Capítulo 3

Teoría de la Información. Canales sin ruido

Un problema fundamental en el estudio de los sistemas de comunicación es el de establecer un modo de medir la información que éstos utilizan o procesan con la finalidad de mejorar la eficacia de la transmisión: aumentar la capacidad del canal y/o minimizar los efectos adversos del ruido que vician la transmisión. A partir del ya citado trabajo de Shannon (1948), la medida de la información de un mensaje es totalmente independiente de su contenido semántico, es decir, independiente de lo que el mensaje "dice"; dicha medida depende esencialmente de la probabilidad de producción del mismo por parte de alguna fuente de información. Las dos preguntas fundamentales a las que da respuesta la teoría de la información son las siguientes: 1) ¿cuál es el límite en la compresión de datos? y 2) ¿cuál es el límite de la tasa de transmisión en una comunicación? En este capítulo vamos a ver la respuesta a la primera pregunta a partir del concepto fundamental de entropía de la fuente.

3.1. Medida de la información

Consideremos una fuente $S = \{a_1, .., a_n\}$ con probabilidades $P = \{p_1, ..., p_n\}$. Deseamos definir el concepto de medida de la información que aporta a_k, que

denotaremos por $I(a_k)$. Parece razonable admitir que la cantidad de información que aporta un mensaje, desde un punto de vista a priori, representa una medida de la incertidumbre sobre el hecho de que se produzca el mensaje en cuestión. Por ello, asumimos que $I(a_k)$ depende de p_k, es decir, $I(a_k) = f(p_k)$, donde f es una función a determinar. Para establecer la forma de la función f, vamos a establecer algunas propiedades evidentes que debería cumplir.

1) Si la probabilidad p_k es cercana a 1, es casi seguro que debe ocurrir a_k. Si ocurre a_k, no hay gran sorpresa. Por el contrario, si p_k es cercana a 0, es casi seguro que a_k no ocurrirá. Por tanto, si llega a ocurrir, la sorpresa es grande. Por ello, es razonable exigir que f sea decreciente.

2) Supongamos ahora que queremos saber la incertidumbre $I(a_1, a_2)$ sobre el hecho de que la fuente emita el mensaje a_1a_2.

Si los sucesos son mutuamente excluyentes, la incertidumbre de $a_1\, a_2$ debe ser la adición de las incertidumbres aisladas, es decir, $I(a_1) + I(a_2)$. Como la probabilidad de $a_1\, a_2$ es $p_1 \cdot p_2$, se deduce que

$$I(p_1 \cdot p_2) = I(p_1) + I(p_2).$$

Por tanto, si ponemos $I(p) = f(p)$, la función f verifica las propiedades siguientes:

a) f es una función definida en $(0, 1]$ que debe ser decreciente.

b) $f(x \cdot y) = f(x) + f(y)$.

El lema siguiente prueba que las funciones que verifican las condiciones anteriores tienen una forma muy concreta.

Lema 3.1.1. *Las únicas funciones que cumplen las propiedades a) y b) anteriores tienen la forma $f(x) = k \log x$, siendo k una constante negativa .*

DEMOSTRACIÓN: En primer lugar, nótese que de la condición b) se sigue que

$$f(x) = f(x \cdot 1) = f(x) + f(1).$$

Entonces debe ser $f(1) = 0$. Usando ahora a) deducimos que f es no negativa. Escogemos $x_0 \in (0,1)$ de modo que $f(x_0) \neq 0$. Como la sucesión (x_0^m) es convergente monótonamente a 0, dados cualesquiera $x \in (0,1)$ y $n \in \mathbb{N}$, existe un entero m de modo que

$$x_0^{m+1} \leq x^n < x_0^m.$$

Por el decrecimiento de f, se sigue que

$$f(x_0^m) < f(x^n) \leq f(x_0^{m+1}).$$

En virtud de la condición b), de las desigualdades anteriores se sigue que

$$mf(x_0) < nf(x) \leq (m+1)f(x_0).$$

Si dividimos todos los miembros por $nf(x_0)$, resulta

$$\frac{m}{n} < \frac{f(x)}{f(x_0)} \leq \frac{m+1}{n}. \tag{3.1}$$

La función $g(x) = -\log x$ verifica las condiciones a) y b), por tanto, debe verificar las desigualdades (3.1), es decir, se tiene

$$\frac{m}{n} < \frac{\log x}{\log x_0} \leq \frac{m+1}{n}. \tag{3.2}$$

De (3.1) y (3.2) se sigue

$$\left| \frac{f(x)}{f(x_0)} - \frac{\log x}{\log x_0} \right| \leq \frac{1}{n}.$$

