Notas sobre Codificación Aritmética para el curso de Compresión de Datos sin Pérdida.

Álvaro Martín

29 de abril de 2021

1. Código de Shannon-Fano-Elias

Consideramos el orden lexicográfico para secuencias de largo n, y definimos

$$F_n(x^n) = \sum_{y^n < x^n} P_n(y^n)\,. \tag{1.1}$$

La idea para la construcción del código consiste en asociar a cada secuencia de largo n con probabilidad positiva, x^n, un número real en el intervalo $\Big[F_n(x^n), F_n(x^n) + P_n(x^n)\Big)$. Como todos los intervalos son disjuntos, entonces cada uno de estos números reales identifica unívocamente a una secuencia. Elegimos un número real que pueda representarse con precisión finita en base 2 y tomamos los bits de esta representación como una palabra de código. Específicamente, definimos

$$\bar{F}_n(x^n) = F_n(x^n) + \frac{1}{2}P_n(x^n)\,, \tag{1.2}$$

y codificamos x^n con los $\ell(x^n)$ dígitos después de la coma en la[1] representación binaria del número $\bar{F}_n(x^n)$, donde

$$\ell(x^n) = \lceil -\log P_n(x^n) \rceil + 1\,. \tag{1.3}$$

Cuando no existe ambigüedad, omitimos x^n en la notación $\ell(x^n)$. Denotamos $\lfloor \cdot \rfloor_\ell$ al número que resulta de truncar la representación en base dos de un real a ℓ dígitos a la derecha de la coma. Con esta notación, el número que representa a la secuencia x^n según nuestro código es

$$\lfloor \bar{F}_n(x^n) \rfloor_\ell\,. \tag{1.4}$$

Para probar que el Código de Shannon-Fano-Elias es unívocamente decodificable nos basamos en el siguiente lema.

Lema 1.1. *Para toda secuencia x^n se cumple*

$$\Big[\lfloor \bar{F}_n(x^n) \rfloor_\ell, \lfloor \bar{F}_n(x^n) \rfloor_\ell + 2^{-\ell}\Big) \subseteq \Big[F_n(x^n), F_n(x^n) + P_n(x^n)\Big)\,. \tag{1.5}$$

[1] Si un número tiene más de una representación binaria (una de largo finito y otra de largo infinito), puede usarse cualquiera de ellas consistentemente.

Demostración. Por definición de ℓ, tenemos que $\ell \geq -\log P_n(x^n) + 1$, o, equivalentemente,

$$2^{-\ell} \leq \frac{1}{2} P_n(x^n)\,. \tag{1.6}$$

En consecuencia, como además $\lfloor \bar{F}_n(x^n) \rfloor_\ell \leq \bar{F}_n(x^n)$, tenemos que

$$\lfloor \bar{F}_n(x^n) \rfloor_\ell + 2^{-\ell} \leq \bar{F}_n(x^n) + \frac{1}{2} P_n(x^n) = F_n(x^n) + P_n(x^n)\,. \tag{1.7}$$

Por otra parte, al truncar a ℓ bits la representación binaria de $\bar{F}_n(x^n)$ obtenemos un número que satisface

$$\bar{F}_n(x^n) - \lfloor \bar{F}_n(x^n) \rfloor_\ell \leq 2^{-\ell}\,, \tag{1.8}$$

donde la desigualdad puede no ser estricta si $\bar{F}_n(x^n)$ es de la forma $0.xyz11111\ldots$. Por lo tanto, usando (1.6), obtenemos

$$\lfloor \bar{F}_n(x^n) \rfloor_\ell \geq \bar{F}_n(x^n) - \frac{1}{2} P_n(x^n) = F_n(x^n)\,,$$

lo cual, junto con (1.7), prueba (1.5). □

Teorema 1.2. *El Código de Shannon-Fano-Elias es de prefijo y el largo medio está acotado superiormente por* $H(X^n) + 2$.

