KPI 16.2.2004

Introduction to Data Compression, K. SAYOOD

CS/~JIRASEK/KPI

KÓDOVANIE

slová v abecede : vstupná abeceda $\Sigma_S = \{s_1 \dots s_m\}$

slová $\subseteq \Sigma_S$ .. reťazce $w = s_{i_1} \dots s_{i_k} \in \Sigma_S^*$

Σ_C .. kódovacia abeceda

$= \{c_1, \dots c_n\}$ priradenie $s_i \mapsto K(s_i) = c_{1}, \dots c_e \in \Sigma_C^*$

$K: \Sigma_S \to \Sigma_C^*$

$s_i \to \boxed{\text{kóder}} \to c_1 \dots c_e$ prenos $\rightsquigarrow \boxed{\text{dekóder}} \to \Sigma_C$

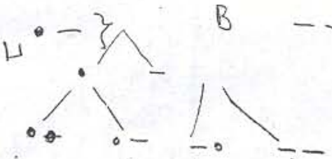
v čase / priestore

- bezstratové metódy (loseless): najprv jednoznačné kódovanie a dekódovanie $K^{-1}(K(w)) = w$
  - bez šumu $K(w)$ sa pri prenose nemení
  - šum v prenos. kanále
- stratová metóda (lossy)

morseovka $\Sigma_S = \{A, \dots, Z\}$ A $\cdot -$

$\Sigma_C = \{\sqcup, \cdot, -\}$ B $- - - \cdot$

stromový zápis



teplota vody: ľadová studená vlažná teplá

0 01 011 111

001 ľad, stud

010 11 01 (odhad, to je jednoznač. dekódovateľné)

0011 1111 | tu už nie je jednoznačnosť (prefix kód)

Kódovanie je jednoznačné (= kód je jednoznačne dekód.)

ak $K(s_{i_1}), \dots, K(s_{i_r}) = K(s_{j_1}) \dots K(s_{j_e})$

tak $k = l$ & $\forall q: K(s_{i_q}) = K(s_{j_q})$

blok. kódy sú jednoznačne dekódovateľné

$\llcorner \forall i: dl(K(s_i)) = const$

prefix. kód: kódové slovo nie je časťou iného kód. slova

$\forall s_1, s_2 \in \Sigma_S$: $K(s_1)$ nie je zač. kódu $K(s_2)$

$K(s_2)$ — " — $K(s_1)$

prefix kód je jednoznačne dekódovateľný

0 10 11] prefix kód

(7) sporom

nech $K(a_1) \dots K(a_r) = K(b_1) \dots K(b_s)$

$a_i, b_i \in \Sigma_S$

v morseovke $E = \cdot \sqcup$ medzera na konci rozhodí

$A = \cdot - \sqcup$ prefix. kód

chcem 0, 1, .. 9 do binár. kódu, ak sa najčastejšie vyskytujú 0, 1

$0 \to 00$ $2 \to 1010$ $4 \to 10006 \to$ $8 \to$

$1 \to 01$ $3 \to 1100$ $5 \to$ $7 \to$ $9 \to$

$\boxed{V}$ (Kraftova nerovnosť) $|\Sigma_S| = n$

Kódovanie n znakmi ($|\Sigma_C| = m$) môžeme urobiť prefixovým kódom, akk platí Kraftova nerovnosť s dĺžkami kód. slov $l_1, \dots l_n$

$$\sum_{i=1}^{n} m^{-l_i} \le 1$$

$\Leftarrow$ $l_1 \le l_2 \le \dots \le l_n$ BÚNV

indukciou

- zvolím kód pre prvý znak (nezáleží na ičom, len na dĺžke) $K(a_1)$ dĺžky l_1
- v IP máme $K(a_1), \dots K(a_r)$ a $K(a_i) = l_i$

chceme kódovať a_r, možností kódov dĺžky l_r v m znakoch je m^{l_r}

nemôžem použiť tie, kt. tvoria prefixy

1 slovo dĺžky l_1 vyradím $m^{l_r - l_1} = m^{l_r - l_1}$

- kód. slová dĺžky l_2: vyradím $m^{l_r - l_2}$

spolu vyradím $m^{l_r - l_1} + m^{l_r - l_2} + \dots + m^{l_r - l_{r-1}}$

mám rôznych slov m^{l_r}

platí Kraft. nerovnosť $m^{-l_1} + \dots + m^{-l_r} \le 1 \quad / m^{l_r}$

$\underbrace{m^{l_r - l_1} + \dots + m^{l_r - l_{r-1}}}_{\text{vyradené}} + 1 \le m^{l_r}$

