Wzory

- Informacja: I(x) $-\log_2 P(x)$
- Entropia: $H(X) = \sum P(x)$.
- Śr. długość kodu: $L = \sum P_i$
- Nierówność Krafta: $\sum 2^{-l_i} \le 1$
- Wydajność: $r = \frac{1}{n} \log |\mathcal{C}|$

• Hamming: d = liczba różnic

- Wykrywanie: d-1, Korekcja: $\lfloor (d-1)/2 \rfloor$
- Wsp. informacji: η = k/n
- Kod doskonały: $\sum_{i=0}^{t} \binom{n}{i} \leq$ 2^{n-k}
- Cykliczność: przesunięcie ⇒ w kodzie
- Wielomian g(x): $g(x) \mid x^n + 1$ $w \mathbb{Z}_2[x]$

Przykłady:

$$I(1/2) = 1$$
, $H(\{1/2, 1/2\}) = 1$, $L = 0.5 \cdot 1 + 0.5 \cdot 2 = 1.5$, $\sum 2^{-l}i = 0.75$

Kod Huffmana

Algorytm:

- $\bullet\;$ Symbole a_i z wagami (P_i lub liczba wystąpień)
- Połącz 2 najmniejsze wagi \rightarrow nowe drzewo
- Krawędzie: lewa = 0, prawa =
- Powtarzaj aż zostanie jedno
- Kody = ścieżki z korzenia

Wariant dynamiczny:

- Drzewo budowane w trakcie od-
- Nowe symbole ⇒ użycie NYT (Not Yet Transmitted)
- Nowy symbol dodany do drzewa, reszta aktualizowana
- Znane symbole: kod z aktualnego drzewa

Kod prefiksowy, jednoznaczny, optymalny (min. L)

Kod Shannon-Fano

- Dla a_i z p_i : oblic
z $l_i = \lceil -\log_2 p_i \rceil$
- Oblicz w_j : $w_1 = 0$, $w_j = \sum_{i=1}^{j-1} 2^{l_j l_i}$
- $\bullet~$ Kod $a_j\colon$ binarna postać w_j dopełniona zerami z lewej do l_j bitów

Kod prefiksowy, nie zawsze optymalny. Przykład: $p = [0.5, 0.25, 0.25] \Rightarrow l = [1, 2, 2]$ $w = [0, 2, 3] \Rightarrow \text{kody: } 0, 10, 11$

Kod Golomba/Rice

Dla liczby n i parametru M:

 $q=\left\lfloor\frac{n}{M}\right\rfloor, r=n \bmod M$ Golomb: kod = q w unarnym + r w binarnym (długość zależna od M) Rice: Golomb z $M=2^k \Rightarrow r=k$ -bitowy binarny

Kompresja bezstratna o stałej długości wyjściowej (kod: n bitów):

- $\bullet\;$ Dla alfabetu a_i z
 p_i budujemy drzewo od najczęstszych symboli.
- Rozwijaj liść S: twórz ciągi Sa_1, \dots, Sa_m z wagą $P \cdot p_i.$
- Powtarzaj aż liczba liści (kodów) = 2^n .
- Kody to indeksy liści (słowa) \Rightarrow długość kodu = n.

Kod nieprefiksowy, ale jednoznaczny przy stałej długości.

Przykład: p(a) = 0.6, $p(b) = 0.4 \Rightarrow$ start: a, bRozwijaj a: aa, ab itd. $a\dot{z}$ do np. 4 słów (dla n=2)

Kodowanie Eliasa

Dla liczby całkowitej x, nieznanej długości binarnej:

- $\bullet \ \ n = \lfloor \log_2 x \rfloor + 1$
- $\gamma(x) = 0^{n-1} (x)_2$
- $\omega(x) = \omega(n-1) + (x)_2 + 0$ (rekurencja)

 $\begin{array}{ll} \mathbf{Przykład:} \ \, x=9, \, (x)_2=1001, \, n=4 \\ \gamma(9)=0001001 & \delta(9)=\gamma(4)+001 \end{array}$ $\omega(9) = \omega(3) + 1001 + 0$

Kodowanie Fibonacciego

- Reprezentacja liczby x: $x = \sum a_i f_i$, $a_i \in \{0, 1\}$
- Warunek: brak dwóch jedynek obok siebie (Zeckendorfa)
- Na końcu dodajemy dodatkowe 1 jako znacznik końca

Przykład: x = 6, $6 = 5 + 1 = f_4 + f_1 \Rightarrow 100100 + 1 = 1001001$

Kodowanie arytmetyczne

Kodowanie:

- l ← l + d · F(j)

- d = p l

Dekodowanie:

- d = p l
- Znajdź j: $F(j) \leq$ F(j+1)

1

- $p \leftarrow l + d \cdot F(j+1)$
- $l \leftarrow l + d \cdot F(j)$
- $p \leftarrow l + d \cdot F(j+1)$

