Kodowanie, Kompresja, Kryptografia

konspekt II, 8.III.2022

Przesyłamy słowa $\alpha \in \mathcal{A}^*$, $\mathcal{A} = \{a_1, a_2, ..., a_k\}$, i zakładamy, że litery a_i pojawiają się w nich z prawdopodobieństwami $p_i = p(a_i)$.

Szukamy kodu $c : \mathcal{A} \to \mathcal{B}^*$, który minimalizuje *średnią* długość słowa kodowego

$$\hat{l} = \sum_{i=1}^{k} p(a_i)\, l(a_i) \;=\; \min \qquad \text{pod warunkiem, że} \qquad \sum_{i=1}^{k} D^{-l(a_i)} \;\leq 1\,.$$

Szukamy więc układu długości $l_i = l(a_i)$ minimalizujących $\hat{l}$ przy ograniczeniu

$$g(l_1, ..., l_k) \;=\; \sum_{i=1}^{k} D^{-l_i} \;-\; 1 \;=\; 0\,.$$

Oznaczamy

$$F(l_1, ..., l_k, \lambda) \;\;=\;\; \hat{l}(l_1, ..., l_k) + \lambda g(l_1, ..., l_k)$$

i szukamy ekstremum F, $\nabla F = 0$, czyli

$$\frac{\partial \hat{l}}{\partial l_i} = -\lambda \frac{\partial g}{\partial l_i} \qquad \Rightarrow \qquad p(a_i) = \lambda \,\ln D\, D^{-l_i}$$

$$\frac{\partial F}{\partial \lambda} = 0 \qquad \Rightarrow \qquad \sum_{i=1}^{k} D^{-l_i} = 1\,.$$

Sumując stronami po i otrzymujemy $\lambda \ln D = 1$, a stąd $p_i = D^{-l_i}$, czyli po zlogarytmowaniu

$$l_i \;=\; -\log_D p_i \;=\; -\frac{\log p_i}{\log D} \quad \text{a więc} \quad \sum_{i=1}^{k} p_i\, l_i = \frac{-\sum_i p_i \log p_i}{\log D} \;=\; \frac{H(p)}{\log D}$$

Ponieważ jednak $l_i = l(a_i)$ muszą mieć całkowite wartości, wybieramy

$$-\log_D p_i \;\leq\; l(a_i) \;=\; \lceil -\log_D p_i \rceil \;<\; 1 - \log_D p_i\,. \qquad (*)$$

Pierwszą nierówność można zapisać jako $D^{-l(a_i)} \leq p_i$, co po zsumowaniu względem i daje $\sum_i D^{-l(a_i)} \leq 1$.

Jeśli teraz pomnożymy $(*)$ stronami przez p_i i wysumujemy po i, otrzymamy (po zamianie logarytmu na dwójkowy)

$$\frac{H(A)}{\log_2 D} \;\leq\; \hat{l}(c) \;<\; 1 + \frac{H(A)}{\log_2 D}\,.$$

Dzięki warunkowi $\sum_i D^{-l(a_i)} \leq 1$ nierówność Krafta II gwarantuje, że c można wybrać jako kod bezprefiksowy. Udowodniliśmy tym samym ważne

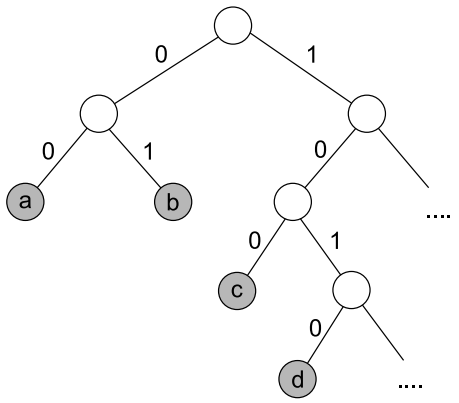
Twierdzenie Shannona-Fano o kodowaniu bez szumu.