Demostración. La cota sobre el largo medio es consecuencia inmediata de (1.3). Por otra parte, si el código de x^n es prefijo del de y^n, entonces los primeros $\ell(x^n)$ dígitos a la derecha de la coma de $\bar{F}_n(x^n)$ y $\bar{F}_n(y^n)$ coinciden y, por lo tanto, debe cumplirse

$$\lfloor \bar{F}_n(x^n) \rfloor_{\ell(x^n)} \leq \lfloor \bar{F}_n(y^n) \rfloor_{\ell(y^n)} < \lfloor \bar{F}_n(x^n) \rfloor_{\ell(x^n)} + 2^{-\ell(x^n)}\,,$$

donde la desigualdad de la derecha es estricta, ya que las palabras de código son de largo finito. Como el lema 1.1 implica que los intervalos $\left[\lfloor \bar{F}_n(x^n) \rfloor_{\ell(x^n)}\,, \lfloor \bar{F}_n(x^n) \rfloor_{\ell(x^n)} + 2^{-\ell(x^n)} \right)$ son disjuntos para secuencias diferentes, debemos tener $x^n = y^n$. □

1.1. Cálculo de la palabra de código asociada a una secuencia

Definimos

$$C_n(a|x^{n-1}) = \sum_{b<a} P(b|x^{n-1}) \tag{1.9}$$

y la siguiente recurrencia

$$F_0 = 0\,, \tag{1.10}$$

$$F_n = F_{n-1} + C_n(x_n|x^{n-1}) P_{n-1}(x^{n-1})\,, \tag{1.11}$$

donde, por convención, $P_0(x^0) = 1$. Aplicando (1.11) recursivamente para reemplazar F_{n-1} en el lado derecho, obtenemos

$$F_n = F_{n-2} + C_{n-1}(x_{n-1}|x^{n-2}) P_{n-2}(x^{n-2}) + C_n(x_n|x^{n-1}) P_{n-1}(x^{n-1})\,, \tag{1.12}$$

y repitiendo n veces concluimos que

$$F_n = \sum_{i=1}^{n} C_i(x_i|x^{i-1}) P_{i-1}(x^{i-1})\,. \tag{1.13}$$

Observamos que, si $P(b|x^{i-1})$ se puede calcular "eficientemente" para $i = 1 \dots n$, entonces $C_n(x_i|x^{i-1})$ y $P_{i-1}(x^{i-1})$ también, lo cual permite calcular F_n eficientemente.

Proposición 1.3. *La recurrencia (1.10)-(1.11) calcula $F_n(x^n)$, es decir, $F_n(x^n) = F_n$.*

Demostración. Probamos por inducción que $F_n = \sum_{y^n<x^n} P_n(y^n)$. Claramente se cumple para $n = 1$; supongamos que $n > 1$ y que se cumple para $n - 1$.

$$\begin{aligned}
\sum_{y^n<x^n} P_n(y^n) &= \sum_{y^{n-1}<x^{n-1}, a\in\mathcal{A}} P_n(y^{n-1}a) + \sum_{a<x_n} P_n(x^{n-1}a) && (1.14)\\
&= \sum_{y^{n-1}<x^{n-1}} P_{n-1}(y^{n-1}) + P_{n-1}(x^{n-1}) \sum_{a<x_n} P(a|x^{n-1}) && (1.15)\\
&= F_{n-1}(x^{n-1}) + P_{n-1}(x^{n-1}) C_n(x_n|x^{n-1}) && (1.16)\\
&= F_{n-1} + P_{n-1}(x^{n-1}) C_n(x_n|x^{n-1}) && (1.17)\\
&= F_n && (1.18)
\end{aligned}$$

□

Ejemplo 1.4. *Calculamos la palabra de código para $x^n = 101$ y un modelo de Bernoulli con $P(0) = 1/4$.*

Por (1.10)-(1.11) tenemos

$$\begin{aligned}
F_1 &= C_1(1) = \frac{1}{4} && (1.19)\\
F_2 &= F_1 + 0 = \frac{1}{4} && (1.20)\\
F_3 &= F_2 + C_3(1|10) P_2(10) = \frac{1}{4} + \frac{1}{4}\frac{3}{4}\frac{1}{4}\,. && (1.21)
\end{aligned}$$

La representación en binario de estos términos es

$$\begin{aligned}
\frac{1}{4} = 2^{-2} &= 0{,}01 && (1.22)\\
\frac{1}{4}\frac{3}{4}\frac{1}{4} = 3 \times 2^{-6} &= 0{,}000011 && (1.23)
\end{aligned}$$

y sumando obtenemos

$$F_3 = 0{,}010011\,. \tag{1.24}$$

Por otra parte, tenemos

$$P(x^n) = \frac{3}{4}\frac{1}{4}\frac{3}{4} = 9 \times 2^{-6} = \left(2^3 + 2^0\right) 2^{-6}\,. \tag{1.25}$$

