 2^{-l})$$

y son disjuntos para todas las palabras.

Códigos de Shannon-Fano-Elias

Si $l(x) = \lceil -\log p(x) \rceil + 1$ la cota para el largo queda

$$L = \sum_x p(x) l(x) = \sum_x p(x) \left(\left\lceil \log \frac{1}{p(x)} \right\rceil + 1 \right) < H(X) + 2$$

Ejemplo

x_i	$p(x)$	$F(x)$	$\bar{F}(x)$	$\bar{F}(x)$ en binario	$l(x)$	palabra
x_1	0.25	0.25	0.125	0.00100	2+1	001
x_2	0.5	0.75	0.5	0.10000	1+1	10
x_3	0.125	0.875	0.8125	0.11010	3+1	1101
x_4	0.125	1	0.9375	0.11110	3+1	1111

- $L = 2.75$ bits, y $H(X) = 1.75$ bits. (Huffman alcanza la entropía.)
- El último bit de algunas palabras se puede sacar y mejorar el largo medio; pero si se quita el último bit de *todas* las palabras, no queda de prefijo.

Códigos de Shannon-Fano-Elias

Ejemplo

x_i	$p(x)$	$F(x)$	$\bar{F}(x)$	$\bar{F}(x)$ en binario	$l(x)$	palabra
x_1	0.25	0.25	0.125	0.001	3	001
x_2	0.25	0.25	0.375	0.011	3	011
x_3	0.2	0.7	0.6	0.1**0011**...	4	1001
x_4	0.15	0.85	0.775	0.110**0011**...	4	1100
x_5	0.15	1	0.925	0.111**0110**...	4	1110

- La entropía es $H(X) = 2.28$ bits. El largo medio queda $L(C) = 3.5$ bits, 1.2 bits mós que Huffman.

Codificación aritmética

- La codificación de Huffman es óptima, pero dado que las palabras deben tener largo entero hay una pérdida de eficiencia de hasta 1 bit.
- Codificar en bloques es la solución a esto. Sin embargo, la complejidad aumenta exponencialmente con el largo del bloque.
- En codificación aritmética se resuelve este problema codificando una secuencia de mensaje mediante un subintervalo de del intervalo $[0, 1]$.
- El ancho del subintervalo se va reduciendo a medida que aumenta el largo de la secuencia a transmitir. Esto permite emplear un sistema *secuencial*.
- Cómo representar el subintervalo correspondiente ésto basado en el esquema de codificación de Shannon-Fano-Elias.

Codificación aritmética

Teorema

Sea Y una v.a. con función de distribución de probabilidad continua $F(y)$. Sea $U = F(Y)$. Entonces U tiene distribución uniforme en [0,1].

Demostración

Dado que $F(y) \in [0, 1]$, el rango de U es [0,1]. también para $u \in [0, 1]$, vale

$$\begin{aligned} F_U(u) &= \Pr\{U \leq u\} \\ &= \Pr\{F(Y) \leq u\} \\ &= \Pr\left\{Y \leq F^{-1}(u)\right\} \\ &= F(F^{-1}(u)) \\ &= u \end{aligned}$$

□

Codificación aritmética

- Consideremos una secuencia infinita de v.a. $X_1, X_2, \ldots$ tomadas de un alfabeto finito $\mathcal{X} = 0, 1, 2, \ldots, m$.
- Una secuencia $x_1, x_2, \ldots$ generada de $\mathcal{X}$ puede considerarse como el número real $0.x_1x_2\ldots$ (en base $m+1$) entre 0 y 1.
- Llamemos $X = 0.X_1X_2\ldots$ a la v.a. real. La función distribución de X es

$$\begin{aligned} F_X(x) &= \Pr\{X \leq x = 0.x_1x_2\ldots\} \\ &= \Pr\{0.X_1X_2\cdots \leq 0.x_1x_2\ldots\} \\ &= \Pr\{X_1 < x_1\} + \Pr\{X_1 = x_1, X_2 < x_2\} + \ldots \end{aligned}$$

- Sea $U = F_X(X) = F_X(0.X_1X_2\ldots) = 0.F_1F_2\ldots$
- Si la distribución de las secuencia X^∞ no tiene átomos, entonces el lema anterior garantiza que $U \sim$ Uniforme$[0, 1]$. Por lo tanto, los bits de la expansión binaria de $F_1F_2\ldots$ tiene una distribución Bernoulli$(\frac{1}{2})$.
- Entonces, esta secuencia de bits es incompresible
- La dificultad de este planteo es estimar eficientemente la distribución acumulativa $F(\cdot)$