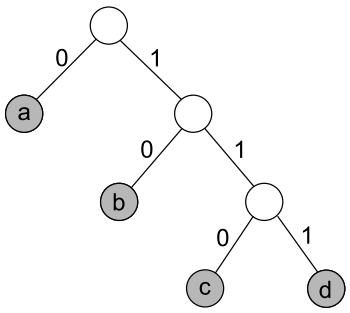
Niech A będzie zmienną losową o wartościach w zbiorze $\mathcal{A}$, (tj. dany jest rozkład $p(a) = P(A = a)$). Istnieje kod bezprefiksowy $c : \mathcal{A} \to \mathcal{B}^*$, $|\mathcal{B}| = D$, dla którego

$$\frac{H(A)}{\log_2 D} \;\leq\; \mathbb{E}[c(A)] \;<\; 1 + \frac{H(A)}{\log_2 D}\,.$$

Przykład $\mathcal{A} = \{a, b, c, d\}$ z prawdopodobieństwami p_i odp. 0.4, 0.3, 0.2, 0.1. $H(A) = 1.85$.

x	$p(x)$	$-\log p(x)$	$\lceil -\log p(x) \rceil$
a	0.4	1.32	2
b	0.3	1.74	2
c	0.2	2.32	3
d	0.1	3.32	4

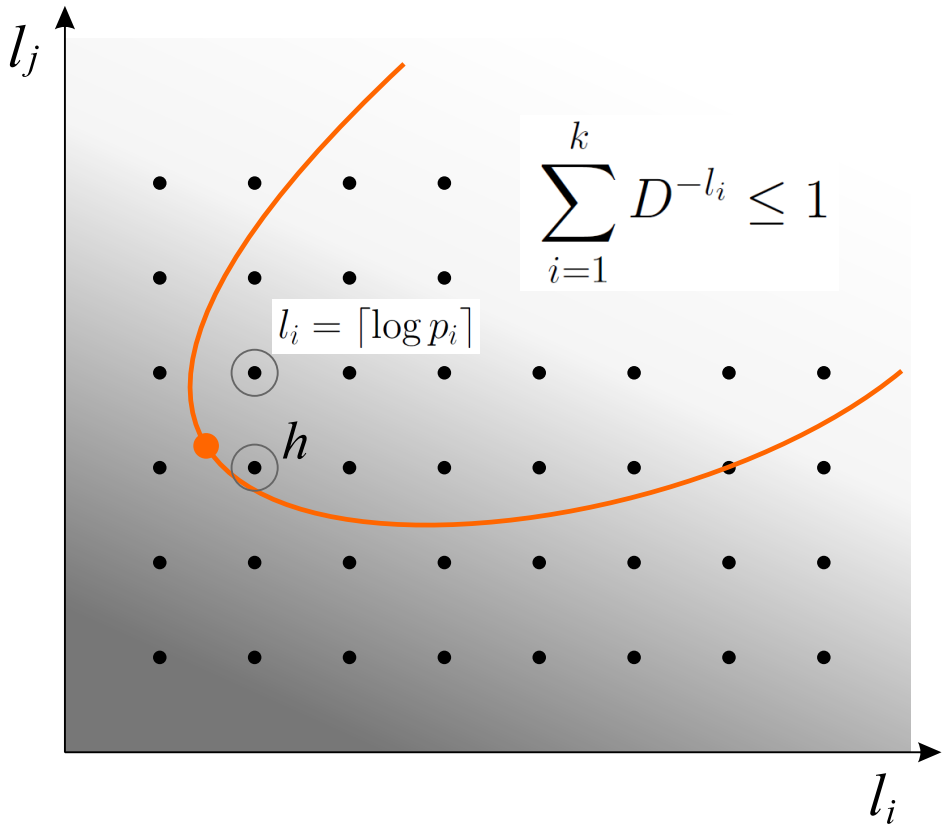




	a	b	c	d
$c(a)$	00	01	100	1010

	a	b	c	d
$h(a)$	0	10	110	111

Kod c Shannona-Fano ma średnią długość $\mathbb{E}[c(A)] = 2.4$, drugi kod Huffmana h jest optymalny, $\mathbb{E}[h(A)] = 1.9$



Punkt optymalny (czerwony) $l_i = \log p_i$, $i = 1, 2, \ldots k$ i jego całkowitoliczbowe przybliżenie $l_i = \lceil \log p_i \rceil$. Kolor szary wizualizuje rozkład wartości $\mathbb{E}[c(A)] = \hat{l}(c) = \sum p_i l_i$. W otoczeniu może się jednak nadal znaleźć lepsze rozwiązanie h (kod Huffmana) spełniające nierówność Krafta, mimo że $l_i < \lceil \log p_i \rceil$ dla niektórych długości l_i.