La desigualdad anterior es válida, manteniendo fijo $x \in (0,1)$, para todo natural n. Por tanto, se concluye que

$$\frac{f(x)}{f(x_0)} = \frac{\log x}{\log x_0},$$

para cualquier valor de $x \in (0,1)$, lo que prueba que el cociente $f(x)/\log x$ es constante e igual a $k = f(x_0)/\log x_0$ (nótese que $k < 0$). □

Esta es la razón de definir el grado de incertidumbre como sigue

Definición 3.1.2. *(Grado de incertidumbre). Se define el grado de incertidumbre sobre la ocurrencia de a_k por*

$$I(a_k) = \log\left(\frac{1}{p_k}\right) = -\log p_k.$$

El grado de incertidumbre es un concepto a priori de la ocurrencia de un suceso. Una vez que sabemos que la fuente ha producido a_k, podemos considerar $I(a_k) = -\log p_k$ como la cantidad de información que nos aporta.

Si se escoge la base del logaritmo como 2, entonces la unidad de información se llama bit. Para comprender qué significa esta unidad, consideremos una fuente binaria $S = \{a_1, a_2\}$ con ambos símbolos equiprobables. 1 bit es el grado de incertidumbre correspondiente a cada uno de los símbolos equiprobables y excluyentes: $1bit = I(a_1) = I(a_2) = -\log_2 \frac{1}{2}$. Por esta razón, en lo que sigue $\log x$ denotará el logaritmo en base 2 (la información se mide en hartleys o en nats, si la base del logaritmo es 10 ó e).

3.2. Entropía

El término entropía fue usado por primera vez por Clausius en 1864. El primero en introducir el término en la teoría de la información fue Claude Shannon en 1948.

Consideramos una fuente $S = \{a_1, .., a_n\}$ con la distribución de probabilidades $P = \{p_1, .., p_n\}$. Se llama **entropía** de la fuente a la media ponderada de los grados de incertidumbre de los símbolos a_k:

$$H(S) = \sum_{k=1}^{n} p_k \log \frac{1}{p_k} = -\sum_{k=1}^{n} p_k \log p_k.$$

En la expresión anterior se entiende que $0 \cdot \log 0$ es igual a 0. Por tanto, añadir términos con probabilidad nula no cambia el valor de la entropía. De la definición anterior, se sigue que la entropía es un promedio de la información

que proporcionan los símbolos fuente o, a priori, la incertidumbre media antes de que la fuente emite un símbolo.

Nótese que la entropía es función de la distribución de probabilidades. Por ello, también suele denotarse por $H(p_1, .., p_n)$.

Ejemplos 3.2.1. 1) Consideremos la fuente $S = \{a_1, a_2, .., a_8\}$ y supongamos que las probabilidades son:

$$1/2, 1/4, 1/8, 1/16, 1/64, 1/64, 1/64, 1/64.$$

La entropía viene dada por

$$H(S) = -(1/2)\log(1/2) - (1/4)\log(1/4) - (1/8)\log(1/8) - (1/16)\log(1/16) - \\ -(4/64)\log(1/64) = 2\text{bits}.$$

Si deseamos codificar los símbolos fuente con cadenas binarias de igual longitud, debemos emplear cadenas de longitud 3. Ahora bien, como las probabilidades son diferentes, parece más adecuado usar descripciones más cortas para los símbolos más probables y más largas para los otros. De esta forma la media de las longitudes será menor. Por ejemplo, podríamos usar la siguiente descripción:

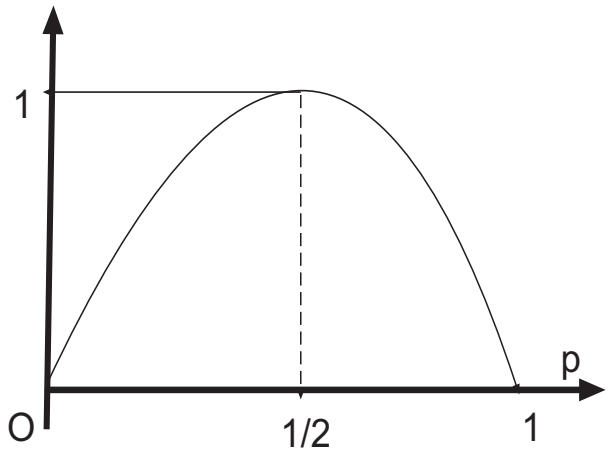
símbolo	1	2	3	4	5	6	7	8
Etiqueta	0	10	110	1110	111100	111101	111110	111111

Nótese que la media de las longitudes de las etiquetas es 2 bits, lo que coincide con el valor de la entropía. En el siguiente apartado veremos que esto no es una mera coincidencia.