Por lo tanto, la representación en binario de $\frac{1}{2}P(x^n)$ *es*

$$\frac{1}{2}P(x^n) = \Big(2^3 + 2^0\Big)2^{-7} = 1001 \times 2^{-7} = 0{,}0001001\,. \tag{1.26}$$

Sumando (1.24) y (1.26) obtenemos

$$\bar{F}_n(x^n) = 0{,}0101111\,. \tag{1.27}$$

Para calcular el largo de código, partimos de (1.25) para obtener

$$-\log P(x^n) = 6 - \log 9\,,$$

de donde, como $2^3 < 9 < 2^4$*, obtenemos* $\lceil -\log P(x^n)\rceil = 3$ *y por lo tanto* $\ell = 4$*. Entonces, de (1.27) concluimos que*

$$\lceil \bar{F}_n(x^n)\rceil_\ell = 0{,}0101$$

y la palabra de código para x^n *es* 0101*.*

1.2. Decodificación

La decodificación también se puede resolver "eficientemente" en base a la siguiente recurrencia.

- Para $i = 0$, definimos G_0 como el número representado por la palabra de código recibida,

$$G_0 = \lfloor \bar{F}_n(x^n)\rfloor_\ell\,. \tag{1.28}$$

- Para $i > 0$, definimos

$$\tilde{x}_i = \text{máx}\left\{b \in \mathcal{A} : C_i(b|\tilde{x}^{i-1})P_{i-1}(\tilde{x}^{i-1}) \le G_{i-1}\right\}\,, \tag{1.29}$$

  y

$$G_i = G_{i-1} - C_i(\tilde{x}_i|\tilde{x}^{i-1})P_{i-1}(\tilde{x}^{i-1})\,, \tag{1.30}$$

donde el máximo en (1.29) es con respecto al orden lexicográfico sobre $\mathcal{A}$ y veremos, en la demostración del lema 1.6 más adelante, que es sobre un conjunto no vacío.

De forma similar a lo que hicimos para F_n, aplicando (1.30) recursivamente para reemplazar G_{i-1} en el lado derecho, obtenemos

$$G_i = G_{i-2} - C_{i-1}(\tilde{x}_{i-1}|\tilde{x}^{i-2})P_{i-2}(\tilde{x}^{i-2}) - C_i(\tilde{x}_i|\tilde{x}^{i-1})P_{i-1}(\tilde{x}^{i-1})\,, \tag{1.31}$$

y repitiendo i veces concluimos que

$$G_i = G_0 - \sum_{j=1}^{i} C_j(\tilde{x}_j|\tilde{x}^{j-1})P_{j-1}(\tilde{x}^{j-1})\,. \tag{1.32}$$

Para probar que mediante (1.28)-(1.30) efectivamente se reconstruye x^n, usamos el siguiente lema, que establece que los intervalos $\Big[F_i(x^i), F_i(x^i) + P_i(x^i)\Big)$ están "anidados" para prefijos de largo creciente de x^n.

Lema 1.5. *Para toda secuencia x^n y todo natural i, $i < n$, se cumple*

$$\Big[F_n(x^n), F_n(x^n) + P_n(x^n)\Big) \subseteq \Big[F_i(x^i), F_i(x^i) + P_i(x^i)\Big)\,.$$

Demostración. Alcanza con probarlo para $i = n - 1$; la tesis general surge de aplicar el resultado recursivamente para valores sucesivos de i.

