Complex theories do not work, simple algorithms do.[1]

ROZDZIAŁ II

KOMPRESJA DANYCH

W rozdziale tym omawiamy metody kodowania oparte na dążeniu, aby średnia długość słów kodowych była jak najmniejsza. Metody te mają zastosowanie w bezstratnej kompresji danych, np. w celu zmniejszenia objętości przechowywanych plików lub przesyłanych danych. Prowadzone rozważania są ograniczone do kodowania binarnego.

4. Kodowanie Shannona[2]

Nierówność Krafta

Nierówność Krafta[3] umożliwia ocenę, czy można zbudować kod przedrostkowy (a więc i natychmiastowy) dla zadanych długości poszczególnych słów kodowych.

Przyjmiemy, że alfabet wejściowy składa się z n symboli. Rozważymy kod binarny, dla którego określone są długości poszczególnych słów kodowych: $k_1, k_2, \ldots, k_n$. Jest oczywiste, że im słowa kodowe są krótsze (tzn. im wartości k_i mniejsze), tym kodowanie jest bardziej zwięzłe. Na pytanie, czy zestaw

---

[1] Na podstawie: V.N. Vapnik, *The Nature of Statistical Learning Theory*, Preface to the First Edition, Springer-Verlag, 2000.

[2] Podstawowa praca Claude'a Elwooda Shannona (1916-2001) to: *A mathematical theory of communication*, Bell System Technical Journal, Vol. 27 (1948), No 3, 379-423, No 4, 623-656. Przedstawiono tam m.in.: modele źródła informacji, pojęcie entropii oraz tzw. twierdzenie o kodowaniu (wraz ze związaną z nim metodą kodowania).

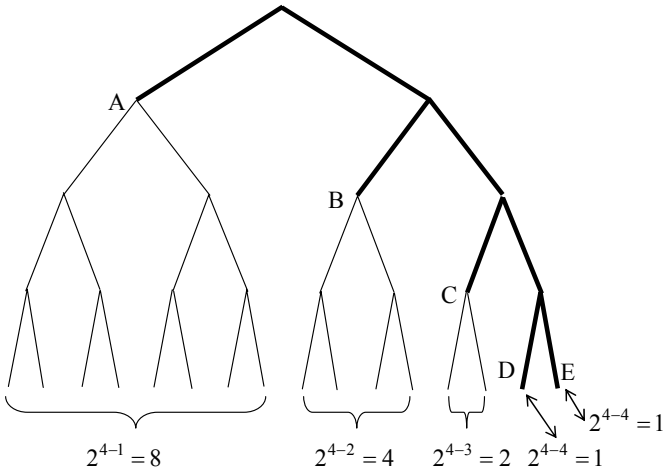
[3] Praca oryginalna: Leon G. Kraft (1949), *A device for quantizing, grouping, and coding amplitude modulated pulses*, Cambridge, MA: MS Thesis, Electrical Engineering Department, Massachusetts Institute of Technology.

długości $k_1, k_2, \ldots, k_n$ słów kodowych jest dopuszczalny, tzn. czy dążąc do maksymalnie zwięzłego kodu, nie planujemy zbyt krótkich słów kodowych, odpowiada następujące twierdzenie.

Kod przedrostkowy[4] spełnia następującą nierówność (Krafta):

$$\sum_{i=1}^{n} 2^{-k_i} \le 1 \tag{4.1}$$

Wykorzystywanie tej nierówności opiera się na twierdzeniu przeciwstawnym: jeśli nierówność Krafta nie jest spełniona, to nie istnieje kod przedrostkowy o rozpatrywanych długościach słów kodowych. Jeśli np. $[k_1, k_2, k_3, k_4] = [1,2,2,3]$, to $\sum_{i=1}^{4} 2^{-k_i} = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} = \frac{9}{8} > 1$. Zatem przy zadanych długościach słów kodowych nie można zbudować kodu przedrostkowego.



Rys. 4.1. Ilustracja do dowodu nierówności Krafta: $k = 4$, $k_1 = 1$, $k_2 = 2$, $k_3 = 3$, $k_4 = 4$, $k_5 = 4$, $\sum_{i=1}^{5} 2^{k-k_i} = 2^3 + 2^2 + 2^1 + 2^0 + 2^0 = 16$

Dowód nierówności oprzemy na wykorzystaniu konstrukcji kodu przedrostkowego na podstawie drzewa binarnego.

[4] Twierdzenie to można uogólnić, dowodząc tezę dla kodu jednoznacznie dekodowalnego. Wynik tego uogólnienia jest znany jako twierdzenie McMillana. Artykuł oryginalny: B. McMillan, *Two inequalities implied by unique decipherability*, Information Theory, IRE Transactions, Volume 2, Issue 4, December 1956, 115-116.