2) Un ejemplo importante lo constituye la función

$$H(p) = p\log\big(\frac{1}{p}\big) + (1-p)\log\big(\frac{1}{1-p}\big).$$

Se trata de la entropía que corresponde a una fuente binaria con probabilidades p y $1-p$. En la figura siguiente puede verse la representación gráfica de esta función.



Necesitamos el siguiente

Lema 3.2.2. *(Gibbs). Sea* $\{p_k\}_{k=1,..,n}$ *una distribución de probabilidades y* $\{q_k > 0\}_{k=1,..,n}$ *una sucesión de valores positivos tales que* $\sum q_k \leq 1$*, entonces*

$$\sum_{k=1}^{n} p_k \log \frac{1}{p_k} \leq \sum_{k=1}^{n} p_k \log \frac{1}{q_k}. \tag{3.3}$$

La igualdad se da sólo si $p_k = q_k$*, para cada* $k = 1, .., n$*.*

DEMOSTRACIÓN: Como $\log_2 x = \frac{\log_e x}{\log_e 2}$, basta probar el lema para el logaritmo neperiano. Por tanto, en la demostración suponemos que $\log x$ denota el logaritmo neperiano. Vamos a usar la desigualdad

$$\log x \leq x - 1, \tag{3.4}$$

válida para $x \in (0, +\infty)$. En efecto, ponemos $f(x) = \log x - x + 1$ y calculamos su derivada $f'(x) = \frac{1}{x} - 1$. El signo de $f'(x)$ viene dado por

$$\text{sig}(f'(x)) = \begin{cases} + & \text{si} \quad 0 < \text{x} < 1 \\ - & \text{si} \quad \text{x} > 1 \end{cases}$$

Por tanto, f es estrictamente creciente en $(0, 1]$ y estrictamente decreciente en $[1, +\infty)$, de donde se obtiene que $f(x) < f(1) = 0$, para cada $x \in (0, 1)$ y $f(1) = 0 > f(x)$, para $x > 1$. En virtud de la desigualdad, se tiene:

$$\sum_{k=1}^{n} p_k \log \frac{q_k}{p_k} \leq \sum_{k=1}^{n} p_k (\frac{q_k}{p_k} - 1).$$

De donde sigue

$$\sum_{k=1}^{n} p_k \log q_k - \sum_{k=1}^{n} p_k \log p_k \leq \sum q_k - \sum p_k \leq 1 - 1 = 0.$$

Finalmente, nótese que por (3.4), para cada $k = 1, 2, .., n$, se verifica la desigualdad

$$p_k \log \frac{q_k}{p_k} \leq p_k (\frac{q_k}{p_k} - 1).$$

Entonces la igualdad

$$\sum_{k=1}^{n} p_k \log \frac{q_k}{p_k} = \sum_{k=1}^{n} p_k (\frac{q_k}{p_k} - 1)$$

sólo es posible si cada sumando del primer miembro es igual al correspondiente del segundo. Es decir, sii

$$\log \frac{q_k}{p_k} = (\frac{q_k}{p_k} - 1).$$

Pero $\log x = x - 1$ sólo en el caso de que $x = 1$. Por tanto, hemos probado que se da la igualdad en (3.3) cuando $p_k = q_k$, para todo $k = 1, .., n$. □

Teorema 3.2.3. *En las condiciones y notaciones anteriores, se tiene que* $0 \leq H(S) \leq \log card(S)$.