Por un lado, como $\{F_j\}_{j=0\ldots n}$ es no decreciente, sabemos que $F_n(x^n) \geq F_{n-1}(x^n)$. Por otro lado, a partir de la definición de $F_n(x^n)$ en (1.1) obtenemos

$$\begin{aligned} F_{n-1}(x^{n-1}) + P_{n-1}(x^{n-1}) &= \sum_{y^{n-1}<x^{n-1}} P_{n-1}(y^{n-1}) + \sum_{a\in\mathcal{A}} P_n(x^{n-1}a) \\ &\geq \sum_{y^{n-1}<x^{n-1}} P_{n-1}(y^{n-1}) + \sum_{a\leq x_n} P_n(x^{n-1}a) \\ &= F_n(x^n) + P_n(x^n)\,. \end{aligned}$$

□

Proposición 1.6. *La secuencia $\tilde{x}^n$ definida por (1.28)-(1.30) coincide con x^n.*

Demostración. La prueba es por inducción. Para $i = 1$, la ecuación (1.29) se reduce a

$$\tilde{x}_1 = \text{máx}\left\{b \in \mathcal{A} : C_1(b) \leq \lfloor \bar{F}_n(x^n)\rfloor_\ell\right\}, \tag{1.33}$$

o, observando que por (1.1) y (1.9) se cumple $C_1(b) = F_1(b)$,

$$\tilde{x}_1 = \text{máx}\left\{b \in \mathcal{A} : F_1(b) \leq \lfloor \bar{F}_n(x^n)\rfloor_\ell\right\}. \tag{1.34}$$

Combinando los lemas 1.1 y 1.5, sabemos que se cumple

$$F_1(x_1) \leq \lfloor \bar{F}_n(x^n)\rfloor_\ell < F_1(x_1) + P_1(x_1)\,, \tag{1.35}$$

de donde concluimos que el máximo en (1.34) se obtiene para $b = x_1$.

Ahora, asumiendo que $\tilde{x}^{i-1} = x^{i-1}$, por (1.32) tenemos que

$$G_{i-1} = G_0 - \sum_{j=1}^{i-1} C_j(x_j|x^{j-1})P_{j-1}(x^{j-1}) = \lfloor \bar{F}_n(x^n)\rfloor_\ell - F_{i-1}\,, \tag{1.36}$$

donde la última igualdad surge de (1.13) y (1.28). Por lo tanto, la ecuación (1.29) se convierte en

$$\tilde{x}_i = \text{máx}\left\{b \in \mathcal{A} : F_{i-1} + C_i(b|x^{i-1})P_{i-1}(x^{i-1}) \leq \lfloor \bar{F}_n(x^n)\rfloor_\ell\right\}, \tag{1.37}$$

o, por (1.11),

$$\tilde{x}_i = \text{máx}\left\{b \in \mathcal{A} : F_i(x^{i-1}b) \leq \lfloor \bar{F}_n(x^n)\rfloor_\ell\right\}. \tag{1.38}$$

Nuevamente, combinando los lemas 1.1 y 1.5 obtenemos

$$F_i(x^{i-1}x_i) \leq \lfloor \bar{F}_n(x^n)\rfloor_\ell < F_i(x^{i-1}x_i) + P_i(x^{i-1}x_i)\,, \tag{1.39}$$

que implica que el máximo en (1.38) se alcanza para $b = x_i$. □

2. Codificación aritmética con cálculos en precisión acotada

El contenido de esta sección está basado en [1].

2.1. Descripción del algoritmo

- Alfabeto de entrada $\mathcal{A} = \{1 \dots M\}$. Salida en base $D \geq 2$.
- $Q_i[m] \simeq P(X_i = m | x^{i-1})$ con J dígitos de precisión: $Q_i[m] = \sum_{j=1}^{J} q_j D^{-j}$. Exigimos

$$0 < Q_i[m] < 1\,, \quad \text{y} \quad \sum_{m=1}^{M} Q_i[m] = 1\,.$$

  Q_i puede depender de x^{i-1} (distribución condicional del siguiente símbolo dado el pasado). En el caso del decodificador, la dependencia es con respecto a los símbolos decodificados, $\tilde{x}^{i-1}$, que veremos que coinciden con x^{i-1}.