Codificación aritmética

Ejemplo

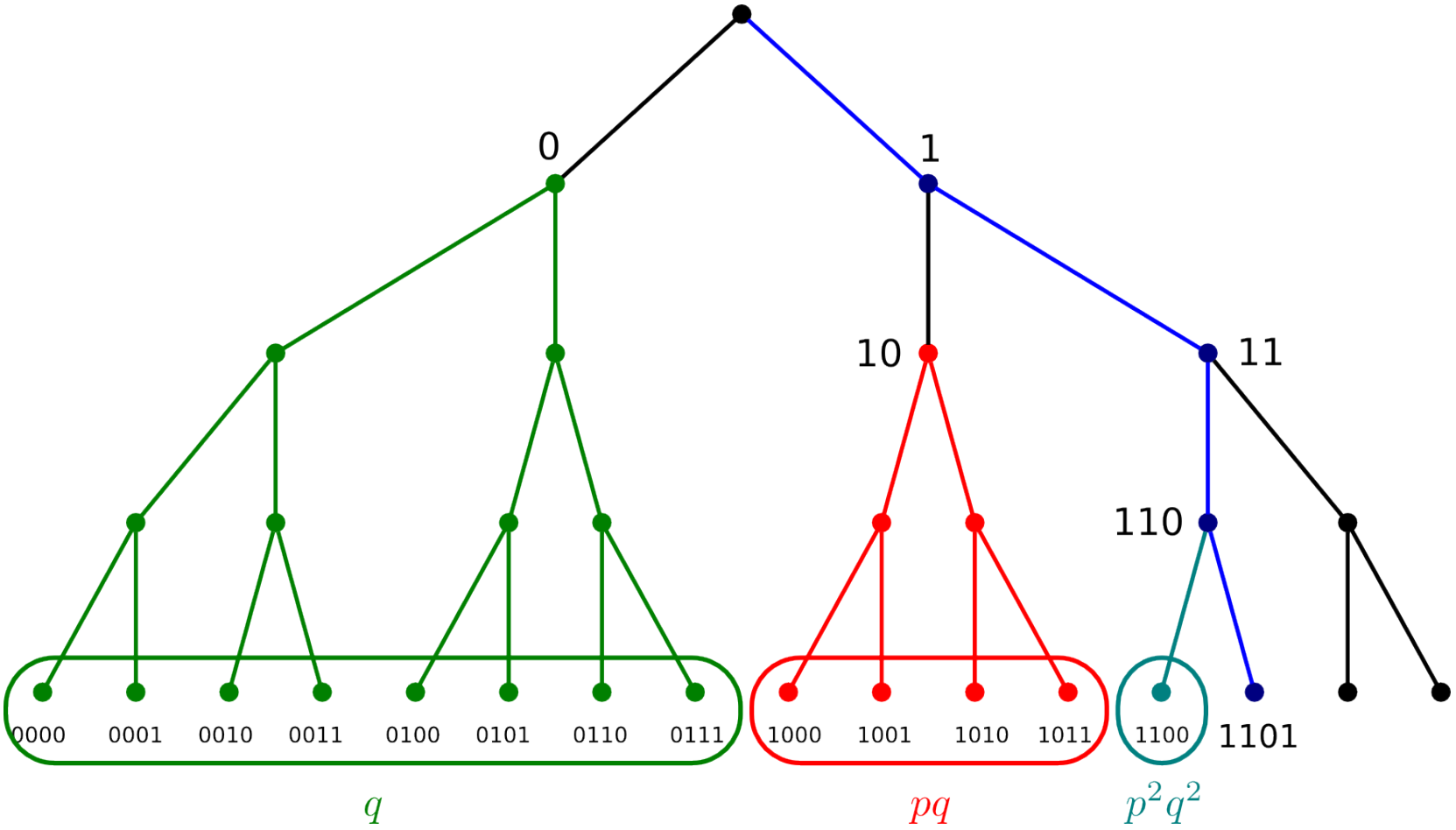
Sea $X_1 X_2 \ldots, X_n \sim$ Bernoulli(p). La secuencia $x^n = 1101$ se mapea en

$$\begin{aligned} F(x^n) &= \Pr\{X_1 < 1\} + \\ &\quad \Pr\{X_1 = 1, X_2 < 1\} + \\ &\quad \Pr\{X_1 = 1, X_2 = 1, X_3 < 0\} + \\ &\quad \Pr\{X_1 = 1, X_2 = 1, X_3 = 0, X_4 < 1\} \\ &= q + p \times q + p^2 \times 0 + p^2 q \times q \\ &= q + pq + p^2 q^2 \end{aligned}$$

