

8. Podstawy teorii informacji

8.1. Ilość informacji – dla sygnału o znanym prawdopodobieństwie pojawiania się

$$I = \log_2\left(\frac{1}{p}\right) = -\log_2(p)$$

8.2. Entropia źródła informacji - wartość średnia ilości informacji generowanej przez źródło

$$H = \sum_i p_i \cdot I_i$$

$$H = \sum_i p_i \cdot \log_2\left(\frac{1}{p_i}\right) = -\sum_i p_i \cdot \log_2(p_i)$$

8.3. Średnia (oczekiwana) długość ciągu kodowego

Średnia ilość informacji przypadająca w danym kodzie na jedną transmitowaną wartość.

$$\bar{R} = \sum_i p_i \cdot l_i$$

Jeżeli stosujemy transmisję ciągów symboli to czasami mówimy też o średniej długości kodu na symbol.

$$R_n = \frac{\bar{R}}{l}$$

gdzie: l jest długością ciągu w symbolach.

8.4. Sprawność kodu

Stosunek entropii do średniej długości ciągu kodowego. Miara idealności kodu.

$$\eta = \frac{H}{\bar{R}}$$

8.5. Pojemność informacyjna kanału – granica Shannona

Maksymalna – nieprzekraczalna ilość informacji, którą można przesłać kanałem o określonych parametrach technicznych, niezależnie od rodzaju stosowanej modulacji.

$$C = B \cdot \log_2(1 + SNR) \text{ [bit/s]}$$

gdzie:

B – szerokość pasma [Hz]

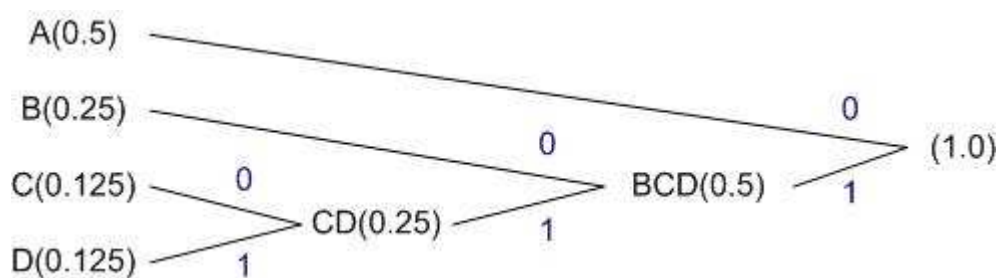
SNR – stosunek mocy sygnału do szumu w kanale [W/W]

8.6. Optymalny kod Huffmana

Kod Huffmana jest kodem optymalnym pod względem sprawności. Huffman udowodnił, że nie można stworzyć kodu bezprzecinkowego o mniejszej średniej długości.

Algorytm:

1. Posortować symbole od największego do najmniejszego prawdopodobieństwa
2. Połączyć dwa najmniej prawdopodobne symbole w jeden o łącznym prawdopodobieństwie
3. Powórzyć od pkt. 1 aż do otrzymania jednego symbolu o prawdopodobieństwie równym 1.0
4. W otrzymanym drzewie (de)kodującym przypisujemy 0 i 1 na każdym rozwidleniu – konsekwentnie 1-mniejsze prawdopodobieństwo 0-większe.



Rys. Drzewo (de)kodujące optymalnego kodu Huffmana z przykładu 81.

Przykład 81:

I	p_i	I_i	Kod Huffmana	l_i
A	0.50	1	0	1
B	0.25	2	10	2
C	0.125	3	110	3
D	0.125	3	111	3

$$R = 0.5 \cdot 1 + 0.25 \cdot 2 + 0.125 \cdot 3 + 0.125 \cdot 3 = 1.75 \text{ [bit/symbol]}$$

$$H = 0.5 \cdot \log_2(2) + 0.25 \cdot \log_2(4) + 0.125 \cdot \log_2(8) + 0.125 \cdot \log_2(8) = 1.75 \text{ [bit/symbol]}$$

Przykład 82:

Określ minimalną jakość kanału telefonicznego (300-3400Hz) umożliwiającą transmisję z szybkością 33600 bps.