- $C_i[m] = \sum_{j<m} Q_i[j]$ (acumulativa).
- $T = \sum_{j=0}^{K-1} t_j D^{-\tau-j}$: ancho del intervalo con K dígitos de precisión y representación de punto flotante.

$$T = t_0.t_1 t_2 \dots t_{K-1} \times D^{-\tau}\,, \quad t_0 > 0\,. \tag{2.1}$$

  Se cumple que

$$D^{-\tau} \leq T < D^{-(\tau-1)}\,. \tag{2.2}$$

Codificador

1. Calcular F_n y T_n:

$$\begin{aligned} F_0 &= 0\,, T_0 = 1 && (2.3)\\ F_i &= F_{i-1} + T_{i-1} C_i[x_i] && (2.4)\\ T_i &= \Big\lfloor T_{i-1} Q_i[x_i] \Big\rfloor_K && (2.5) \end{aligned}$$

2. $L = \tau_n + 1$
3. $W = \Big\lfloor F_n \Big\rfloor_L + D^{-L}$
4. Emitir L dígitos de W

Decodificador

1. Sea G_0 la secuencia de entrada truncada al máximo valor posible de L (ver Práctico).
2. Sea $U_0 = 1$.
3. Calcular $\tilde{x}_i$, $i = 1 \dots n$:

$$\begin{aligned} \tilde{x}_i &= \text{máx}\left\{ b \in \mathcal{A} : C_i[b] \leq \frac{G_{i-1}}{U_{i-1}} \right\} && (2.6)\\ G_i &= G_{i-1} - U_{i-1} C_i[\tilde{x}_i] && (2.7)\\ U_i &= \Big\lfloor U_{i-1} Q_i[\tilde{x}_i] \Big\rfloor_K && (2.8) \end{aligned}$$

4. Devolver $\tilde{x}^n$

Observación 2.1. Tomando logaritmo en (2.2), obtenemos $\tau_n \geq -\log_D T_n > \tau_n - 1$, de modo que $\tau_n = \lceil -\log_D T_n \rceil$ y, por lo tanto, el cálculo de L es análogo al cálculo de ℓ en el código de Shannon-Fano-Elias.

Observación 2.2. Solo se necesita calcular los J dígitos más significativos del cociente en (2.6).

De las recurrencias en el codificador y el decodificador, surge enseguida que, para $i = 0 \ldots n$, se cumple

$$F_i = \sum_{j=1}^{i} C_j[x_j] T_{j-1} \tag{2.9}$$

$$G_i = G_0 - \sum_{j=1}^{i} C_j[\tilde{x}_j] U_{j-1} \,. \tag{2.10}$$

2.2. Corrección

Lema 2.3. *Para todo par de enteros i, k tales que $i \geq k \geq 0$ se cumple que*

$$\Big[F_i, F_i + T_i\Big) \subseteq \Big[F_k, F_k + T_k\Big) \,. \tag{2.11}$$

Demostración. Para $i > 0$, de (2.4) obtenemos

$$F_i - F_{i-1} = T_{i-1} C_i[x_i] \,. \tag{2.12}$$

Sumando y restando $Q_i[x_i]$, obtenemos

$$F_i - F_{i-1} = T_{i-1}\Big(C_i[x_i] + Q_i[x_i] - Q_i[x_i]\Big) \tag{2.13}$$

$$= T_{i-1}\Big(C_i[x_i] + Q_i[x_i]\Big) - T_{i-1} Q_i[x_i] \,. \tag{2.14}$$

Ahora, como $C_i[x_i] + Q_i[x_i] \leq 1$ y, por (2.5), $T_{i-1} Q_i[x_i] \geq T_i$, concluimos que

$$F_i - F_{i-1} \leq T_{i-1} - T_i \,, \tag{2.15}$$

o, equivalentemente,

$$F_i + T_i \leq T_{i-1} + F_{i-1} \,. \tag{2.16}$$

Como además, por (2.4), tenemos que $F_i \geq F_{i-1}$, concluimos que $\Big[F_i, F_i + T_i\Big) \subseteq \Big[F_{i-1}, F_{i-1} + T_{i-1}\Big)$ y, siguiendo un razonamiento inductivo, $\Big[F_i, F_i + T_i\Big) \subseteq \Big[F_k, F_k + T_k\Big)$ para $i \geq k$. □

Lema 2.4. *Para todo i, $0 \leq i \leq n$, se cumple $G_0 \in \Big[F_i, F_i + T_i\Big)$.*

Demostración. En virtud del lema 2.3, alcanza con considerar el caso $i = n$.