teda je na nakódovanie 1 kód mám ostal voľný

$\Rightarrow$ majme prefix kód s dĺžkami $l_1, \dots l_n$

BÚNV $l_1 \le \dots \le l_n$

existuje m^{l_n} slov dĺžky l_n s prefix. vlastnosťou

vezmime slová s prefixom $K(a_1)$ $m^{l_n - l_1}$

skupiny sú disjunktné s prefixom $K(a_2)$ $m^{l_n - l_2}$

v prefixovosti s prefixom $K(a_n)$ $m^{l_n - l_n}$ $\le m^{l_n}$ všetkých ale v [PR] s rovnou sadou platí Kraft, ale nie je to prefix kód └ postfix (treba dekódovať odzadu)

0	01	011	111	} postfix
E	S	V	T	
1/2	1/4	1/8	1/8	= 1 .. platí Kraft

[V] Macmillanova veta

Pre každé jednoznačne dekódovateľné kódovanie platí Kraftova nerovnosť

(dá sa teoreticky obmedziť na prefix)

(D)

nech $l_1, \dots l_m$ dĺžky kód slov v jednoznačne dekód. kódovaní

$c = \sum_{i=1}^{m} n^{-l_i}$. Uvažujme $c^k = \left(\sum_{i=1}^{m} n^{-l_i}\right)^k =$

$= \sum_{i_1, \dots, i_k}^{m} n^{-(l_{i_1} + \dots + l_{i_k})} = \sum_{i=1}^{k l_m} S_i \, n^{-i}$ nejaký koeficient

Táto suma môže vzniknúť, tak aj mriežke rozmiestňovať na DG: $S_i \le n^i$ └ počet slov dĺžky i

$K(a_{i_1}) \dots K(a_{i_k})$ | $dl(\dots) = i$ Takých slov je S_i v jednoznač. dekód. $S_i \le n^i$ | logický krok! prečo?

potom $S_i n^{-i} \le n^i n^{-i} \Rightarrow \sum_{i=1}^{k l_m} 1 = k \, l_m$ ✓

$c^k \le k . l_m$ ak $c > 1$: $\frac{c^k}{k} \to \infty$

$c \le 1$ ✓

HUFFMANOV kód

$\{a_1 \dots a_m\} = \Sigma_S$, $P(a_i) = p_i$ .. pravdepodobnosť výskytu a_i vo zdroji

$\sum p_i = 1$ $l_i = dl(K(a_i))$

stredná dĺžka kód slova: $l = \sum_{i=1}^{m} l_i \cdot p_i$

			l			
A	0,4	0	0,4	} $l=2$	00	} $l=2$
B	0,2	10	$2 . 0{,}2 = 0{,}4$		01	
C	0,2	110	$3 . 0{,}2 = 0{,}6$		10	
D	0,2	111	$3 . 0{,}2 = 0{,}6$		11	

pre k znakov bude celý kód opäť dĺžky 2k

A	0,4	0	A	0,4	A	0,4	
B	0,2	10	B	0,2	B		
C	0,2	1100	C	0,2	C''		
D	0,1	1101	C*	0,2			
E	0,1	1110					

• usporiadaj pravdepodobnosti pri $p_i \ge p_{i+1}$

• urob redukciu. abeceda $a_1, a_2, \dots a_{m-2}, a^*$ $p_* = p_m + p_{m-1}$

• usporiadaj podľa veľkosti

ak $n=2$.. binárne už som OK

A 0,4	A 0,4	A 0,4	H 0,6	0_H	01_B	1_A
B 0,3	B 0,3	B 0,3	A 0,4	00_G	000_F	001_C
C 0,1	C 0,1	G 0,3		0001_D	0000_E	
D 0,1	F 0,2					
E 0,1						

A	1	0,4	0,4	} $l = 2{,}1$
B	01	0,3	0,6	
C	001	0,1	0,3	
D	0001	0,1	0,4	
E	0000	0,1	0,4	

lepšie to nejde

[L] je možné nahradiť nejaký kód, že $K(a_m)$ a $K'(a_{m-1})$ sa líšia len v poslednom znaku

$l_1^* \le l_2^* \le l_m^*$ └ rovnaké $p_1 \ge p_2 \ge \dots \ge p_m$

v posled. kód. slova $K(a_m)$ odstránim posled. znak, vznikne kód K' s kratšou strednou dĺžkou kód. slova

spor: máme kratší, ale neprefix kód K'