Oznaczymy: $k \geq \max\{k_1, k_2, \ldots, k_n\}$. Na podstawie rys. 4.1 mamy: $\sum_{i=1}^{n} 2^{k-k_i} \leq 2^k$. Zauważmy bowiem, że jeśli *i*-ty liść leży na poziomie k_i, to blokuje (nie pozwala na wykorzystanie) 2^{k-k_i} leżących niżej liści (na poziomie *k*). Aby uzyskać tezę, należy pomnożyć obie strony uzyskanej nierówności przez 2^{-k}.

Model źródła informacji

Praktyczny problem zwięzłego kodowania polega na tym, aby zadane teksty kodować za pomocą jak najkrótszych ciągów symboli wyjściowych. Rozwiązanie tego problemu wymaga sformułowania założeń dotyczących kodowanego tekstu. Założenia te zwykle przedstawia się, formułując *model źródła informacji*.

Dla dalszych rozważań będziemy wykorzystywać model odpowiadający sytuacji, kiedy odbiorca zna jedynie liczby wystąpień poszczególnych symboli w kodowanym ciągu symboli, a nie zna zasad (mechanizmu) emisji kolejnych symboli. Oznaczymy przez $N_1, N_2, \ldots, N_n$ liczby wystąpień poszczególnych symboli w kodowanym ciągu, przy czym całkowita długość ciągu jest równa $N = N_1 + N_2 + \cdots + N_n$. Iloraz

$$p_i = \frac{N_i}{N} \tag{4.2}$$

oznacza częstość występowania symbolu N_i. Brak znajomości zasad emisji kolejnych symboli odzwierciedla się w modelu, przyjmując, że emisja pojedynczego symbolu jest wynikiem losowego wyboru. Losowanie odbywa się z urny zawierającej odpowiednio $N_1, N_2, \ldots, N_n$ poszczególnych symboli. Symbole są losowane ze zwracaniem (wylosowany symbol powraca do urny przed następnym losowaniem). Przy tym założeniu częstość p_i jest równa prawdopodobieństwu wylosowania tego symbolu[5]. Zauważmy, że w przyjętym schemacie losowania (ze zwracaniem) prawdopodobieństwa p_i emisji symboli nie zmieniają się. Prawdopodobieństwa te nie zależą więc od poprzednio wygenerowanych symboli. W takim przypadku mówi się, że źródło jest *bez pamięci*.

Dalsze rozważania przeprowadzimy dla zadanego modelu źródła informacji (bez pamięci): alfabet źródła składa się z *n* symboli,

[5] Będziemy używać zamiennie określeń: *prawdopodobieństwo* (w celu podkreślenia losowości emisji symboli) i *częstość* (w celu uwypuklenia sposobu wyznaczenia wartości prawdopodobieństwa).

a prawdopodobieństwa wygenerowania (wyemitowania) symboli są odpowiednio równe: $p_1, p_2, \ldots, p_n$.

Oznaczymy dla zadanego kodu długości poszczególnych słów kodowych jako $k_1, k_2, \ldots, k_n$. Średnia długość słów kodowych jest równa:

$$L = \frac{N_1 k_1 + N_2 k_2 + \cdots + N_n k_n}{N} \tag{4.3}$$

gdzie, jak poprzednio, $N_1, N_2, \ldots, N_n$ oznaczają liczby wystąpień poszczególnych symboli w kodowanym ciągu (całkowita długość ciągu jest równa $N = N_1 + N_2 + \cdots + N_n$). Wykorzystując (4.2), uzyskujemy wzór:

$$L = \frac{N_1}{N} k_1 + \frac{N_2}{N} k_2 + \cdots + \frac{N_n}{N} k_n = \sum_{i=1}^{n} p_i k_i \tag{4.4}$$

Jak powiedzieliśmy wcześniej, im słowa kodowe są krótsze (tzn. im wartości k_i mniejsze), tym kodowanie jest bardziej zwięzłe. Słowa kodowe nie mogą być jednak zbyt krótkie ze względu na konieczność spełnienia nierówności Krafta.

Wśród kodów spełniających nierówność Krafta będziemy poszukiwać takiego, który dla zadanego źródła informacji zapewni najmniejszą średnią długość słów kodowych. Dość oczywiste jest spostrzeżenie, że symbole występujące częściej powinny być reprezentowane przez krótsze słowa kodowe. Spostrzeżenie to sformułujemy następująco:

Spostrzeżenie 1

Dla kodu o najmniejszej średniej długości słów kodowych spełniony jest warunek: jeśli $p_l > p_j$, *to* $k_l \le k_j$.