Kodowanie Huffmanna

Załóżmy, że $\mathcal{A} = \{a, b, c, d, e, f, g, h\}$. Zwykły kod typu ASCII byłby 3-bitowy, $c(a) = 000, \ldots c(h) = 111$. Należy zakodować tekst α 100-znakowy. Wtedy $|c^*(\alpha)| = 300$. Litery występują jednak w tym tekście niejednakowo często, np.

a	b	c	d	e	f	g	h
22	8	12	6	24	15	9	4

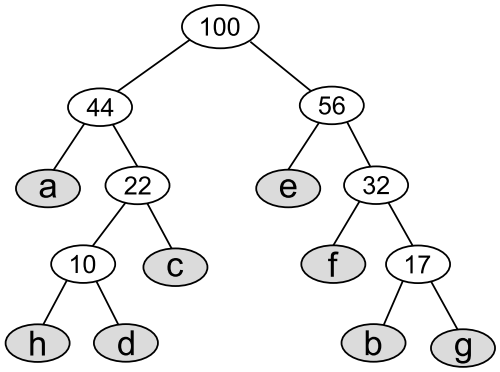
a więc $p(a) = 0.22$, $p(b) = 0.08$, itd. Ponieważ $H(A) = 2.79$, jeśli uda nam się skonstruować kod o *średniej* długości słowa bliskiej $H(A)$, cały tekst α powinien zająć niewiele więcej niż $100 \cdot H(A) = 279$ bitów. Konstrukcja kodu Huffmana:

— Budujemy drzewo binarne kodu od dołu, dołączając do niego kolejno litery od najrzadszych do najczęstszych.

— Zaczynamy łącząc w drzewo dwie najrzadsze litery, tu h i d, w korzeniu drzewa zapisujemy łączną liczbę wystąpień tych liter (sumę prawdopodobieństw), czyli 10.

— Wybieramy 2 kolejne najrzadsze elementy (litery lub częściowe drzewa ze względu na wagi zapisane w ich korzeniach) i łączymy w jedno drzewo, znowu sumując wagi w korzeniu

— Po wyczerpaniu elementów drzewo reprezentuje optymalny kod prefiksowy.



W naszym przykładzie

litera	a	b	c	d	e	f	g	h
kod	00	1110	011	0101	10	110	1111	0100
l. wystąpień	22	8	12	6	24	15	9	4
l. bitów	44	32	36	24	48	45	36	16

W sumie 281 bitów. Mamy więc $\mathbb{E}[h(A)] = 2.81 > 2.79 = H(A)$.

Konstrukcja kodu Huffmana
$\mathcal{A} = \{a_1, a_2, ..., a_k\}$, $h : \mathcal{A} \to \{0,1\}^*$, $p_i = p(a_i)$ spełniają $p_1 \geq p_2 \geq ... \geq p_k$.

1. Jeśli $k = 2$, weźmy $h(a_1) = 0$, $h(a_2) = 1$. Mamy $\mathbb{E}[h(A)] = 1$. Jest to oczywiście kod optymalny.

2. Przypuśćmy, że potrafimy zbudować kod Huffmana dla dowolnego alfabetu o $k-1$ literach. Wybierzmy w $\mathcal{A}$ dwie litery o najmniejszych prawdopodobieństwach, a_{k-1} i a_k. Tworzymy nowy alfabet $\hat{\mathcal{A}} = \{a_1, a_2, ..., a_{k-1} \cup a_k\}$, w którym obydwie litery a_{k-1} i a_k reprezentuje "kolektywny" symbol $a_{k-1} \cup a_k$ o prawdopodobieństwie $p_{k-1} + p_k$. Niech $\hat{h} : \hat{\mathcal{A}} \to \{0,1\}^*$ będzie kodem Huffmana dla nowego alfabetu. Kod Huffmana h dla $\mathcal{A}$ powstaje z $\hat{h}$ następująco:

