1 Wzory

Informacja zdarzenia A:

$$I(A) = -\log_x P(A)$$

Entropia źródła X ze zdarzeniami $A_1, \ldots, A_n$:

$$H(X) = \sum_{i=1}^{n} P(A_i) I(A_i)$$

Średnia długość kodu $\mathcal{C}$:

$$I(\mathcal{C}) = \sum_{i=1}^{n} P(\mathcal{C}_i) \cdot l_i$$

Nierówność Krafta (warunek konieczny jednoznacznej dekodowalności):

$$K(\mathcal{C}) = \sum_{i=1}^{n} 2^{-l_i} \leq 1$$

Współczynnik informacji kodu $\mathcal{C}$:

$$\frac{1}{n} \log |\mathcal{C}|$$

2 Kod Huffmana

Znajdź dwa najrzadziej występujące elementy i połącz je w jeden element o prawdopodobieństwie $p_1 + p_2$. Rozróżnij je 0 lub 1. Powtórz ten krok na liście $n-1$ długiej aż zostanie jeden element.

Jeśli nie znamy prawdopodobieństw, to możemy drzewo tworzyć dynamicznie, traktując ilość wystąpień jako wagę, które łączymy tworząc poddrzewa.

3 Kod Shannon-Fano

Dla symboli $a_1, \ldots, a_n$ o prawdopodobieństwach $p_1, \ldots, p_n$, ustalmy kody długości $l_n = \lceil -\log p_i \rceil$. Następnie zdefiniujmy zmienne pomocnicze $w_1, \ldots w_n$ jako:

$$w_1 = 0, w_j = \sum_{i=1}^{j-1} 2^{l_j - l_i}$$

Jeżeli $\lceil \log w_j \rceil = l_j$ to j-te słowo kodowe jest binarną reprezentacją w_j. Jeżeli $\lceil \log w_j \rceil < l_j$ to reprezentację uzupełniamy zerami z lewej strony.

4 Kod Tunstalla

Chcemy stworzyć kod na n bitach dla $a_1, \ldots, a_m$ symboli o prawdopodobieństwach $p_1, \ldots, p_m$. Tworzenie kodu Tunstalla polega na iteracyjnym wyborze ze zbioru symbolu o największym prawdopodobieństwie S i łączenie go z wszystkimi innymi symbolami tworząc symbole Sa_m, nadając im prawdopodobieństwa $P \cdot p_m$. Proces ten powtarzamy aż do uzyskania kodu o długości n.

5 Kodowanie Eliasa

$$n = \lfloor \log_2(x) \rfloor + 1$$

5.1 γ

$$\gamma(x) = 0^{n-1}(x)_2$$

5.2 δ

$$\delta(x) = \gamma(n) + (x)_2$$

5.3 ω

Na koniec umieszczane jest 0, potem kodowana jest liczba $k = x$. Potem ten krok jest powtarzany dla $k = n - 1$ gdzie n to liczba bitów z poprzedniego kroku.

$$\omega(x) = \omega(n-1) + (x)_2 + 0$$

6 Kodowanie Fibonacciego

$$f_0 = f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2} : n \geq 2$$

$$x = \sum_{i=0} a_i \cdot f_i, a_i \in \{0, 1\}$$

7 Kodowanie arytmetyczne

- $[l, p) = [0, 1)$
- $d = p - l$
- $p = l + d \cdot F(j+1)$
- $l = l + F(j)d$

8 Kodowanie słownikowe

8.1 LZ77

$$(o, l, k) = \mathcal{C}_{i-o} \cdots \mathcal{C}_{i-o+l} k$$

8.2 LZ78

1. Szukaj w słowniku najdłuższy prefiks aktualnego okna, jeśli nie znajdziesz to użyj ϵ.
2. Dodaj prefiks + znak do słownika.
3. Zakoduj symbol jako (i, k), gdzie i to numer prefiksu w słowniku, a k to symbol.

$$(i, k) = s(i) + k$$

8.3 LZW

Podobne do LZ78, tylko że zaczynamy ze słownikiem.

$$(i) = s(i)$$

9 bzip2/BWT

Układamy tabelę z dwoma kolumnami. Pierwsza kolumna to słowo posortowane leksykograficznie. Druga kolumna to poprzedni znak. Na podstawie tej tabeli zapisujemy ostatnią kolumnę, i numer wiersza w którym w pierwszej kolumnie znajduje się początek słowa, a w drugiej kolumnie jego koniec.

e	h
h	o
ll	e
lo	l
o	l

0	1	2	3	4
e	h	l	l	o
2	0	3	4	1

10 Move To Front

Jest to transformacja zmniejszająca entropię. Zaczynamy od tabeli liter ze słowa posortowanych alfabetycznie. Następnie dla każdej litery ze słowa kodujemy jej pozycję w tabeli, a następnie przesuwamy ją na początek tabeli. W ten sposób hello to 11203.

11 PPM

Poprzez zbudowanie drzewa kontekstowego, które wyraża prawdopodobieństwo wystąpienia symbolu w danym kontekście, można potem zbudować bardzo dobry kod. W tym przypadku mamy kontekst o długości 2.

Kontekst	Symbol	Licznik
th	ESC	1
	i	1
hi	ESC	1
	s	1
is	ESC	1
	-	1
s-	ESC	1
	i	1
-i	ESC	1
	s	1

12 Kody Hamminga

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

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