Przykład: F = [0, 0.5, 1], dla ciagu AB, A = 0-0.5, B = 0.5-1Koniec przedziału: liczba z zakresu (0.25, 0.75) reprezentuje słowo AB

Kodowanie słownikowe

LZ78

 $(o,l,k) = \mathcal{C}_{i-o} \dots \mathcal{C}_{i-o+l} \; k_{o,j}^{\mathrm{Znajd\hat{z}}} \; \mathrm{najd\hat{z}}$ najdłuższy prefiks o — przesunięcie, l — dłu-Dodaj: prefiks + nowy gość, k — nowy znak **Przykład:** aabcaabc (0,0,a), (0,0,a), (0,0,a)znak Zakoduj: (i, k)(0,0,b), (0,0,a), (0,0,a), (0,0,c), (4,3,c)Przykład: ababcbababaaaaaa

(0,a), (0,b), (1,b), (0,c), (2,a), (4,a), (6,a), (7,a)

LZW

Startowy słownik = alfa-Kodujemy indeksy ciągów Dodaj nowy ciąg gdy brak Przykład: abababa a (97), b (98), ab (256), ba (257), aba (258) Kody: 97, 98, 256, 257, 258

BWT / bzip2

LZ77

1. Dla słowa tworzymy wszystkie rotacje cykliczne. 2. Sortujemy je leksykograficznie. 3. Zapisujemy ostatnią kolumnę (BWT) i numer wiersza, gdzie znajduje się oryginalne słowo. 4. Odwracanie: znając BWT i indeks, możemy

	e	h]					
	h	О	ĺ	0	1	2	3	4
Ì	ll	е	1	е	h	1	1	0
	lo	1]	2	0	3	4	1
	0	1	1					

Move-To-Front (MTF)

Transformacja zmniejszająca entropię (często po BWT).

- Start: tabela liter w porządku alfabetycznym
- Dla każdej litery: zapisz jej pozycję, przesuń ją na początek

 $h\,\rightarrow\,1$ tabela: [h, e, l, o] tabela: [e, h, l, o] tabela: [l, e, h, o] $e \rightarrow 1$ $l \rightarrow 2$ Przykład: hello Alfabet: [e, h, l, o] $1 \rightarrow 0$ tabela: [l, e, h, o] tabela: [o, l, e, h]

Wvnik: 11203

PPM

- $\bullet~$ Dla symbolu sprawdzamy jego kontekst (max długość n).
- Przegladamy konteksty: $n \to 0 \to -1$
- Dla n > 0: zliczamy wystapienia; dla −1: tylko obecność.
- Budujemy model prawdopodobieństw np. do Huffmana.

!: szukaj symbolu w najdłuższym kontekście. Jeśli nie ma – wypisz ESC i przejdź

krótszego.	Kontekst	Symbol	LICZIIIK	- 1	Kontekst	Symbol	LICZIIIK
arotozego.	t	ESC	1	- [th	ESC	1
Symbol Licznik		h	1			i	1
Symbol ESC 1	h	ESC	1	1	hi	ESC	1
t t 1	İ	i	1	- 1	i i	s	1
n h 1	i	ESC	1	- [is	ESC	1
s i 1	İ	s	2	- 1		-	1
s s 2	s	ESC	1	1	8-	ESC	1
		-	1	- 1		i	1
	-	ESC	1	Ī	-i	ESC	1
		i	1 1			8	1

Kody Hamminga

- Macierz parzystości H: kolumny = binarne zapisy $\underline{1}, \ldots, n$

$$H_H(3) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

• Macierz generująca $G\colon\thinspace [I_k\mid P],$ gdzie Ppochodzi z H

$$G_H(3) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Rozszerzony kod Hamminga: dodany bit parzystości – długość n+1Cykliczny kod Hamminga: generowany przez g(x), który dzieli $x^n + 1 \le \mathbb{Z}_2[x]$ Kod Hamminga (7,4) jest kodem doskonałym dla d=3



Fig. 1: Exam Emotional Support System (przykład: jpeg-ls)

 ${\bf Hatsune}\ {\bf przypomina:}\ {\rm Nie}\ {\rm da}\ {\rm się}$ skompresować wszystkich możliwych zbiór skompresowanych musi być mniejszy => nie istnieje bijekcja.

Lew: Twoja pewność siebie jest jak kod Huffmana konkretna i skuteczna.

Ryby: Chaos w danych? Nie szkodzi PPM poradzi sobie nawet z tobą.

Wodnik: BWT to — nieoczytwoja dusza wista, ale piękna w dekodowaniu.

Koziorożec: Twoja determinacja przypomina kod Hamminga wykrywa, poprawia i nie odpuszcza.

Fig. 2: Zakreśl, jeśli: Błąd v obliczeniach? Brak pomysłu? Nie rozumiesz pytania? "Wszystko da się skompresować. Poza czasem na egzaminie.

- autor nieznany, ale na pewno sfrustrowany