Kódování

Kondr

11. února 2009

Obsah

Teorie informace

- matematický popis přenosu informace
- pouze kvantitativní parametry
- teorie optimálního kódování
- zakladatelem C. E. Shannon Claude Elwood Shannon (30. 4. 1916 - 2001)

Vymezení pojmů

Zpráva je uspořádaná posloupnost znaků určená k přenášení informací.

Kanál je cesta, na níž je zpráva vystavena možnosti poškození.

Symetrický kanál je kanál se stejnou pravděpobností chyby pro všechny znaky.

Zdroj je genrátor znaků konečné abecedy A.

Bezpaměťový zdroj generuje znaky nezávisle na předchozích znacích.

Vymezení pojmů

Možství informace ve zprávě je $-\log_2(p)$, kde p je pravděpodobnost této zprávy.

Entropie zdroje je očekávaná míra informace v jednom znaku.

$$H(p) = -\sum_{a} \in Ap(a)\log_2(p(a)).$$

Přitom $H(p) \leq log_2(p)$, rovnost $\iff$ $p(a) = \frac{1}{|A|}$.

Motivace

- reálné zdroje (obrázky, psaný text, hudba) mají relativně nízkou entropii
- jeden bit zprávy přenáší méně než jeden bit informace
- snaha o efektivnější využití

Huffmanovo kódování

- jeden znak možno chápat jako posloupnost $\log_2 |A|$ bitů
- frekventované posloupnosti nahradíme kratšími
- málo frekventované můžeme prodloužit
- kódování má dvě části: zabalování a rozbalování

Zabalování

- znaky seřadíme sestupně podle četnosti
- dva znaky a, b sloučíme do znaku (ab), jeho četnost je $p(a) + p(b)$
- zařadíme (ab) na správné místo v posloupnosti
- nakonec nám vznikne jeden znak
- jeho závorkování lze intepretovat jako strom

Rozbalování

- interpretace stromu
- vytvoření tabulky
- rekurzivně

Huffman v praxi

- JPEG, MP3 (+ transformace, downsampling, potlačení údajů)
- část algoritmu ve faxech a v PKZIPu (ZIP formát)
- nad $\mathbb{Z}_3$ lepší než morseovka

Aritmetické kódování

- zprávu bereme jako desetinné číslo z intervalu $\langle 0, 1 \rangle$
- tento interval zdeformujeme tak, aby nejčastější zprávy nebyly příliš hustě
- k rozlišení zpráv pak bude stačit méně znaků

Implementace

- interval $\langle 0, 1 \rangle$ rozdělíme na $|A|$ intervalů
- i-tá interval má velikost $p(a_i)$.
- na začátku označíme celý interval
- vždy když načteme znak a_i, rozdělíme označený interval na $|A|$ dílů a označíme i-tý
- když známe nějaké cifry, pošleme je na výstup
- tím se uvolní místo pro bitový posun $\Rightarrow$ můžeme načítat další znaky.

Aritmetické kódování v praxi

- dokonalejší než Huffman (pro některá rozdělení výrazně)
- starší verze `bzip`
- podporováno JPEGem
- americké patenty :-(

Slovníkové algoritmy

- předchozí algoritmy optimalizovaly pouze kódování znaků
- v reálných datech se často opakují celé sekvence znaků
- ideální stav: každá dostatečně dlouhá sekvence v souboru pouze jednou
- další výskyty nahradit odkazem na originál

Lempel – Ziv – Welch

- udržujeme si slovník
- na začátku obsahuje symboly z A
- když se v textu vyskytne slovo w ze slovníku následované symbolem a, uložíme do slovníku wa a na výstup vypíšeme kód pro w.

Časování

- v elektronice často 1=jde proud, 0=nejde proud
- důležitý čas, kdy se to měří
- i při malé chybě snadno zaměníme 1111111111 a 11111111111

4B5B

0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Hammingovská vzdálenost

- slovo je posloupnost znaků
- vzdálenost slov u, v stejné délky je počet znaků, ve kterých se liší
- kódování přiřadí každému slovu kódové slovo z kódu C.
- při přenosu kódového slova nastane k chyb
- pokud je vzdálenost každých dvou kódových slov alespoň $k + 1$, podaří se chyby detekovat
- pokud je vzdálenost každých dvou kódových slov alespoň $2k + 1$, podaří se chyby opravit
- čím větší je minimální vzdálenost dvou kódových slov, tím méně slov může kód dané délky obsahovat

Lineární kódy

- pokud chceme např. 2^{256} kódových slov, je nepraktické pamatovat si je všechna
- lineární kód splňuje, že pokud jsou u a v kódová slova, je $u \oplus w$ také kódové slovo ($\oplus$ je sčítání po znacích modulo $|A|$)
- příklad pro $|A| = 3$: $1202 \oplus 1221 = 2120$, $212 \oplus 121 = 000$.
- každý kód jde zapsat maticí generátorů
- příklad: binární kód $C = \{1100, 1111, 1010, 0101, 1001, 0110, 0000, 0011\}$ má generátory $G = \begin{pmatrix} 1100 \\ 0110 \\ 0011 \end{pmatrix}$
- druhá možnost: kontrolní matice (ortogonální doplněk) – pro předchozí kód $H = (1\,1\,1\,1)$

Kódování a dekódování

Kódování

- slovu $w = a_1 a_2 \ldots a_k$ přiřadíme kódové slovo $c(w) = a_1 g_1 \oplus a_2 g_2 \oplus \cdots \oplus a_k g_k$, g_i jsou generátory
- maticově $c(w) = wG$

Dekódování

- slovu w' přiřadíme slovo s, tzv. syndrom
- i-tý symbol s je skalární součin w' s i-tým řádkem H
- vyrobíme si tabulku, v níž v prvním řádku jsou kódová slova (počínaje $00\ldots0$), v každém dalším řádku jsou slova se stejným syndromem
- podle syndromu určíme řádek, v něm hledáme

Omezení

- omezení kulového pokrytí

$$M \leq \sum_{i=0}^{i=d} \binom{n}{i} |A-1|^i$$

- kódy, které toto splňují jsou perfektní
- Singletonova hranice pro lineární kód dimenze k v A^n:

$$k \leq n+1-d$$

- kód který hranici splňuje tuto hranici je MDS (maximum distance separable)

Lineární kódy kolem nás

- rodné číslo – $|A| = 11$, $H = (1\ -1\ 1\ -1\ 1\ -1\ 1\ -1\ 1\ -1)$
- ISBN – $|A| = 11$, $H = (1\,2\,3\,4\,5\,6\,7\,8\,9\,10)$
- číslo účtu – $|A| = 11$, $H_1 = (16\,8\,4\,2\,1)$,
  $H_2 = (128\,64\,32\,16\,8\,4\,2\,1)$
- přenos dat z družic

Cyklické kódy

- ikdyž 256 slov je málo, 1 je ještě méně
- 1 slovo délky 256 a všechna jeho otočení generují lineární kód
- posunutím kódového slova dostaneme kódové slovo
- vhodný formalismus: polynomy
- slovo $a_0 a_1 ... a_{n-1}$ odpovídá $a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$
- ztotožňujeme x^n a 1
- bitový posun je pak násobení x-em
- kódování: násobení polynomem
- dekódování: dělení polynomem. Zbytek $\approx$ syndrom.

CRC

- Cyclic redundancy check
- zbytek po dělení vhodným polynomem
- použití např. v PNG