El mensaje recibido por el decodificador consta de los L dígitos de W seguidos de otros dígitos correspondientes al siguiente mensaje. En consecuencia, se cumple que $W \leq G_0$. Como además F_n difiere de $\Big\lfloor F_n \Big\rfloor_L$ en una cantidad no mayor a D^{-L}, de la definición de W surge que $F_n \leq W$. Concluimos por lo tanto que $F_n \leq G_0$.

Por otra parte, los L dígitos más significativos de G_0 coinciden con los de W, implicando que

$$G_0 < W + D^{-L} \,, \tag{2.17}$$

donde la desigualdad es estricta, ya que G_0 se representa con una cantidad finita de dígitos. Además, por la definición de W, tenemos que

$$W \leq F_n + D^{-L}\,, \tag{2.18}$$

que combinado con (2.17) y recordando que $D \geq 2$ resulta en

$$G_0 < F_n + 2D^{-L} \leq F_n + D^{-(L-1)}\,. \tag{2.19}$$

Como $L - 1 = \tau_n$ y, por (2.2), se cumple $D^{-\tau_n} \leq T_n$, concluimos que $G_0 < F_n + T_n$. □

Teorema 2.5. *El algoritmo de decodificación reconstruye correctamente $\tilde{x}^n = x^n$.*

Demostración. Probaremos por inducción en i que, para cada $i = 1 \ldots n$, se cumple

$$U_i = T_i\,, \tag{2.20}$$

$$U_{i-1}C_i[x_i] \leq G_{i-1} < U_{i-1}\Big(C_i[x_i] + Q_i[x_i]\Big)\,. \tag{2.21}$$

Observar que (2.21) implica que en (2.6) el decodificador selecciona el símbolo $\tilde{x}_i = x_i$.

Comenzamos por el paso base con $i = 1$. Por el lema 2.4 tenemos que

$$F_1 \leq G_0 < F_1 + T_1\,. \tag{2.22}$$

Vemos entonces que (2.21) se cumple para $i = 1$, ya que $U_0 = 1$ y además, por (2.4), tenemos que $F_1 = C_1[x_1]$ y, por (2.5), se cumple $T_1 \leq Q_1[x_1]$. En consecuencia, el decodificador selecciona $\tilde{x}_1 = x_1$, y luego calcula $U_1 = \Big\lfloor U_0 Q_1[\tilde{x}_1] \Big\rfloor_K$, que coincide con T_1, ya que $U_0 = 1 = T_0$. El paso base está probado.

Para $i > 1$, asumimos que (2.20) y (2.21) se cumplen para todo $j < i$. Entonces, debemos tener $\tilde{x}_j = x_j$ para todo $j < i$ y por lo tanto, por (2.10), obtenemos

$$G_{i-1} = G_0 - \sum_{j=1}^{i-1} C_j[\tilde{x}_j]U_{j-1} = G_0 - \sum_{j=1}^{i-1} C_j[x_j]T_{j-1}\,,$$

que, por (2.9), se reduce a

$$G_{i-1} = G_0 - F_{i-1}\,.$$

A partir del lema 2.4 obtenemos entonces

$$G_{i-1} \in \Big[F_i - F_{i-1},\;\; F_i - F_{i-1} + T_i\Big)\,, \tag{2.23}$$

o, usando (2.9),

$$G_{i-1} \in \Big[T_{i-1}C_i[x_i],\;\; T_{i-1}C_i[x_i] + T_i\Big)\,. \tag{2.24}$$

Como, por (2.5), se cumple que $T_i \leq T_{i-1}Q_i[x_i]$, concluimos que

$$G_{i-1} \in \Big[T_{i-1}C_i[x_i],\;\; T_{i-1}(C_i[x_i] + Q_i[x_i])\Big)\,, \tag{2.25}$$

de donde se desprende (2.21) observando que $T_{i-1} = U_{i-1}$ por la hipótesis de inducción.

En consecuencia, el decodificador selecciona $\tilde{x}_i = x_i$, y luego calcula $U_i = \Big\lfloor U_{i-1}Q_i[\tilde{x}_i] \Big\rfloor_K$, que, por la hipótesis de inducción, coincide con T_i, lo cual demuestra (2.20). □

2.3. Tasa de compresión

Para analizar la tasa de compresión de un codificador aritmético con precisión acotada, usaremos el siguiente lema.