DEMOSTRACIÓN: a) Es claro que $H(S) \geq 0$ por ser una suma de sumandos no negativos $p_k \log \frac{1}{p_k}$.

b) $H(S) \leq \log \text{card}(S)$. Sea $q_k = \frac{1}{\text{card}(S)}$, para cada $k = 1, .., n$. y apliquemos el Lema de Gibbs:

$$H(S) = \sum_{k=1}^{n} p_k \log \frac{1}{p_k} \leq \sum_{k=1}^{n} p_k \log \text{card}(S) = \log \text{card}(S) \sum p_k = \log \text{card}(S).$$

□

El Corolario siguiente nos informa de cuándo se alcanzan las cotas,

Corolario 3.2.4. a) $H(S) = 0$ si y sólo si S consta de un sólo elemento.
b) $H(S) = \log \text{card}(S)$ si y sólo si $p_k = \frac{1}{\text{card}(S)}$, para cada $k = 1, .., n$.

DEMOSTRACIÓN: a) Como $p_k \log_2(\frac{1}{p_k}) \geq 0$, para cada k, $H(S) = 0$ implica que cada sumando debe ser nulo. Entonces, para cada $k = 1, 2, .., n$, se tiene $p_k = 0$ ó $p_k = 1$. Pero $\sum_k p_k = 1$, por tanto, sólo cabe la posibilidad de que haya un único elemento a_1 en S y $p_1 = 1$.

Finalmente, nótese que (b) es consecuencia directa del Lema de Gibbs.□

3.3. El Teorema de codificación en un canal sin ruido

En esta sección abordamos los principales resultados sobre codificación en un canal sin ruido. El resultado fundamental es el denominado Teorema de Codificación para un canal sin ruido que establece que la longitud media en un código instantáneo supera a la entropia de la fuente de información. Recuérdese que la longitud media de un código C se define como $L(C) = \sum_k p_k L_k$, donde $S = \{a_1, a_2, .., a_n\}$ es la fuente, $P = \{p_1, p_2, .., p_n\}$ son las probabilidades de tansmitir a_k y L_k es la longitud de la palabra-código que codifica a_k.

Teorema 3.3.1. *Sea $C = \{c_1, c_2, .., c_n\}$ un código instantáneo para la fuente $S = \{a_1, a_2, .., a_n\}$ con probabilidades $p(a_k) = p_k$. Entonces*

$$\frac{H(p_1, .., p_n)}{\log_2 q} \leq L(C). \tag{3.5}$$

La igualdad se da si y sólo si $L_k = \log_q(\frac{1}{p_k})$.

DEMOSTRACIÓN: Como C es un código instantáneo, debe verificarse la desigualdad de Kraft. Es decir, se tiene

$$\sum_k q^{-L_k} \leq 1.$$

Vamos a aplicar el Lema de Gibbs con $q_k = q^{-L_k}$, pues $\sum_k q_k \leq 1$, por la desigualdad anterior. En virtud del Lema, tenemos

$$\sum_{k=1}^{n} p_k \log_q(\frac{1}{p_k}) \leq \sum_{k=1}^{n} p_k \log_q(\frac{1}{q_k}). \tag{3.6}$$

De (3.6) se sigue

$$\sum_{k=1}^{n} p_k \log_q(\frac{1}{p_k}) \leq \sum_{k=1}^{n} p_k \log_q q^{L_k} = \sum_{k=1}^{n} p_k L_k = L(C).$$

Ahora teniendo en cuenta que $\log_q x = \frac{\log_2 x}{\log_2 q}$, se obtiene el resultado deseado.

Finalmente, recuérdese que el propio Lema de Gibbs nos dice que se da la igualdad en (3.6) sólo cuando $q_k = p_k$, para todo k. Es decir, si y sólo si $p_k = \frac{1}{q^{L_k}}$, lo que equivale a que $L_k = \log_q(\frac{1}{p_k})$. □

El teorema anterior establece que se da la igualdad en (3.5) sólo en caso de que

$$L_k = \log_q\left(\frac{1}{p_k}\right) = -\log_q p_k.$$

Esto obliga a que $\log_q p_k$ sea entero. Por tanto, lo usual será que no se dé la igualdad. No obstante, vamos a ver que siempre existe un código instantáneo cuya longitud media está muy próxima al cociente $\frac{H(S)}{\log_2 q}$.