$$h(a_i) \;=\; \begin{cases} \hat{h}(a_i) & \text{dla } \; i = 1, 2, ..., k-2 \\ \hat{h}(a_{k-1} \cup a_k)0 & \text{dla } \; i = k-1 \\ \hat{h}(a_{k-1} \cup a_k)1 & \text{dla } \; i = k \end{cases} \qquad (*)$$

   Jest to oczywiście kod bezpefiksowy.

Twierdzenie Kod Huffmana jest optymalny, tj. jego średnia długość jest *najmniejsza* wśród wszystkich dekodowalnych kodów $c : \mathcal{A} \to \{0,1\}^*$.

Lemat Niech $\mathcal{A} = \{a_1, a_2, ..., a_k\}$ oraz $p_1 \geq p_2 \geq ... \geq p_k > 0$, $p_i = p(a_i)$. Istnieje kod optymalny $c : \mathcal{A} \to \{0,1\}^*$, dla którego $\mathbb{E}[c(A)]$ jest minimalne oraz

a) długości $l_i = |c(a_i)|$ spełniają $l_1 \leq l_2 \leq ... \leq l_k$

b) słowa kodowe $c(a_{k-1})$ i $c(a_k)$ mają jednakową długość i różnią się tylko ostatnim bitem.

DOWÓD. Weźmy kod bezprefiksowy d o długościach słów $d_i = \lceil -\log p_i \rceil$. Niech $\mathbb{E}[d(A)] = D$. Wówczas istnieje jedynie skończenie wiele bezprefiksowych kodów binarnych c, dla których $\sum l_i p_i \leq D$ przy zachowaniu nierówności $\sum 2^{-l_i} \leq 1$. Jeden z nich osiąga minimalną średnią długość — to jest kod optymalny.

a) Jeśli $p_i \geq p_j$, lecz $l_i > l_j$, wówczas można zmniejszyć $\sum l_i p_i$ przez zamianę słów kodowych $c(a_i)$ i $c(a_j)$. Zatem dla optymalnego kodu mamy $l_1 \leq l_2 \leq ... \leq l_k$.

b) Niech $L = l_k$, $c(a_k) = \omega b$, gdzie $|\omega| = l_k - 1$, a b jest ostatnim bitem $c(a_k)$. Utwórzmy nowy kod zamieniając $c(a_k)$ na ω, co oczywiście zmniejsza $\sum l_i p_i$. Ponieważ jednak c jest z założenia optymalny, ten nowy kod nie może być bezprefiksowy. Istnieje zatem inne słowo $c(a_i)$ długości L, takie że $\omega \lhd c(a_i)$. Ale oznacza to, że $c(a_i)$ i $c(a_k)$ różnią się tylko ostatnim bitem. Wystarczy teraz przenumerować słowa kodowe o długości L tak, aby $c(a_i)$ znalazło się na przedostatniej pozycji $c(a_{k-1})$.

DOWÓD TWIERDZENIA. Indukcja względem liczby liter k, $\mathcal{A} = \{a_1, a_2, ..., a_k\}$.