$K'(a_m)$ je prefixom nejakého $K^*(a_i)$

KPI 23.2.04

$\{a_1, \dots, a_n\}$ SHANNON - HARTLEY

$p_1 \dots p_n$

$\sum p_i = 1$

nech a_i je jav náhodný $I(a_i) = f(P(a_i))$ - chceme charakterizovať „prekvapivosť" danej informácie

spojitosť f — miera informácie

ak $a_1 \dots a_n$ sú nezávislé javy, potom

$$I(a_1 \cap \dots \cap a_n) = f(P(a_1 \cap \dots \cap a_n)) = f\left(\prod_{i=1}^{k} P(a_i)\right)$$

chceme $I\left(\bigcap_{i=1}^{k} a_i\right) = \sum_{i=1}^{k} I(a_i) = \sum_{i=1}^{k} f(P(a_i))$

nech $P(a_i) = x \ \forall i$. Potom $f\left(\prod_{i=1}^{k} P(a_i)\right) = f(x^k) = \sum_{i=1}^{k} f(P(a_i)) = k \cdot f(x)$

tj: $f(x^k) = k \cdot f(x)$

má platiť $f\left(\left(\frac{1}{2}\right)^k\right) = k \cdot f\left(\frac{1}{2}\right)$ ... $x = \frac{1}{2}$

$x^m = \frac{1}{2^{\frac{1}{m}}}$: $f\left(\left(\frac{1}{2^{\frac{1}{m}}}\right)^m\right) = m \cdot f\left(\frac{1}{2^{\frac{1}{m}}}\right)$, $f\left(\left(\frac{1}{2^{\frac{1}{m}}}\right)^m\right) = f(1/2)$

$x = \frac{1}{2^{1/m}}$ $\quad f\left(\left(\frac{1}{2^{1/m}}\right)^n\right) = n \cdot f\left(\frac{1}{2^{1/m}}\right) = \frac{1}{m} f\left(\frac{1}{2}\right)$

$f\left(\frac{1}{2^{n/m}}\right) = n \cdot \frac{1}{m} f\left(\frac{1}{2}\right) = \frac{n}{m} f(1/2)$

v limite: $f\left(\frac{1}{2^x}\right) = x \cdot f\left(\frac{1}{2}\right) \quad \forall x \in (0, \infty)$

nech $g(x) = f(x) + f\left(\frac{1}{2}\right) \log_2 x$

$$= \underbrace{f(2^{\log_2 x})}_{} + f\left(\tfrac{1}{2}\right) \log_2 x$$

$$= -\log_2 x \cdot f\left(\tfrac{1}{2}\right) + f\left(\tfrac{1}{2}\right) \log_2 x$$

$$= 0$$

tj: $f(x) = \underbrace{-f\left(\frac{1}{2}\right)}_{\text{konšt, môžem zvoliť}} \cdot \log_2 x = -k \cdot \log_2 x$

funkcia spĺňajúca 3 podmienky je jediná

Pr $P(0) = P(1) = 1/2$, $\Sigma_S = \{0, 1\}$

$f(\{0\}) = -k \cdot \log(1/2) = k$ ... ak $k = 1$ mám $\boxed{1 \text{ bit}}$

miera informácie:

$I(a_i) = -\log_2(P(a_i))$ bitov pre abecedu $\{0,1\}$ s rovnomerným rozdelením

$-\ln(P(a_i))$ natov

pravdepodobnosť dopadu mince $\Rightarrow I = 1$ v ideál. minci

ak $P(0) = 1/8$ $\quad P(1) = 7/8$

$I(0) = 3$ $\quad I(1) = 0{,}193$

časté javy majú menšiu mieru informácie

menej časté vyššiu I.

majme $\Sigma_S = \{a_1, \dots, a_n\}$, $p_1 \dots p_n$ } S.. systém

entrópia: miera neusporiadanosti celého systému

$$H(S) = \sum_{i=1}^{n} P(a_i) \cdot I(a_i) = \sum_{i=1}^{n} P(a_i) \cdot \overbrace{\log_2\left(\frac{1}{P(a_i)}\right)}^{-\log_2 P(a_i)} \quad (*)$$

rôzne definície entropie:

- popisom vlastností funkcie
  - : $H(p_1, \dots, p_n) \geq 0$
  - : H je spojitá
  - : H je symetrická : zmena poradia argumentov nemá vplyv
  - : H je koherentná

$$H(p_1, \dots, p_n) = H(p^*, p_3, \dots, p_n) + p^* \cdot H\left(\frac{p_1}{p^*}, \frac{p_2}{p^*}\right)$$

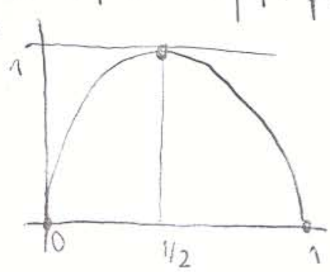
na každý výsledok p_1 alebo p_2 dám nový znak p^* $\quad p^* = p_1 + p_2$

jediná funkcia spĺňajúca 4 požiadavky je (*) až na konštantu.