W celu wykazania prawdziwości tego spostrzeżenia wybierzemy dwa symbole o indeksach l, j takie, że $p_l > p_j$. Obliczymy średnią długość słów kodowych:

$$L = \sum_{\substack{i \neq l \\ i \neq j}} p_i k_i + p_j k_j + p_l k_l \tag{4.5}$$

Przyjmiemy: $k_l = u$, $k_j = v$, przy czym $u \le v$ (tzn. częściej występującemu symbolowi odpowiada krótsze słowo kodowe). Średnia długość słów kodowych wtedy wynosi:

$$L_1 = \sum_{\substack{i \neq l \\ i \neq j}} p_i k_i + p_j v + p_l u \tag{4.6}$$

Przyjmiemy teraz przeciwne wartości długości słów kodowych: $k_l = v$, $k_j = u$. Średnia długość słów kodowych wtedy wynosi:

$$L_2 = \sum_{\substack{i \neq l \\ i \neq j}} p_i k_i + p_j u + p_l v \tag{4.7}$$

Różnica między uzyskanymi wartościami jest równa:

$$L_1 - L_2 = u(p_l - p_j) - v(p_l - p_j) = (p_l - p_j)(u - v) \tag{4.8}$$

Wynika stąd, że $L_1 - L_2 \leq 0$. A więc pierwsze przyporządkowanie daje mniejszą wartość średniej długości ciągów kodowych.

Kod Eliasa

Metoda Eliasa opiera się na zasadzie, aby często występujący symbol wejściowy reprezentować krótkim słowem kodowym. Stąd pomysł, aby kodowane symbole uporządkować według częstości występowania i kolejno ponumerować. Większy numer (liczba naturalna dodatnia) oznacza mniejszą częstość występowania oznaczonego symbolu. Tak uporządkowany alfabet wejściowy może stanowić punkt wyjścia do zastosowania *algorytmu gamma* Eliasa. Algorytm ten służy do kodowania dodatnich liczb całkowitych w przypadku, gdy nie jest znana liczba największa. Ponieważ zgodnie z tym algorytmem mniejsze numery otrzymują krótsze słowa binarne, kod gamma Eliasa (ang. *Elias gamma code*) może być wykorzystywany do kodowania zwięzłego.

Tab. 4.1. Kod Eliasa

1	**1**	**7**	**00111**
2	**010**	**8**	**0001000**
3	**011**	**9**	**0001001**
4	**00100**	**10**	**0001010**
5	**00101**	**11**	**0001011**
6	**00110**	**12**	**0001100**

Algorytm kodowania:

1. Zapisz kodowaną liczbę binarnie. Liczbę bitów oznacz przez b.
2. Przed liczbą zapisaną binarnie dopisz $b-1$ zer.

Algorytm dekodowania:

1. Policz wszystkie zera aż do znalezienia pierwszej jedynki. Oznacz liczbę zer przez b.
2. Odczytaj liczbę zapisaną na $b+1$ pozostałych bitach słowa kodowego. Jest to wynik dekodowania.

Początek tabeli kodowej kodu uzyskanego zgodnie z algorytmem gamma został przedstawiony w tab. 4.1. Zauważmy, że największa kodowana liczba nie musi być znana, w każdej chwili można dodać nowy kodowany symbol[6].

Kod Shannona

Kodowanie Eliasa oparte było jedynie na uporządkowaniu (rankingu) symboli według częstości ich występowania. Sposób kodowania Shannona także opiera się na zasadzie, aby często występujący symbol wejściowy reprezentować krótszym słowem kodowym. W odróżnieniu od metody Eliasa zastosowanie kodu Shannona wymaga znajomości pełnej informacji o źródle informacji, tj. wartości prawdopodobieństw występowania poszczególnych symboli źródła.

Tab. 4.2. Przykład kodu Shannona

Symbol	**Indeks symbolu *i***	**Częstość symbolu** p_i	**Częstość skumulowana**	**Długość Shannona ciągu kodowego**	**Słowo kodowe**
A	1	0,5	$(0)_{10} = (0{,}000)_2$	1	0
B	2	0,25	$(0{,}5)_{10} = (0{,}100)_2$	2	10
C	3	0,125	$(0{,}75)_{10} = (0{,}110)_2$	3	110
D	4	0,125	$(0{,}875)_{10} = (0{,}111)_2$	3	111

Przed kodowaniem metodą Shannona należy wykonać następujące przygotowania:

1) obliczyć częstości p_i ($i = 1, 2, \ldots, n$) występowania symbolu (jako iloraz liczby wystąpień N_i symbolu o indeksie i w kodowanym słowie i długości tego słowa N),
2) symbole uporządkować[7] według częstości p_i.

[6] W tym sensie mówi się niekiedy, że jest to kod *uniwersalny* (podobnie jak kod unarny).

[7] W celu *uporządkowania* danych według ich liczbowej charakterystyki wykorzystywane są algorytmy *sortowania*.