Teorema 3.3.2. *Existe un código instantáneo C cuya longitud media verifica*

$$\frac{H(S)}{\log_2 q} \leq L(C) \leq \frac{H(S)}{\log_2 q} + 1. \tag{3.7}$$

DEMOSTRACIÓN: Para cada k, escogemos L_k de modo que

$$q^{L_k} > \frac{1}{p_k} \geq q^{L_k - 1} \tag{3.8}$$

Con esta elección de L_k se tiene

$$\sum_{k=1}^{n} q^{-L_k} < \sum_{k=1}^{n} p_k = 1 \leq \sum_{k=1}^{n} q^{-(L_k - 1)}.$$

La primera de las desigualdades anteriores nos dice que se verifica la desigualdad de Kraft, que nos garantiza que existe un código instantáneo con palabras-código con tales longitudes. Para terminar la prueba, vamos a comprobar que la longitud media de este código verifica las desigualdades (3.7). Si tomamos logaritmo en base 2 en cada uno de los miembros de las desigualdades (3.8), multiplicamos por p_k y sumamos en $k = 1, 2, .., n$, resulta

$$\sum_{k=1}^{n} L_k p_k \log_2 q > H(S) \geq \sum_{k=1}^{n} (L_k - 1) p_k \log_2 q.$$

Dividiendo por $\log_2 q$, obtenemos

$$L(C) = \sum_{k=1}^{n} L_k p_k > \frac{H(S)}{\log_2 q} \geq \sum_{k=1}^{n} (L_k - 1) p_k = L(C) - 1.$$

□

Los resultados anteriores permiten obtener el teorema fundamental siguiente.

Teorema 3.3.3. *(Codificación de un canal sin ruido) Para cualesquiera probabilidades $p(a_k) = p_k$, se verifica*

$$\frac{H(p_1, .., p_n)}{\log_2 q} \leq L(C) \leq \frac{H(p_1, .., p_n)}{\log_2 q} + 1,$$

donde C es un código óptimo que codifica la fuente $S = \{a_1, .., a_n\}$.

Nota 3.3.4. En realidad la cota superior $H(p_1, .., p_n) + 1$ puede ser manifiestamente mejorada. En 1978 Gallagher probó lo siguiente. Denotemos por $p_{\max}$ la mayor de las probabilidades de los símbolos fuente. Si $p_{\max} \geq 0.5$, entonces $H(p_1, .., p_n) + p_{\max}$ es una cota superior de la longitud media del código de Huffman. Si $p_{\max} \leq 0.5$, entonces una cota superior es $H(p_1, .., p_n) + p_{\max} + 0.086$.

En el caso de alfabeto binario las relaciones anteriores toman la forma

$$H(p_1, .., p_n) \leq L(C) \leq H(p_1, .., p_n) + 1.$$

Ejemplo 3.3.5. (a) Se desea codificar la fuente $\{a_1, a_2, a_3\}$. Probar que existe un código instantáneo binario con longitudes $L_1 = L_2 = L_3 = 2$.

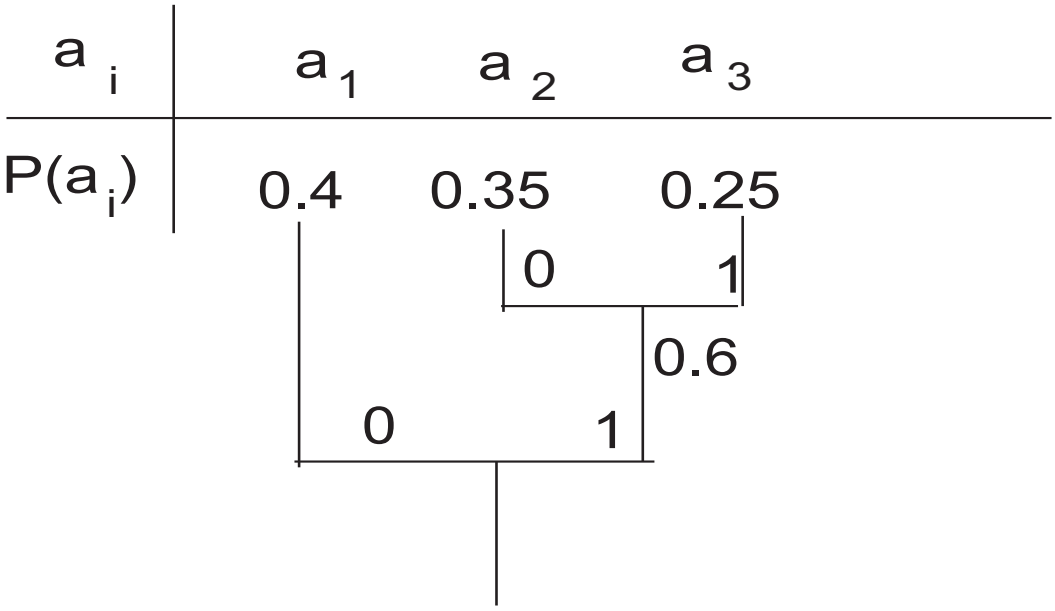
(b) Si las probabilidades son $P_1 = 0.4$, $p_2 = 0.35$ y $P_3 = 0.25$, aplicar el algoritmo de Huffman para determinar un código óptimo.