PR $\Sigma = \{0, 1\}$ $\quad 0 \cdot \log 0 = 0$ jav, ktorý nikdy nenastane

$P(0) = p$ $\quad P(1) = 1 - p$

$H(\{0,1\}) = H(p, 1-p) = -p \log_2 p - (1-p) \log_2 (1-p)$



najv. entropia je pre rovnomerné rozdelenie 0, 1

$H(S) = 0$ akk $(\exists j)\left(P(a_j) = 1 \ \& \ \forall i \neq j: P(a_i) = 0\right)$

$\Leftarrow$ dos. do vzorca — na výstupe je stále ten istý znak

$\Rightarrow$ sporom

maximál entropia $\forall_{i\in[1,m]}\ p_i = \frac{1}{m}$ (☐)

$H(S) = \sum_{i=1}^{m} \frac{1}{m} \log_2 m = \log_2 m$ bitov

ich $H(S) - \underset{\log_2 m}{c} = \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} - \sum_{i=1}^{m} p_i \log_2 m$

$\sum_{i=1}^{m} p_i \left(\log_2 \frac{1}{p_i} - \log_2 m\right) = \frac{1}{\ln 2} \sum_{i=1}^{m} p_i \ln \frac{1}{p_i \cdot m}$

vieť: $\ln x \le x - 1$, rovnosť ak $x = 1$

chcete analýzu? ani počuť..., ale vieť to nie len takéto obrázky...

$\frac{1}{\ln 2} \sum_{i=1}^{m} p_i \left(\frac{1}{p_i \cdot m} - 1\right) = \frac{1}{\ln 2}\left(\underbrace{\sum_{i=1}^{m} \frac{1}{m}}_{1} - \underbrace{\sum_{i=1}^{m} p_i}_{1}\right) = 0$

$H(S) \le \log_2 m = H(S)$. $\log_2 m$ je najväčšia entropia pre rozdelenie (☐) ✓

---

ch $\Sigma_S = \{a_1, \dots, a_m\}$, $p_1 \dots p_m$. Zvolíme dĺžky kódových slov, aby $d_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil$

abecede $\{0,1\}$. vezmem d_i takto:

$\log_2 \frac{1}{p_i} \le d_i \le \left(\log_2 \frac{1}{p_i}\right) + 1$ (♡)

$\frac{1}{p_i} \ge 2^{d_i}$

$2^{-d_i} \le p_i$

$\sum 2^{-d_i} \le \sum_{i=1}^{m} p_i = 1$

$\sum 2^{-d_i} \le 1$: existuje prefixové kódovanie s požad. dĺžkami slov d_i

úloha nerovnosti

RD

každý prefix kód má strednú dĺžku kód. slova $L \ge H(S)$

$H(S) - L = \left(\sum_{i=1}^{m} p_i \cdot \log_2 \frac{1}{p_i}\right) - \left(\sum_{i=1}^{m} p_i\, d_i\right)$, $d_i = \log_2 2^{d_i}$

$\sum_{i=1}^{m} p_i \left(\log_2 \frac{1}{p_i} - \log_2 2^{d_i}\right) = \sum_{i=1}^{m} p_i \cdot \log_2 \frac{1}{p_i \cdot 2^{d_i}}$

vieť

$\frac{1}{\ln 2} \sum p_i \left(\frac{1}{p_i 2^{d_i}} - 1\right) = \frac{1}{\ln 2}\left(\sum_{i=1}^{m}\left(\frac{1}{2^{d_i}} - p_i\right)\right)$

$\frac{1}{\ln 2}\left(\sum_{i=1}^{m} 2^{-d_i} - \underbrace{\sum_{i=1}^{m} p_i}_{1}\right) \le 0$

Nedá sa dostať riadnym jednoznač. kódovaním pod entropiu strednou dĺžkou slova.

Ak Huffmanom je najlepšie kódovanie, mali by sme dôjsť presne k entropii.

majme kódovanie (♡).
$L_{min}(S)$ nech je min dĺžka kód. slova.