Lema 2.6. *Para todo $i = 1 \dots n$ se cumple*

$$T_{i-1} Q_i[x_i] \Big(1 - D^{1-K}\Big) \leq T_i \,, \tag{2.26}$$

donde recordamos que K es la cantidad de dígitos de precisión en la representación de T.

Demostración. Como, por (2.5) y la definición implícita de τ en (2.1), T_i coincide con $T_{i-1}Q_i[x_i]$ hasta los primeros $\tau_i + K - 1$ dígitos, se cumple

$$T_{i-1} Q_i[x_i] - T_i \leq D^{-(\tau_i + K - 1)} = D^{1-K} D^{-\tau_i} \,.$$

Como además, por (2.2), $D^{-\tau_i} \leq T_i$ y, por (2.5), $T_i \leq T_{i-1}Q_i[x_i]$, obtenemos

$$T_{i-1} Q_i[x_i] - T_i \leq D^{1-K} T_{i-1} Q_i[x_i] \,,$$

lo cual prueba el lema. □

Sea $Q(x^n)$ la probabilidad asignada a x^n por el modelo estadístico que alimenta al codificador aritmético, esto es $Q(x^n) = \prod_{i=1}^n Q_i[x_i]$. El siguiente teorema acota superiormente el largo de código para x^n en función de la precisión K usada para representar el ancho de los intervalos.

Teorema 2.7. *Si se usa una precisión de K dígitos para representar el ancho de los intervalos, el largo de código $L(x^n)$ para cualquier secuencia de largo n satisface*

$$\frac{L(x^n)}{n} \leq -\frac{\log_D Q(x^n)}{n} + \frac{2}{n} + \nu_D(K) \,, \tag{2.27}$$

donde $\nu_D(K) = -\log_D \Big(1 - D^{1-K}\Big)$ decrece exponencialmente con K.

Demostración. Tomando logaritmos sobre (2.26) obtenemos

$$\log_D T_{i-1} - \log_D T_i \leq -\log_D Q_i[x_i] + \nu_D(K) \,. \tag{2.28}$$

Sumando para $i = 1 \dots n$, obtenemos una suma telescópica del lado izquierdo que, recordando que $T_0 = 1$, resulta en

$$-\log_D T_n \leq -\log_D Q(x^n) + n\nu_D(K) \,, \tag{2.29}$$

donde en la evaluación de la suma del lado derecho usamos que $Q(x^n) = \prod_{i=1}^n Q_i[x_i]$. Como, por la definición de L en el codificador y la observación 2.1, sabemos que $L \leq -\log_D T_n + 2$, (2.29) implica (2.27).

Sólo resta probar que $\nu_D(K) = O\Big(D^{-K}\Big)$. Claramente, como un cambio de base de logaritmo consiste en multiplicar por una constante y $D^{-K} = \frac{1}{D} D^{1-K}$, alcanza con probar que existen constantes positivas C, K_0, tales que

$$-\ln \Big(1 - D^{1-K}\Big) \leq C D^{1-K} \,, \quad \text{para todo } K > K_0 \,. \tag{2.30}$$

Equivalentemente, con el cambio de variable $Z = D^{1-K}$, probaremos que existen constantes positivas C, Z_0, con $Z_0 < 1$, tales que

$$-\ln\Big(1-Z\Big) \le CZ\,, \quad \text{para todo } Z,\ 0 \le Z < Z_0\,. \tag{2.31}$$

Observar que aplicando el cambio de variable inverso, $K = 1 - \log_D Z$, y tomando $K_0 = 1 - \log_D Z_0$ que es positivo porque $Z_0 < 1$, la condición $0 \le Z < Z_0$ en (2.31) es equivalente a la condición $\infty \ge K > K_0$ en en (2.30).

Multiplicando por -1 y tomando exponenciales en ambos lados de (2.31), obtenemos la expresión equivalente

$$1 - Z - e^{-CZ} \ge 0\,, \quad \text{para todo } Z,\ 0 \le Z < Z_0\,. \tag{2.32}$$

La función $g(z) = 1 - z - e^{-Cz}$ tiene derivada $g'(z) = -1 + Ce^{-Cz}$, que en cero evalúan a $g(0) = 0$ y $g'(0) = C - 1$. Eligiendo $C > 1$, la derivada en cero es positiva y, como g es continua en $z \ge 0$, existe $z_0 > 0$ tal que $g(z)$ es positiva en $(0, z_0)$. Tomando $Z_0 \in (0, \text{mín}\{1, z_0\})$ el teorema queda probado. □

El siguiente corolario es una consecuencia inmediata del teorema 2.7.