Niech $h : \mathcal{A} \to \{0,1\}^*$ będzie kodem Huffmana, a $c : \mathcal{A} \to \{0,1\}^*$ kodem optymalnym. Niech $\hat{h}$ będzie kodem Huffmana na alfabecie $\{a_1, a_2, ..., a_{k-1} \cup a_k\}$, gdzie symbol kolektywny $a_{k-1} \cup a_k$ ma prawdopodobieństwo $p_{k-1} + p_k$. Z $(*)$ wynika, że

$$\mathbb{E}[h(A)] \;=\; \mathbb{E}[\hat{h}(\hat{A})] \;+\; (p_{k-1} + p_k)\,.$$

Dla optymalnego kodu c na podstawie lematu możemy założyć, że $c(a_{k-1}) = \omega 0$ i $c(a_k) = \omega 1$. Zdefiniujmy teraz nowy kod $\hat{c}$ na $\hat{\mathcal{A}}$ podstawiając $\hat{c}(a_{k-1} \cup a_k) = \omega$ oraz $\hat{c}(a_i) = c(a_i)$ dla $i = 1, 2, ..., k-2$. Ten kod jest bezprefiksowy i

$$\mathbb{E}[c(A)] \;=\; \mathbb{E}[\hat{c}(\hat{A})] \;+\; (p_{k-1} + p_k)\,.$$

Na podstawie założenia indukcyjnego $\hat{h}$ jest optymalny, więc $\mathbb{E}[\hat{h}(\hat{A})] \leq \mathbb{E}[\hat{c}(\hat{A})]$, skąd

$$\mathbb{E}[h(A)] \;\leq \mathbb{E}[c(A)]\,.$$

Ale c realizuje minimum średniej długości kodu, więc w nierówności powyżej musi zachodzić równość, a zatem h jest optymalny.

Kod blokowy

Zamiast pojedynczych liter $a \in \mathcal{A}$ można kodować ich bloki

$$\alpha \;=\; \underbrace{a_1 a_2 ... a_r}\; \underbrace{a_{r+1} a_{r+2} ... a_{2r}}\; ...$$

Kod blokowy $c_r : \mathcal{A}^r \to \mathcal{B}^*$, zgodnie z twierdzeniem Shannona-Fano

$$\frac{H(A_1, A_2, ..., A_r)}{\log D} \;\leq\; \mathbb{E}[c_r(A_1, A_2, ..., A_r)] \;<\; 1 + \frac{H(A_1, A_2, ..., A_r)}{\log D}$$

czyli na jedną literę $a \in \mathcal{A}$

$$\frac{H(A_1, A_2, ..., A_r)}{r \log D} \;\leq\; \frac{\mathbb{E}[c_r(A_1, A_2, ..., A_r)]}{r} \;<\; \frac{1}{r} + \frac{H(A_1, A_2, ..., A_r)}{r \log D}\,.$$

Oczekiwana długość kodu na 1 literę wynosi więc w przybliżeniu $\dfrac{H(A_1, A_2, ..., A_r)}{r \log D}$.

$$H(A_1, A_2, ..., A_r) \;\leq\; H(A_1) + H(A_2) \;+\; \cdots + H(A_r) \;=\; rH(A_1)\,,$$

więc jeśli zmienne A_i są niezależne, średnia długość kodu na literę dąży do $\dfrac{H(A_1)}{\log D}$, gdy długość bloku rośnie do nieskończoności.

W rzeczywistych komunikatach (teksty, obrazy, cyfrowy zapis dźwięku) kolejne "znaki" są silnie zależne, więc $H(A_1, ..., A_r) < rH(A_1)$. Załóżmy np., że rozkład dla kolejnej litery zależny jest od co najwyżej p poprzednich liter

$$P(A_r = a_r \mid A_{r-1} = a_{r-1},\; \ldots,\; A_1 = a_1) \\ =\; P(A_r = a_r \mid A_{r-1} = a_{r-1},\; \ldots,\; A_{r-p} = a_{r-p})\,,$$

można tą samą drogą pokazać, że wtedy

$$\frac{H(A_p \mid A_1, A_2, ..., A_{p-1})}{\log D} \;\leq\; \frac{\mathbb{E}[c_r(A_1, ..., A_r)]}{r} \;<\; \frac{1}{r} + \frac{H(A_p \mid A_1, A_2, ..., A_{p-1})}{\log D}$$

Przypomnienie: $H(A_r \mid A_1, ..., A_{r-1}) \;=\; H(A_1, ..., A_r) \;-\; H(A_1, ..., A_{r-1})$.

Oczywiście w takiej sytuacji kod bokowy długości p jest właściwym wyborem.