(a) Vamos a comprobar que se verifica la desigualdad de Kraft:

$$2^{-2} + 2^{-2} + 2^{-2} = 3/4 \leq 1.$$

Por tanto, existe un código instantáneo binario cuyas palabras-código tienen dichas longitudes. De hecho, es fácil construir un ejemplo: $a_1 = 00, a_2 = 10$, y $a_3 = 11$. Nótese que la longitud media es $L(C) = 2$. La entropía vale $H = 1.559$ bits.

(b) Vamos a aplicar el algoritmo de Huffman para determinar un código C instantáneo óptimo. Veremos que la longitud media, $L(C) = 1.6$, está más ajustada a la entropía.



Vemos que el código resultante es $C = \{0, 10, 11\}$, cuya longitud media es $L(C) = 1.6$.

3.4. Códigos de Huffman extendidos

Hay situaciones en las que el código óptimo que obtenemos por el método de Huffman tiene una longitud media que se aleja bastante, en términos relativos, de la entropía de la fuente, especialmente cuando el número de elementos del alfabeto fuente es pequeño y la probabilidad máxima es grande en comparación con las restantes. El ejemplo siguiente ilustra bien este problema

Ejemplo 3.4.1. Sean $S = \{m_1, m_2, m_3\}$ y $P = \{0.8, 0.02, 0.18\}$. Se comprueba fácilmente que el código óptimo que se obtiene por el método de Huffman es el siguiente

símbolo	palabra-código
m_1	0
m_2	11
m_3	10

La longitud media del código es 1.2 bits y la entropía de la fuente 0.816. La diferencia entre ambos valores representa el 47 % de la entropía. En estos casos es preferible buscar una palabra-código para (cada pareja o terna) de símbolos fuente, en lugar de codificar cada m_i . Es decir, en lugar de considerar la fuente anterior, se considera la dada por

$$S^2 = \{m_1m_2, m_1m_3, m_2m_3, m_2m_1, m_3m_1, m_3m_2, m_1m_1, m_2m_2, m_3m_3\}$$

con las probabilidades

$$P^2 = \{0.016, 0.144, 0.0036, 0.016, 0.1440, 0.0036, 0.64, 0.0004, 0.0324\}.$$

Aplicando el algoritmo de Huffman a esta nueva situación se encuentra el código óptimo siguiente:

mensaje	probabilidades	palabra-código
m_1m_1	0.64	0
m_1m_2	0.016	10101
m_1m_3	0.144	11
m_2m_1	0.016	101000
m_2m_2	0.0004	10100101
m_2m_3	0.0036	1010011
m_3m_1	0.1440	100
m_3m_2	0.0036	10100100
m_3m_3	0.0324	1011

El nuevo código recibe el nombre de código de Huffman extendido. Su longitud media es $L = 1.7228$, mientras que la entropía es igual a 1.632. No obstante, lo realmente importante es notar que 1.7228 es la longitud media de las palabras-código pero ahora cada una de estas palabras codifica

dos símbolos $m_i m_j$. Por tanto, en términos de la fuente original, la longitud media es $1.7228/2 = 0.8614$. Es decir, ahora la diferencia con la entropía de la fuente inicial es tan sólo de 0.045.

Pasamos ahora a desarrollar el método de una forma general. Supongamos que se desea codificar las cadenas de la forma $m_1 m_{i_2} \cdots m_{i_N}$, en lugar de los símbolos m_i. En este caso debemos considerar la fuente

$$S^N = \{m_1 m_{i_2} \cdots m_{i_N} : 1 \leq i_1, i_2, ..., i_N \leq n\}$$

con la función de probabilidad

$$P^N = \{p_1 p_{i_2} \cdots p_{i_N} : 1 \leq i_1, i_2, ..., i_N \leq n\},$$

donde $p_1 p_{i_2} \cdots p_{i_N}$ es la probabilidad de que sea emitida la cadena $m_1 m_{i_2} \cdots m_{i_N}$ (suponemos que el canal no tiene memoria y, por tanto, la probabilidad de que sea emitido m_j es independiente de que se haya emitido previamente m_i).