Corolario 2.8. *Sea P una distribución de probabilidad sobre $\mathcal{A}^n$ y X^n una secuencia aleatoria generada según P. Entonces, el largo medio de código normalizado satisface*

$$\frac{E\,[L(X^n)]}{n} \le \frac{H_D(X^n)}{n} + \frac{D_D(P||Q)}{n} + \frac{2}{n} + \nu_D(K)\,, \tag{2.33}$$

donde Q es la distribución de probabilidad sobre $\mathcal{A}^n$ asignada por el modelo estadístico que alimenta al codificador aritmético, esto es $Q(x^n) = \prod_{i=1}^n Q_i[x_i]$ para toda secuencia $x^n \in \mathcal{A}^n$.

El siguiente teorema completa la caracterización de cómo influye la precisión aritmética sobre la tasa de compresión.

Teorema 2.9. *Sea P una distribución de probabilidad sobre $\mathcal{A}^n$ tal que todas las secuencias tienen probabilidad positiva y sea*

$$P_{\text{mín}} = \min_{x^i \in \mathcal{A}^i, 1 \le i \le n} \{P(x_i|x^{i-1})\}\,.$$

Entonces, para todo δ, $0 < \delta < 1$, si la cantidad de dígitos de precisión usados en la representación de $Q_i[m]$ satisface $J \ge \lceil -\log_D P_{\text{mín}} - \log_D \delta + 1\rceil$, existe una elección de $Q_i[\cdot]$ tal que $\frac{D_D(P||Q)}{n} < \delta$.

Demostración. Dado $i < n$ y una secuencia arbitraria x^{i-1}, tomamos $Q_i[a] = \lfloor P(a|x^{i-1})\rfloor_J$ para todo $a < M$, y $Q_i[M] = 1 - \sum_{a<M} Q_i[a]$ (observar que este número se puede representar exactamente con J dígitos de precisión). Notar que $Q_i[a] \le P(a|x^{i-1})$ para todo $a < M$ y, como tanto $Q_i[\cdot]$ como $P(\cdot|x^{i-1})$ suman 1, debemos tener $Q_i[M] \ge P(M|x^{i-1})$.

Sea ahora a un símbolo cualquiera, $a < M$, y denotemos $q = Q_i[a]$, $p = P(a|x^{i-1})$. Nuestra elección de $Q_i[\cdot]$ garantiza que $p - q \le D^{-J}$ y, por la hipótesis sobre J, tenemos que

$$p - q \le P_{\text{mín}} \frac{\delta}{D} \le p\frac{\delta}{D}\,. \tag{2.34}$$

Observar que (2.34) implica que $q \ge p(1 - \frac{\delta}{D}) > 0$, por lo cual nuestra elección de $Q_i[\cdot]$ satisface que todas las probabilidades son positivas. Probaremos que

$$\frac{\delta}{D} \le 1 - D^{-\delta}\,, \tag{2.35}$$

de modo que (2.34) nos lleva a

$$p - q \leq p\Big(1 - D^{-\delta}\Big) = p - pD^{-\delta}\,,$$

de donde, simplificando y reagrupando, obtenemos

$$\frac{p}{q} \leq D^{\delta}\,.$$

En consecuencia, para toda secuencia x^n tenemos que $\frac{P(x^n)}{Q(x^n)} \leq D^{n\delta}$, de donde el teorema surge inmediatamente.

Para probar (2.35) veremos que $f(\delta) = D - D^{1-\delta} - \delta$ es no negativa en $0 < \delta < 1$. La derivada segunda de f es $f''(\delta) = -(\ln D)^2 D^{1-\delta} < 0$, por lo cual f es cóncava. Por lo tanto, como $f(0) = 0$ y $f(1) = D - 2 \geq 0$, en efecto debemos tener $f(\delta) \geq 0$ en $0 < \delta < 1$.

□

Referencias

[1] Richard Pasco. *Source Coding Algorithms for Fast Data Compression.* PhD thesis, Stanford University, 1976.