- Cada término de la suma se puede calcular en función de términos previos, de la forma

$$F(x^n) = \sum_{k=1}^{n} p(x^{k-1}0) x_k$$

Codificación aritmética



Codificación aritmética: comentarios

- La idea esencial es la de Shannon-Fano-Elias y representar un símbolo $x^n = x_1 x_2 \dots x_n$ por un número en un subintervalo de [0,1] dado por $p(x^n)$ y $F(x^n)$ (estimadas eficientemente).

$$(0.w_1 w_2 \dots w_l, 0.w_1 w_2 \dots w_l + 2^{-l})$$

- Si se usa una precisión de $l(x^n) = \left\lceil \log \frac{1}{p(x)} \right\rceil$ bits **no garantiza** que sea un código de prefijo. Se usa $l(x^n) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$.
- Para hallar el intervalo hay que sumar las probabilidades de todas las secuencias $y^n < x^n$

$$F(x^n) = \sum_{y:y^n \leq x^n} p(y^n) = \sum_{k:x_k=1} p(x_1 x_2 \dots x_{k-1} 0)$$

- Para codificar el bit $i+1$ hay que estimar sólo $p(x^i x_{i+1})$ y $F(x^i x_{i+1}) = \sum_{k=1}^{i+1} p(x^{k-1}0)x_k$.

Codificación aritmética: comentarios

- El procedimiento depende del modelo que genera $p(x^n)$. Es sencillo calcular $p(x^n x_{n+1}) = p(x^n)$ en el caso de procesos i.i.d.

$$p(x^n) = \prod_{i=1}^{n} p(x_i)$$

y procesos Markovianos

$$p(x^n) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1})$$

- El decodificador también estima con el mismo procedimiento $p(x^n)$ y $F(x^n)$.
- Cuando la acumulación supera el código (como número $0.w_1 w_2 \ldots w_l$) se puede decodificar uno o más símbolos.
- Una variante más sofisticada es la Codificación Aritmética Adaptiva donde la distribución de la fuente varia y se adapta con el tiempo.

Codificación aritmética

Ejemplo

Queremos codificar la palabra COMPRESSOR.

Sómb.	C	E	M	O	P	R	S
Prob.	0.1	0.1	0.1	0.2	0.1	0.2	0.2
Interv.	0.0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.5-0.6	0.6-0.8	0.8-1.0

0.0 **0.1** 0.2 0.3 0.5 0.6 0.8 1.0
C E M O P R S

0.05=0.00001100110011001100110011...

0.0 0.01 0.02 **0.03** **0.05** 0.06 0.08 0.10
CC CE CM CO CP CR CS

0.04=0.00001010001111010111000010100 1...

0.03 0.032 **0.034** **0.036** 0.040 0.042 0.046 0.050
COC COE COM COO COP COR COS

0.035=0.00001000111101011100001010010 0...

0.034 0.0342 0.0344 0.0346 **0.0350** **0.0352** 0.0356 0.0360
COMC COME COMM COMO COMP COMR COMS

0.0351=0.00001000111111000101000001001 0...

- Siguiendo el procedimiento se llega a que la palabra COMPRESSOR se mapea en el intervalo $[0.0351279072, 0.0351279136)$.

- A medida que la secuencia es más grande el intervalo se achica y los primeros bits quedan fijos, por lo cual "ya se pueden transmitir".

Optimalidad competitiva de los códigos

Probamos que los códigos de Huffman son óptimos en media. Y vimos un ejemplo donde un código de Shannon asigna un código de menor longitud que Huffman para un símbolo particular (pero no en media).

Comparar los diferentes códigos respecto a la longitud que logran para una secuencia particular no es fácil, Huffman no tiene una fórmula cerrada para saber la longitud del código, Shannon só.