Wynikiem tych czynności jest tablica symboli uporządkowanych według częstości ich występowania (od częstości największej do najmniejszej).

Algorytm kodowania Shannona:

1) Obliczenie długości Shannona słowa kodowego[8]: $k_i = \lceil -\log_2 p_i \rceil$ dla wszystkich *i*.
2) Utworzenie tablicy częstości skumulowanych.
3) Zapis binarny tych częstości.
4) Utworzenie słów kodowych jako pierwszych k_i bitów występujących po przecinku zapisu binarnego częstości skumulowanej.

Wykażemy, dla kodu Shannona spełniona jest nierówność Krafta. W tym celu, wykorzystując wprowadzone wcześniej oznaczenia (częstość występowania słowa kodowego o indeksie *i* jest równa p_i, a długość tego słowa k_i), zasadę Shannona doboru długości słowa kodowego zapiszemy następująco:

$$k_i \geq \log_2 \frac{1}{p_i} \tag{4.9}$$

Stąd mamy: $-k_i \leq \log_2 p_i$ lub inaczej: $2^{-k_i} \leq p_i$. Zatem $\sum_{i=1}^{n} 2^{-k_i} \leq 1$, co oznacza, że spełniona jest nierówność Krafta.

Wykorzystując zasadę (4.9), oszacujemy dolne ograniczenie średniej długości słów kodowych:

$$L = \sum_{i=1}^{n} p_i k_i \tag{4.10}$$

Zgodnie z (4.9) otrzymujemy:

$$L = \sum_{i=1}^{n} p_i k_i \geq \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} \tag{4.11}$$

Można na przykładach wskazać, że słowa kodu Shannona mogą mieć nadmiarowe ostatnie bity[9] (tzn., że bity te mogą być usunięte, a kod będzie poprawny). Przykład taki jest przedstawiony w tab. 4.3 (można skrócić słowo kodowe symbolu D, poprawiony w ten sposób kod ma mniejszą średnią długość

---

[8] Symbolem $\lceil x \rceil$ oznacza się najmniejszą liczbę całkowitą nie mniejszą od *x*. Stosuje się też oznaczenie: $ceil(x)$, gdzie określenie angielskie *ceil* można spolszczyć na *pułap*.

[9] Niezależnie od Shannona podobną metodę kodowania zaproponował R. Fano (1949): *The transmission of information*, Technical report No 65, Research Laboratory of Electronics, M.I.T. Metoda Fano daje słowa kodowe różniące się na ostatnim bicie od uzyskiwanych metodą Shanonna. Obecnie do obu kodów stosuje się wspólną nazwę: kod Shannona–Fano.

słów kodowych). Przedstawiony przykład jest na tyle pouczający, że na jego podstawie można sformułować następujące spostrzeżenie:

Spostrzeżenie 2

W kodzie binarnym o najmniejszej średniej długości słów kodowych dwa najdłuższe słowa kodowe powinny mieć jednakową długość, a różnić się tylko ostatnim bitem.

Tab. 4.3. Przykład kodu Shannona

Sym-bol	**Indeks symbolu *i***	**Częstość symbolu p_i**	**Częstość skumulowana**	**Długość Shannona ciągu kodowego**	**Słowo kodowe**
A	1	0,5	$(0)_{10} = (0{,}00000)_2$	$1 \geq -\log_2 0{,}5$	0
B	2	0,25	$(0.5)_{10} = (0{,}10000)_2$	$2 \geq -\log_2 0{,}25$	10
C	3	0,1251	$(0.75)_{10} = (0{,}11000)_2$	$3 \geq -\log_2 0{,}1251 = 2{,}9988$	110
D	4	0,1249	$(0.8751)_{10} = (0{,}11100)_2$	$4 \geq -\log_2 0{,}1249 = 3{,}0012$	1110

Entropia (nieokreśloność)

Wielkość określoną wzorem:

$$H = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} = -\sum_{i=1}^{n} p_i \log_2 p_i \quad \text{[bit]} \tag{4.12}$$

nazywa się *entropią*[10] (albo *nieokreślonością*) źródła informacji bez pamięci, generującego *n* symboli odpowiednio z prawdopodobieństwami p_i, $i = 1, 2, \ldots, n$.

Średnia długość słów kodowych kodu Shannona

Dla specjalnych wartości prawdopodobieństw:

$$p_i = 2^{-k_i} \tag{4.13}$$

[10] W podstawowej pracy *A mathematical theory of communication* Claude E. Shannon odwoływał się do definicji entropii Boltzmanna. Podobno użycie nazwy *entropia* zamiast *nieokreśloność* (ang. *uncertainty*) zalecał Shannonowi von Neumann, a decydujący argument był następujący: *Nobody knows what entropy really is, so in a debate you will always have the advantage*. Por. także określenie entropii von Neumanna.