$L_{min}(S) \le L = \sum_{i=1}^{m} p_i d_i < \sum_{i=1}^{m} p_i \left(\log_2 \frac{1}{p_i} + 1\right) = \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} + \sum_{i=1}^{m} p_i = H(S) + 1$

$L_{min}(S) \le L \le H(S) + 1$

PE
majme $\underset{0{,}9\ \ 0{,}1}{\{0,1\}} = \Sigma_S$, Huff. kód: $L = 0{,}9 + 0{,}1 = 1$

$H(S) = 0{,}9 \cdot \log_2 0{,}9 + 0{,}1 \log_2 0{,}1 = 0{,}469$

$L \le H(S) + 1$
$1 \le 0{,}469 + 1 = 1{,}469$ ✓

kódujme po 2-ciach

00	0,81		0,81	0
01	0,09		0,19	10
10	0,09	0,10		110
11	0,01			111

$L = 0{,}81 \cdot 1 + 0{,}09 \cdot 2 + 0{,}09 \cdot 3 + 0{,}01 \cdot 3 = 1{,}29$

ale išli sme po dvojiciach: $\frac{L}{2} = 0{,}645$

môžem dosiahnuť entropiu?

vezmime S^2: dvojice znakov vstupnej abecedy

$H(S^2) = -\sum_{i_1=1}^{m} \sum_{i_2=1}^{m} p_{i_1} p_{i_2} \cdot \log_2 (p_{i_1} \cdot p_{i_2})$

$= -\sum\sum p_{i_1} p_{i_2} (\log_2 p_{i_1} + \log_2 p_{i_2})$

$= \left(-\sum p_{i_2} \sum p_{i_1} \log_2 p_{i_1}\right) - \left(\sum p_{i_1} \sum p_{i_2} \log p_{i_2}\right)$

$= H(S) \cdot \underbrace{\sum p_{i_2}}_{1} + H(S) \cdot \underbrace{\sum p_{i_1}}_{1} = 2H(S)$

$H(S^k) = k \cdot H(S)$: entropia pre k-tice brané na vstupe

$H(S^k) \le L_{min}(S^k) \le H(S^k) + 1$

$k \cdot H(S) \le L_{min}(S^k) \le k H(S) + 1$

$\lim_{k \to \infty} \frac{L_{min}(S^k)}{k} = H(S)$

pod pomer 0,469 sa nedostaneme nikdy, blížim sa k nemu kódovaním ...

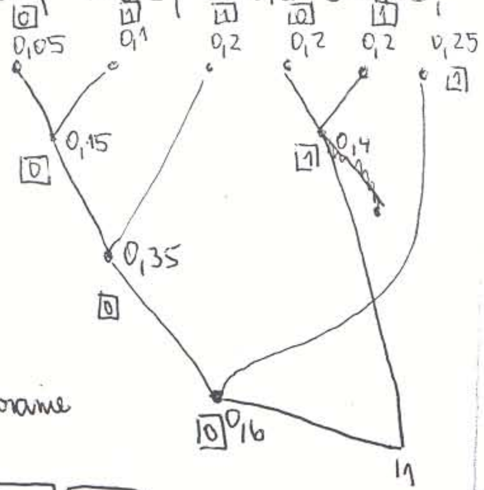
vzrastá veľkosť tabuľky

ii. aj zrýchlime: adaptívne Huffmanovo kódovanie
na začiatku sú pravdepodobnosti rovnaké,
neskôr sa upravujú

2 najznámejšie postupy (náhodná)

[Pr] a = 0,2; b = 0,05; c = 0,2; d = 0,2 e = 0,25,
f = 0,1

binárny kód



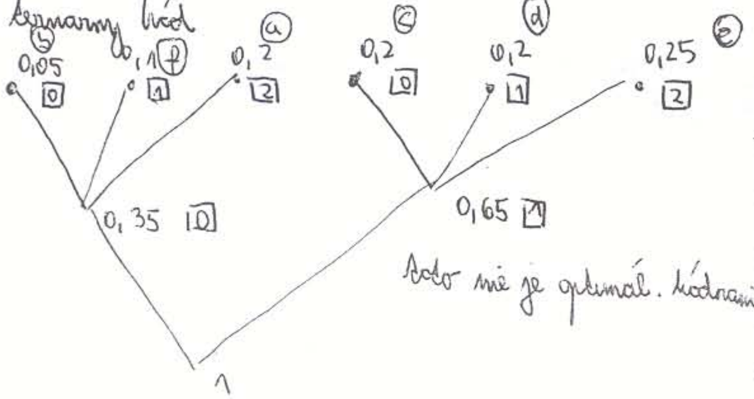
strom stále
- neohodnotené
vrcholy s
najmenšími pravdep.

