ZAJĘCIA NR 3

Dzisiaj pomówimy o entropii, redundancji, średniej długości słowa kodowego i o algorytmie Huffmana znajdowania kodu optymalnego (pod pewnymi względami; aby dowiedzieć się jakimi – doczekaj do końca).

Def entropii
Jeżeli źródło może nadawać n różnych komunikatów, z prawdopodobieństwami odpowiednio p_1, p_2, …, p_n, to średnia ilość informacji w komunikatach z tego źródła wynosi $H = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}$ i nosi nazwę entropii informacyjnej źródła informacji.

Występujący tu symbol $\sum_{i=1}^{n}$ oznacza, że to co po im występuje należy sumować biorąc za i kolejne liczby naturalne od 1 do n, np.: $\sum_{i=1}^{4} i^2 = 1^2 + 2^2 + 3^2 + 4^2 = 1 + 4 + 9 + 16 = 30$.

Przykład:
1) Źródło nadaje 4 komunikaty, każdy z prawdopodobieństwem = ¼ .
Wtedy $H = \frac{1}{4} \cdot \log_2 4 + ... + \frac{1}{4} \cdot \log_2 4 = 4 \cdot \frac{1}{4} \log_2 4 = \log_2 4 = 2$.
2)

i	1	2	3	4	SUMA
p_i	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1

$$H = \frac{1}{2} \cdot \log_2 2 + 3 \cdot \frac{1}{6} \cdot \log_2 6 = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \log_2 (2 \cdot 3) = \frac{1}{2} + \frac{1}{2} \cdot (\log_2 2 + \log_2 3) = \frac{1}{2} + \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \log_2 3 \approx$$
$$\approx 1 + \frac{1}{2} \cdot 1{,}585 = 1 + 0{,}7925 = 1{,}7925,$$

gdzie wartość $\log_2 3$ otrzymaliśmy z tablic matematycznych, posiłkując się wzorem na zamianę podstaw logarytmów: $\log_b c = \frac{\log_a c}{\log_a b}$ podstawiając a=10, b=2, c=3.

Dodatkowo zauważamy, że oczywiście suma wszystkich p_i wynosi 1 (tak też było w poprzednim przykładzie; tak jest zawsze!).
3) $i = 1$, $p_i = 1$ (tj. choćby były inne potencjalne możliwości, to i tak ich „potencjał” wynosi 0, czyli nie bierzemy ich pod uwagę).
Wtedy: $H = 1 \cdot \log_2 1 = 1 \cdot 0 = 0$,
czyli nie przekazujemy żądnej informacji. Przekazujemy komunikat, ale nie przekazujemy informacji (patrz ZAJĘCIA I).
Własności entropii:
1) $H \geq 0$; $H = 0 \Leftrightarrow n = 1 \wedge p = 1$ (jak w przykładzie 3)
2) $H \leq \log_2 n$; $H = \log_2 n \Leftrightarrow \forall_i\, p_i = \frac{1}{n}$ (jak w przykładzie 1)

Teraz dwie definicje związane z samym kodem (jego def + długość)

Def
Kod danego komunikatu nazywa się ciągiem albo słowem kodowym tego komunikatu. Liczba elementów występujących w słowie jest jego długością.

Def
Wielkość $L = \sum_{i=1}^{n} p_i \cdot N_i$ nazywamy średnią długością słowa kodowego. Tu N_i - długość słowa kodowego, którego prawdopodobieństwo wynosi p_i.

Przykład (wszystkie p_i jak w przykładzie 2 powyżej)

i	1	2	3	4	SUMA
p_i	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1
N_i	5	4	2	6	

$$L = \sum_{i=1}^{n} p_i \cdot N_i = \frac{1}{2} \cdot 5 + \frac{1}{6} \cdot (4+2+6) = \frac{5}{2} + \frac{12}{6} = 2{,}5 + 2 = 4{,}5$$

Możemy wreszcie przejść do redundancji

Def
Różnicę $L - H \overset{ozn}{=} R$ i nazywamy redundancją danego sposobu kodowania.

Zachodzi przy tym: $R \geq 0$, bo $L \geq H$ (zawsze!).

Przykład
W rozwijanym tu przykładzie R = 4,5 – 1,7925 = 2,7075
Gdy jednak weźmiemy w nim inny zestaw wag N_i – tak aby komunikaty częste były krótkie, a rzadkie dłuższe (jeśli takie muszą być), a mianowicie przestawimy wartości N_1 z N_3 – to wówczas otrzymamy:

i	1	2	3	4	SUMA
p_i	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1
N_i	2	4	5	6	

$$L = \sum_{i=1}^{n} p_i \cdot N_i = \frac{1}{2} \cdot 2 + \frac{1}{6} \cdot (4+5+6) = 1 + \frac{15}{6} = 1 + 2{,}5 = 3{,}5,$$

co przy identycznym H (= 1,7925), daje nam R = 3,5 – 1,7925 = 1,7075 , a więc mniejszą redundancję.

Redundancja to więc nadmiarowość długości komunikatu ponad niezbędną konieczność.

Ze względu na ekonomiczność języka, jesteśmy zainteresowani jak najkrótszym kodowaniem. W tym celu przypatrzmy się poniższym rozważaniom.

Def
Kodem zwartym nazywamy kod jednoznaczny o minimalnej redundancji.
Przy tym: kod jednoznaczny – to taki kod, w którym żadem komunikat nie jest początkiem innego komunikatu.

