Entropie zprávy (řádu 0 a vyšší), modelování. Statistické metody komprese dat

→ data redundancy removal
- short codes for common events

Model: statické, semi-adaptive, adaptive

Entropie S - finite set of source units, C - finite set of codewords $f: S \to C^+$

entropy of unit x_i $H_i = -\log_2 p_i$ – míra nevědomosti ?

$$H_{avg}(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \text{ bits}$$ – průměrná entropie – abeceda délky n (velikosti)
entropie vstupu !

entropy of source message $X = x_1 x_2 x_3 \dots x_k \in S^+$ $$H_0(X) = -\sum_{j=1}^{k} \log_2 p_j$$ – k je délka zprávy X

length of message X $$L(X) = \sum_{j=1}^{k} d_j \text{ bits}$$

redundancy of code K for message X:
$$R(X) = L(X) - H(X) = \sum_{j=1}^{k} (d_j + \log_2 p_j) \text{ bits}$$

average length of a codeword: $$L_{avg}(K) = \sum_{i=1}^{n} d_i p_i \text{ bits}$$

average redundancy of code K: $$R_{avg}(K) = L_{avg}(K) - H_{avg}(K)$$

Empirical entropy ▷ 0-th order ($T \in S^+$)

$$H_0(T) = -\sum_{a \in S} \frac{n_T^a}{n} \log_2 \frac{n_T^a}{n}$$ ← number of symbols a in message T

▷ k-th order entropy

$$H_k(T) = \frac{1}{n} \sum_{w \in S^k} |w_T| H_0(w_T)$$ ← symbols following each other in T

$$0 \le H_k(T) \le H_{k-1}(T) \le \dots \le H_0(T) \le \log_2 |S|$$

H_0 = T = abbbaabababbaaaababa

a	9/17
b	8/17

$$H_0 = -\left(\frac{9}{17}\cdot\log_2\frac{9}{17} + \frac{8}{17}\cdot\log_2\frac{8}{17}\right) = 0{,}9975$$

H_1 =

ω_T	
a	babbaabb
b	bbaabaaa

$$H_1 = \frac{1}{17}\cdot\left(8\cdot H_0(\underbrace{|babbaabb|}) + 8\cdot H_0(\underbrace{|bbaabaaa|})\right)$$

$$= \frac{1}{17}\cdot\left(8\left(\frac{5}{8}\cdot\log_2\frac{5}{8} + \frac{3}{8}\cdot\log_2\frac{3}{8}\right) + \dots\right) = 0{,}8983$$

Shannon - Fano

- dělení intervalů pravděpodobnosti na poloviny (co nejpřesněji)
- negarantuje optimální kód, jednoduché na implementaci

$$H_{avg} \le L_{avg} \le H_{avg} + 1$$

Huffman coding – optimální prefixový kód

- konstrukce bottom-up s využitím prioritní fronty

Sibling property – každý node má sourozence (kromě root uzlu)

- uzly se dají seřadit dle klesající pravděpodobnosti tak, že sourozenci budou v listu vedle sebe (lichý uzel na liché pozici, sudý pozice pro pravého)

Huffman code <=> Sibling property

Kraft - MacMillan Inequality

$$\sum_{i=1}^{n} 2^{-L_i} \le 1$$

- každý jednoznačný kód s délkami kódu $L_1 \dots L_n$ musí splňovat tuto podmínku

- pokud je suma větší než 1, kód nelze jednoznačně dekódovat

Aritmetické kódování

- kódujeme do intervalu rozděleného dle pravděpodobností jednotlivých symbolů
- konec se musí řešit EOF charakterem nebo explicitní délkou
- blížíme se entropii $H_{avg} = -\sum p(x_i) \log_2 p(x_i)$
- velmi dobré pro adaptivní model

Celočíselná implementace:

Low = xxxx 00... (0,0000)

High = yyyy 99... (0,9999)

- nejnižší hodnoty jsou poslány na výstup a proměnné doleva posunuty

$$Low = Low + (High - Low + 1)\, HighCP(x) / Totalfreq - 1$$

$$High = Low + (High - Low + 1)\, LowCP(x) / Totalfreq$$

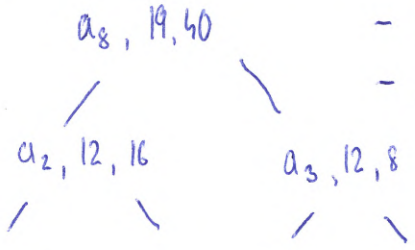
Underflow: Low = 49xxxx & High = 50yyyy

→ Low = 4xxxx0 , High = 5yyyy9 , counter ++

- pokud platí Low[0] = High[0] = 4
  - → output 4 a counter s 9
  - → v opačném případě 5 a nuly

Adaptivní aritmetické kódování

- využívá strom - každý uzel symbol, frekvence a cummulative freq. kvůli podstromu
  - pravděpodobnější blíže ke kořenu
  - strom balancovaný jako heap (také v poli)



[Range encoding
- kóduje output ne jako binární číslo, ale jako číslo jiné báze
- menší počet normalizací, složitější bitové operace]