kód pre znak je vymenovanie
hrán cesty k listu

d f b: 101 011 1001 1000

kódovanie: idem od koreňa podľa vstupu vyberám
vetvu. Ak som v liste, vypíšem príslušný
znak a vrátim sa k vrcholu

ternárny kód



toto nie je optimál. kódovanie! ako chcem)

d f b: 02 11 01 00 L = 2
? H ?

ale v posled. kroku
sa majú zlúčiť 3
vetvy, inak to
nie je optimálne

+ k(n−1)

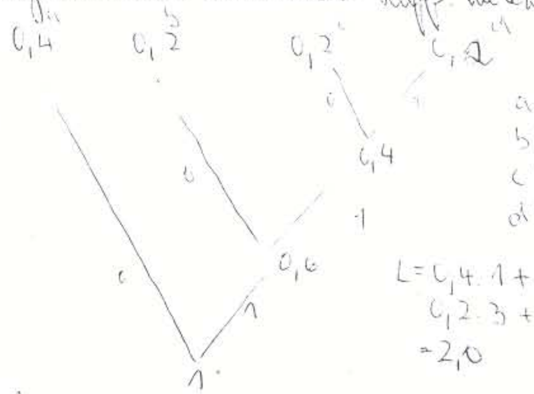
ternár. kód: 2k + 3 ak mám málo znakov, pridám
znaky s pravdep. 0:

$L_{optimál} = 1,7$

0 0,05 0,1 0,2 0,2 0,25 ? H ?

1. 3. 2004 III

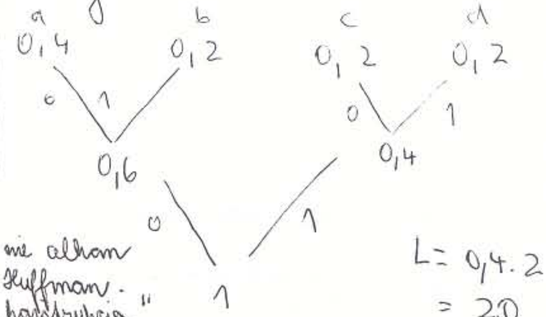
nejednoznačná konštrukcia Huff. kódovania



a ↦ 0
b ↦ 10
c ↦ 110
d ↦ 111

L = 0,4 · 1 + 0,2 · 2 + 0,2 · 3 + 0,2 · 3 = 2,0

ale aj



a ↦ 00
b ↦ 01
c ↦ 10
d ↦ 11

L = 0,4 · 2 + (0,2 · 2) · 3 = 2,0

nie ako Huffman. konštrukcia"

rôzne stromy, rovnaká efektívnosť

„krajšie" efektívnosť rovnaká

nový vrchol dám pred ostatné ⇒ najmenšia priemerná
(vypočítaný 0,4 dám/pred ním pred zadaná P(·) 0,4) dĺžka kód. slov
(min. variancia = najkratšie / najmenší rozptyl dĺžok)

kvôli Huffman a pod.

pri rovnakých pravdepodobnostiach nie je povedané,
ako to mám usporiadať (môžem si to popriraďovať
ako chcem)

---

musím 2 prechody: • zistí rozdelenie pravdepodobnosti výskytov
• kódovanie

(napr. 2 prechody súborom).

Faller
Gallager
Knuth

ADAPTÍVNE HUFFMANOVO KÓD. (FGK)

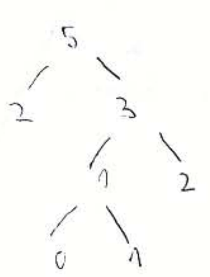
- na začiatku rovnaká P(·) písmen
- postupne vytváram strom a vytváram kódovanie
ďalších písmen

na konci mám celý
strom (ale mi to
vlastne netreba)

pri neadapt. musím posielať aj celý strom
pri FGK nie (strom sa pribudne)

binár. strom reprezentácia: vyjadrená sa sibling proper (SP)
binár. prefix strom je Huffmanov ak má (SP)

SP) Sibling property



i možné vrcholy

ch, aby ...

vrcholov ... nerastúcich

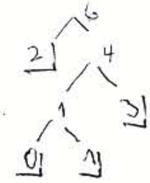
...postupnosť, súrodenci

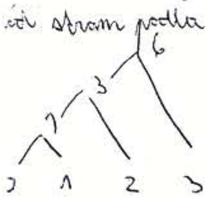
... za sebou, rodič je

a ... hodnota rodiča je súčet hodnôt

...