Dobra sprawa! Kod zwarty – ma minimalną redundancję a przy tym można podać wiele komunikatów jeden-po-drugim, wiedząc przy tym jak wygląda ich tzw delimitacja (podział).

Dodatkowo zachodzi:
Jeśli prawdopodobieństwa występowania komunikatów są potęgami $\frac{1}{2}$, to można konstruować kod zwarty o redundancji = 0.

Jedna z metod konstruowania kodu zwartego jest metoda Huffmana

Przykład (znowu kontynuacja przykładu 2 podanego przy entropii):

i	1	2	3	4	SUMA
p_i	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1
kod k_i	k_1	k_2	k_3	k_4	

Będziemy tworzyć poszczególne kody k_i.
Metoda: łączymy poszczególne k_i parami – te, które mają p_i o najmniejszych wartościach.
W ten sposób powstają konglomeraty. Dalej możemy według tej samej zasady łączyć zarówno k_1 jak i ich konglomeraty. Czynimy to tak długo, aż nie otrzymamy 2 elementów:
1) START

kod k_i	k_1	k_2	k_3	k_4
p_i	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

2)

kody	k_1	k_2	$k_3 \vee k_4$
p_i	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}+\frac{1}{6}=\frac{1}{3}$

3)

kody	k_1	$k_2 \vee (k_3 \vee k_4)$
p_i	$\frac{1}{2}$	$\frac{1}{6}+\frac{1}{3}=\frac{1}{2}$

W tym momencie przechodzimy do „rozpakowywania” konglomeratów, za każdym razem jednemu z jego członów nadając wartość 0, a drugiemu 1:

$k_1 \quad 0$

$$\begin{tikzcd}
 & k_2 \vee (k_3 \vee k_4) \quad 1 \arrow[dl] \arrow[dr] & & \\
k_2 \quad 10 & & (k_3 \vee k_4) \quad 11 \arrow[dl] \arrow[dr] & \\
 & k_3 \quad 110 & & k_4 \quad 111
\end{tikzcd}$$

Sprawdźmy ile teraz wynosi R:

i	1	2	3	4
p_i	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
N_i	1	2	3	3

$$L = \sum_{i=1}^{n} p_i \cdot N_i = \frac{1}{2} \cdot 2 + \frac{1}{6} \cdot (2+3+3) = 1 + \frac{8}{6} = 1 + 1{,}3333 = 2{,}3333,$$

co przy identycznym H (= 1,7925), daje nam R = 2,3333 – 1,7925 = 0,5408 , a więc zdecydowanie mniejszą redundancję niż w poprzednich dwóch sytuacjach dla tego przykłady rozkładu prawdopodobieństw (R było równe 2,7075 i 1,7075).

Nie dość, że zmniejszyliśmy redundancję, to jeszcze uzyskaliśmy kod jednoznaczny.
Stąd: jeśli zapiszemy ciąg tych komunikatów (dowolnej długości), np.:
0110111010 ,
To zawsze możemy go rozłożyć na pojedyncze komunikaty:

1) Na 0 zaczyna się tylko 0. Odcimy to 0 i zostaje nam 110111010
2) 1- nie ma, 11 – też; jest dopiero 110 (a nic innego nie zaczyna się na 110). Odcinamy więc 110 i zostaje nam 111010
3) 1- nie ma, 11 – też; jest dopiero 111 (a nic innego nie zaczyna się na 111). Odcinamy więc 111 i zostaje nam 010.
4) 0 – jest, a na 0 nic więcej się nie zaczyna. Odcinamy więc 0 i zostaje nam 10
5) 1- nie ma, 10 – jest i kończy ciąg.

W ten sposób zdekodowaliśmy 0110111010 na 0 110 111 0 10. Przy tym dokonaliśmy tego w sposób jednoznaczny.

Czy redundancja jest szkodliwa?
Rozpatrzmy następujący przykład:

1) Należy zakodować cyfry od 0 do 9 (prawdopodobieństwo wystąpienia każdej z nich wynosi $\frac{1}{10}$. Ile wynosi R (bez i z wzgl. bitu parzystości – dodajemy jedynkę tam, gdzie w kodzie jest nieparzysta liczba jedynek, by w sumie była ich parzysta liczba))?

cyfra	Kod rozszerzony o bit parzystości				
	kod				bit parzystości
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	1	0	1	1	0
6	1	1	0	0	1
7	1	1	0	1	0
8	1	1	1	0	1
9	1	1	1	1	0

Bez bitu parzystości:

$$L = \sum_{i=1}^{10} \frac{1}{10} \cdot 4 = 10 \cdot \frac{4}{10} = 4,$$

$$H = \sum_{i=1}^{10} \frac{1}{10} \cdot \log_2 10 \approx 10 \cdot \frac{3{,}32}{10} = 3{,}32,$$

stąd R = L – H = 4 – 3,32 = 0,68.

Z bitem parzystości: L= 5, H – nie ulega zmianie (tj. też = 3,32), a więc R = L – H = 5 – 3,32 = 1,68.

Było więc minimalne (ułamkowe) i wzrosło zaledwie o 1.

Taka redundancja wydłuża więc słowo, jest więc szkodliwa. Jednak dodanie owego bitu parzystości umożliwia wykrycie potencjalnego przekłamania jednego znaku (bo wtedy nie będzie parzysta liczba jedynek), a gdy jesteśmy zainteresowani dużą niezawodnością komunikatu, wówczas owo zwiększenie redundancji o 1 nie jest ż tak kosztowne (lecz wręcz przeciwnie – jest korzystne). Mała redundancja nie jest niekorzystna.

CDN…