La entropía de la fuente S^N viene dada por:

$$H(S^N) = -\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_N=1}^{n} P(m_{i_1} m_{i_2} \cdots m_{i_N}) \cdot \log P(m_{i_1} m_{i_2} \cdots m_{i_N}) =$$

$$= -\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_N=1}^{n} P(m_{i_1})P(m_{i_2})\cdots P(m_{i_N})\cdot\log P(m_{i_1})P(m_{i_2})\cdots P(m_{i_N}) =$$

$$= -\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_N=1}^{n} P(m_{i_1})P(m_{i_2})\cdots P(m_{i_N}) \cdot \sum_{j=1}^{N}\log P(m_{i_j}) =$$

$$= -\sum_{i_1=1}^{n} P(m_{i_1}) log P(m_{i_1}) \big[\sum_{i_2=1}^{n}\cdots\sum_{i_N=1}^{n} P(m_{i_2})\cdots P(m_{i_N})\big] -$$

$$-\sum_{i_2=1}^{n} P(m_{i_2}) log P(m_{i_2}) \big[\sum_{i_1=1}^{n}\sum_{i_3=1}^{n}\cdots\sum_{i_N=1}^{n} P(m_{i_2})\cdots P(m_{i_N})\big] -$$

$$\cdots\cdots$$

$$-\sum_{i_N=1}^{n} P(m_{i_N}) log P(m_{i_N}) \big[\sum_{i_1=1}^{n} \cdots \sum_{i_{N-1}=1}^{n} P(m_{i_1}) \cdots P(m_{i_{N-1}}) \big].$$

Ahora basta notar que las sumas que aparecen entre corchetes son todas iguales a 1 (son las sumas de las probabilidades de todos los mensajes fuente de longitud $N-1$). Por tanto, hemos obtenido la igualdad

$$H(S^N) = -\sum_{i_1=1}^{n} P(m_{i_1}) log P(m_{i_1}) - \cdots - \sum_{i_N=1}^{n} P(m_{i_N}) log P(m_{i_N}) = N \cdot H(S).$$

Si C_N es un código óptimo (binario) para la fuente S^N, por el teorema de codificación en un canal sin ruido, se verifica

$$H(S^N) \leq L(C_N) \leq H(S^N) + 1,$$

donde $L(C_N)$ denota la longitud media del código C_N. Entonces el número medio de bits que se requiere para codificar con el código C_N, por cada símbolo de la fuente original, es $R = L(C_N)/N$. Si dividimos por N en la última desigualdad, obtenemos

$$H(S^N)/N \leq R \leq H(S^N)/N + 1/N.$$

Finalmente, teniendo en cuenta la relación $H(S^N) = N \cdot H(S)$, resulta

$$H(S) \leq R \leq H(S) + 1/N.$$

Si C es un código óptimo para la fuente S, sabemos que

$$H(S) \leq L(C) \leq H(S) + 1.$$

comparando las dos últimas relaciones deducimos que puede conseguirse una codificación que garantice que el número medio de bits necesarios (por cada símbolo de la fuente original) esté todo lo próximo que queramos a la entropía. Bastará escoger N lo suficientemente grande y codificar cadenas de N símbolos, en lugar de los m_i.

3.5. Ejercicios

1. Se considera el alfabeto fuente $S = \{a, b, c\}$ y la función de probabilidad $P = \{0.7, 0.15, 0.15\}$. Determinar el código binario de Huffman para codificar las parejas de letras. Calcular la entropía de la fuente y el código de Huffman para codificar los símbolos de la fuente y comparar los resultados obtenidos.

3.6. Prácticas de Programación

1. Dados el alfabeto fuente S y la función de probabilidad P, elaborar un programa de Matlab para determinar un código extendido (binario) de Huffman y codificar y decodificar un mensaje fuente de longitud arbitraria. Usando las funciones de Matlab, el programa deberá determinar primero la entropía de la fuente y la longitud media del código de Huffman y, si la diferencia entre ambos valores sobrepasa cierto porcentaje de la entropía, procederá a elaborar un código de Huffman para los pares de símbolos.