...strom podľa stromu $\xrightarrow{\text{toto mi je strom (SP)}}$





...goritmus

...jme $a_1, \dots a_m$ abecedu. Pošleme binár. strom

... $\lceil \log_2 m \rceil$ bitov. (1) prosté kódovanie

... A, ..., Z stačí $32 = 2^5 \Rightarrow 5$ bitov

FGK nemusí využiť všetkých 5 bitov, niekde stačí 4

$m = 2^l + r$ prvé znaky kódujeme $r+1$ bitov

$$a_1 \mapsto 00000 \quad l+1 \text{ bitov}$$

$$a_k \mapsto k-1 \quad l+1 \text{ bitov} \quad \Big\} k \le 2r$$

$$a_m \mapsto \left\{ \begin{matrix} k-r-1 \\ \underbrace{11111}_{m} \end{matrix} \right. \quad \Big\} l \text{ bitov}$$

angl. abeceda pre $m = 26 = 2^4 + 10 \quad 2r = 20$

$a_1 = 00000$

$a_2 = 00001$

⋮

$a_{20} = 10011$

$a_{21} = 1010 = (10)_2 \quad (21-10-1=10)$

$a_{25} = 1110$

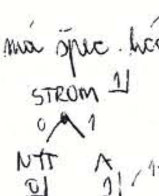
$a_{26} = 1111$

...prvé 4 bity $\le r$, musím ... aj piaty bit ... nemusím.

...) not yet transmitted znak: NYT, má špec. kód

... AARDVARK

$\overset{NYT}{\emptyset}$ VÝSTUP 00000 STROM



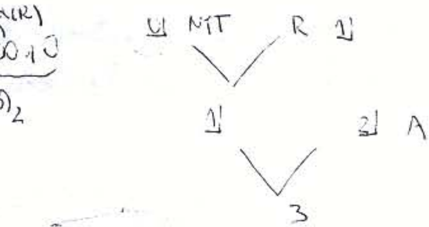
1× prenesené

nepárované

kód(A) = 1

000001

R $00000\,1\,\overbrace{\underbrace{010010}_{(18)_2}}^{\text{kód(R)}}$



priebežné kontroly

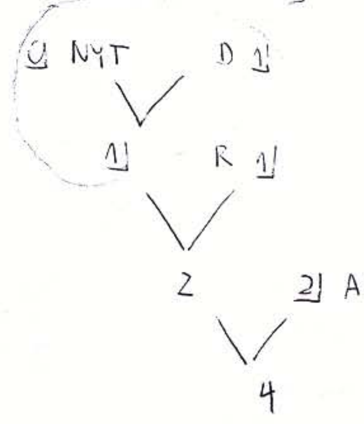
sibling property

D

00000 1 | 010010 |

00000 1 1

NYT



tu je problém:

V

00000 1 010010

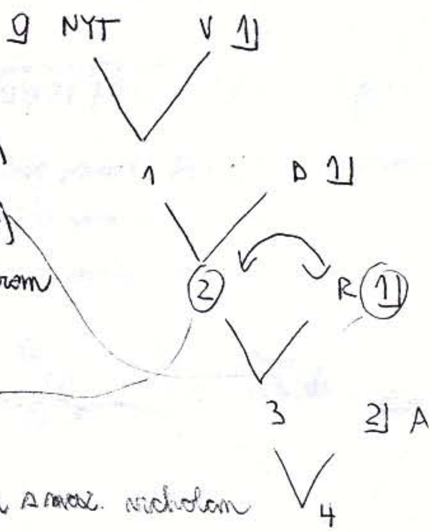
0000011 | 000

NYT kód(D)

už nemáme (SP)

musím reorganizovať strom

na rovnovážnu (SP)