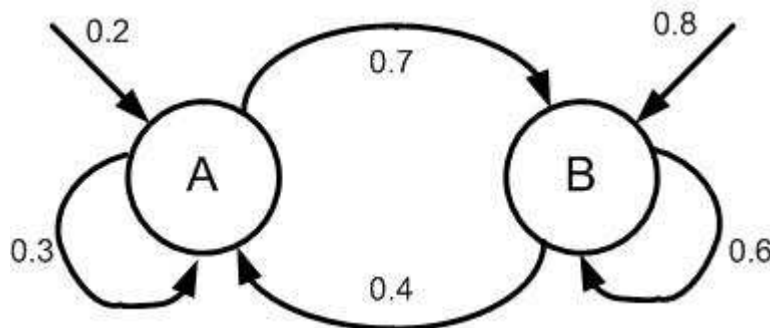
$$33600 = 3100 \cdot \log_2(1 + \text{SNR})$$

stąd:

$$\text{SNR} = 2^{33600 / 3100} - 1 = 1830 = 10 \cdot \log_{10}(1830) \text{ [dB]} = 32.6 \text{ [dB]}$$

Przykład 83:

Porównaj średnie długości kodów Huffmana dla ciągu 1, 2 i 3-ech symboli.



Pojedyncze symbole

I	p_i	I_i	Kod Huffmana	l_i
A	0.2	2.32	1	1
B	0.8	0.32	0	1

$$R = 0.2 \cdot 1 + 0.8 \cdot 1 = 1$$

$$R_n = 1/1 = 1$$

$$H = 0.2 \cdot 2.32 + 0.8 \cdot 0.32 = 0.72$$

$$\eta = 0.72 / 1.0 = 0.72$$

Ciąg dwóch symboli

I	P_i	I_i	Kod Huffmana	l_i
AA	$0.2 \cdot 0.3 = 0.06$	4.06	011	3
AB	$0.2 \cdot 0.7 = 0.14$	2.84	010	3
BA	$0.8 \cdot 0.4 = 0.32$	1.64	00	2
BB	$0.8 \cdot 0.6 = 0.48$	1.06	1	1

$$R = 0.06 \cdot 3 + 0.14 \cdot 3 + 0.32 \cdot 2 + 0.48 \cdot 1 = 1.72$$

$$R_n = 1.72 / 2 = 0.86$$

$$H = 0.06 \cdot 4.06 + 0.14 \cdot 2.84 + 0.32 \cdot 1.64 + 0.48 \cdot 1.06 = 1.67$$

$$\eta = 1.67 / 1.72 = 0.97$$

Ciąg trzech symboli

I	P_i	I_i	Kod Huffmana	l_i
AAA	$0.2 \cdot 0.3 \cdot 0.3 = 0.018$	5.80	10001	5
AAB	$0.2 \cdot 0.3 \cdot 0.7 = 0.042$	4.57	10000	5
ABA	$0.2 \cdot 0.7 \cdot 0.3 = 0.042$	4.57	1001	4
ABB	$0.2 \cdot 0.7 \cdot 0.7 = 0.098$	3.35	101	3
BAA	$0.8 \cdot 0.4 \cdot 0.4 = 0.128$	2.97	001	3
BAB	$0.8 \cdot 0.4 \cdot 0.6 = 0.192$	2.38	000	3
BBA	$0.8 \cdot 0.6 \cdot 0.4 = 0.192$	2.38	11	2
BBB	$0.8 \cdot 0.6 \cdot 0.6 = 0.288$	1.80	01	2

$$R = 2.68$$

$$R_n = 2.68 / 3 = 0.67$$

$$H = 2.63$$

$$\eta = 2.63 / 2.68 = 0.98$$

UWAGA: Warto tu zauważyć, że średnia długość ciągu na symbol R_n w tym przykładzie jest mniejsza od entropii dla pojedynczego symbolu. Kodując optymalnie ciągi symboli możemy uzyskać kompresję silniejszą od entropii.

Przykład:

Typowy tekst w języku angielskim $H = 4.1$

a_i	p_i	$\log_2 \frac{1}{p_i}$	l_i	$c(a_i)$
a	0.0575	4.1	4	0000
b	0.0128	6.3	6	001000
c	0.0263	5.2	5	00101
d	0.0285	5.1	5	10000
e	0.0913	3.5	4	1100
f	0.0173	5.9	6	111000
g	0.0133	6.2	6	001001
h	0.0313	5.0	5	10001
i	0.0599	4.1	4	1001
j	0.0006	10.7	10	1101000000
k	0.0084	6.9	7	1010000
l	0.0335	4.9	5	11101
m	0.0235	5.4	6	110101
n	0.0596	4.1	4	0001
o	0.0689	3.9	4	1011
p	0.0192	5.7	6	111001
q	0.0008	10.3	9	110100001
r	0.0508	4.3	5	11011
s	0.0567	4.1	4	0011
t	0.0706	3.8	4	1111
u	0.0334	4.9	5	10101
v	0.0069	7.2	8	11010001
w	0.0119	6.4	7	1101001
x	0.0073	7.1	7	1010001
y	0.0164	5.9	6	101001
z	0.0007	10.4	10	1101000001
-	0.1928	2.4	2	01