Teorema

Sea $l(x)$ la longitud de palabra asociada al código de Shannon y $l'(x)$ la longitud de palabra asociada por cualquier otro código UD, entonces

$$\Pr\left\{l(X) \geq l'(X) + c\right\} \leq 2^{1-c}$$

- La probabilidad que $l'(X)$ sea 5 bits más corta que $l(X)$ es menor a $\frac{1}{16} = 0.0625$

Optimalidad competitiva de los códigos

Demostración

$$
\begin{aligned}
\Pr\left\{l(X) \geq l'(X) + c\right\} &= \Pr\left\{\left\lceil \log \frac{1}{p(X)} \right\rceil \geq l'(X) + c\right\} \\
&\leq \Pr\left\{\log \frac{1}{p(X)} \geq l'(X) + c - 1\right\} \\
&= \Pr\left\{p(X) \leq 2^{-l'(X)-c+1}\right\} \\
&= \sum_{x:p(x)\leq 2^{-l'(x)-c+1}} p(x) \\
&\leq \sum_{x:p(x)\leq 2^{-l'(x)-c+1}} 2^{-l'(x)-c+1} \\
&\leq \sum_{x} 2^{-l'(x)} 2^{1-c} \leq 2^{1-c}
\end{aligned}
$$

□

Optimalidad competitiva de los códigos

Teorema

Para una distribución de probabilidad diádica $p(x)$, sea $l(x) = -\log p(x)$ la longitud de palabra asociada al código de Shannon y $l'(x)$ la longitud de palabra asociada por cualquier otro código unívocamente decodificable, entonces

$$\Pr\left\{l(X) < l'(X)\right\} \geq \Pr\left\{l(X) > l'(X)\right\}$$

La igualdad se da si y sólo si $l(X) = l'(X)$ para todo X.

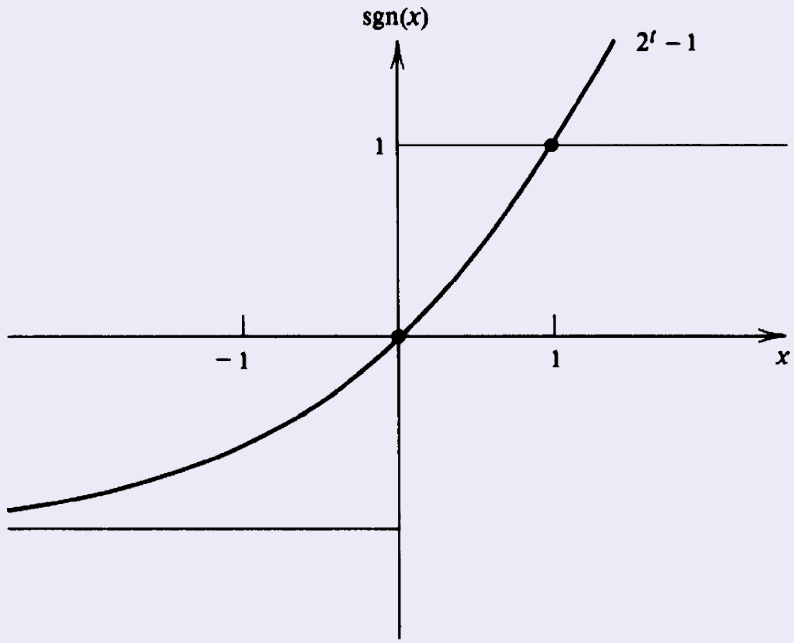
Optimalidad competitiva de los códigos

Demostración

Usaremos la función

$$\text{sgn}(t) = \begin{cases} 1 & \text{si } t > 0 \\ 0 & \text{si } t = 0 \\ -1 & \text{si } t < 0 \end{cases}$$

$$\text{sgn}(t) \leq 2^t - 1 \, \forall t \in \mathbb{N}$$



(cont.)

Optimalidad competitiva de los códigos

Demostración

$$
\begin{aligned}
\Pr\left\{l'(X) < l(X)\right\} \quad - \quad & \Pr\left\{l'(X) > l(X)\right\} = \\
= \quad & \sum_{x:l'(x)<l(x)} p(x) - \sum_{x:l'(x)>l(x)} p(x) \\
= \quad & \sum_{x} p(x)\mathsf{sgn}(l(x) - l'(x)) \\
\leq \quad & \sum_{x} p(x)(2^{l(x)-l'(x)} - 1) \\
= \quad & \sum_{x} 2^{-l(x)}(2^{l(x)-l'(x)} - 1) \\
= \quad & \sum_{x} 2^{-l'(x)} - \sum_{x} 2^{-l(x)} \\
= \quad & \sum_{x} 2^{-l'(x)} - 1 \leq 1 - 1 = 0
\end{aligned}
$$

□

Optimalidad competitiva de los códigos

Corolario

Para distribuciones de probabilidad no diádicas

$$E sgn(l(X) - l'(X) - 1) \leq 0$$

donde $l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil$ *y* $l'(x)$ *es el largo de cualquier otro código.*