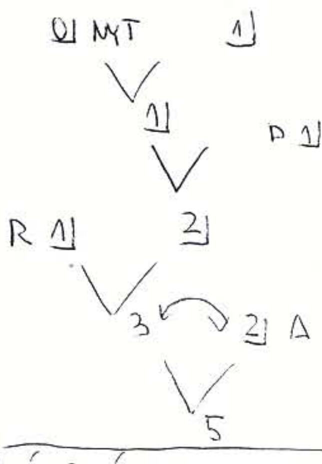


Tento vrchol mám

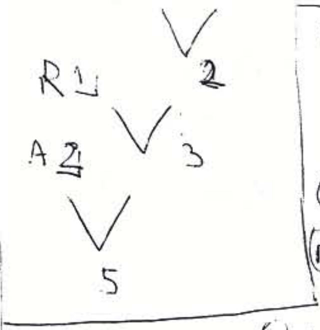
kde spraví

na ktorý 1 nahradím s max. vrcholom

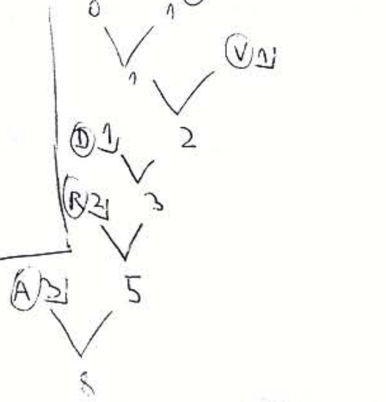
s touto hodnotou



ešte je to zle: opravíme



výsledný strom



v NYT : pošli kód NYT + pridaj kód znaku

NYT 0 / 1 znak

- znak pošli kód podľa stromu
- uprav strom

zväčši hodnotu vrcholu znaku o 1

ak rodič nie je maximálny v bloku vrcholov s rovnakou hodnotou, vzájomne ho vymeň s ... vrcholom najväčším v zmysle usporiadania, ale hodnotou rovnakou, opäť skontroluj strom

---

pri kódovaní obrázkov ČB (256 farieb = 256 znakov)

a sa skrátenie na 1/3.

zmena stromov = výmena pointrov

compact()

rob FGK

---

aritmetické kódovanie

ak je veľa písmeniek v Huff. neroste

ale ak. $p(A) = 0{,}99$ $p(B) = 0{,}01$

v rozšírenom musím zobrať veľké bloky - ale => veľká tabuľka, veľký strom

treba rozsekať ktoré bloky sú najlepšie

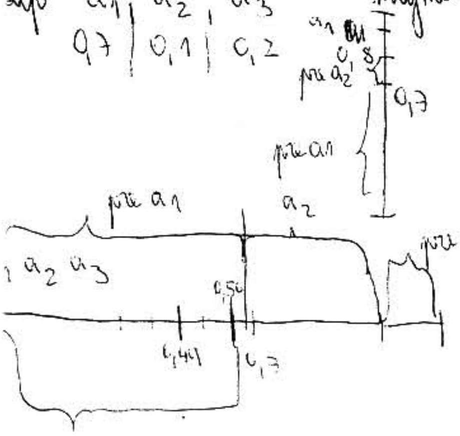
---

ideme kód

napr a_1 | a_2 | a_3

0,7 | 0,1 | 0,2

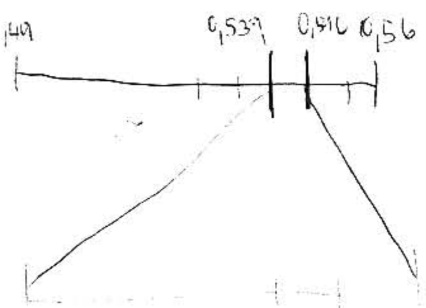
interval zvolím mapa čísla od $\langle 0,1)$



delím interval $\langle 0,1)$ podľa

pre a_3 pomeru pravdepodobnosti

$a_1 a_2 a_2 a_1 \to$ stav ... interval



$\langle 0{,}539; 0{,}546)$

posledná hodnota
- buď stred
- alebo najmenšia
- alebo ... najkratšia

---

prevod na presnosť (desatinné čísla)

viac čísel => lepší kód

n čísel → ...

dekódovanie!

napr. $0{,}55 \in \langle 0; 0{,}7) = a_1$

$0{,}55 \in \langle 0{,}49; 0{,}56) = a_2$

tiež musím mať rozdelenie pre dekódovanie

$l_0 = 0$ $u_1 = 1$

intervaly $\langle 0,0)$ (... postup) nemá kód 1

$C =$ kód $(a_{1}a_{4} \dots a_1)$

$$l_k = l_{k-1} + (u_{k-1} - l_{k-1}) \cdot \underbrace{\sum_{i=1}^{s_k - 1} P(a_i)}_{F(s_k - 1) \text{ .. distr. fcia}}$$

máme slovo $a_{s_1} a_{s_2} \dots a_{s_k}$

$$u_k = l_{k-1} + (u_{k-1} - l_{k-1}) \cdot F(s_k)$$

PR	C:	0	1
a_1:	1	$0 + 1 \cdot \underset{0}{F(0)} = \boxed{0}$	$0 + (1) \cdot \underset{0,7}{F(1)} = \boxed{0{,}7}$
a_2:	2	$0 + 0{,}7 \cdot \underset{0,7}{F(?)} = \boxed{0{,}49}$	$0 + 0{,}7 \cdot \underset{0,8}{F(2)} = \boxed{0{,}56}$
⋮			

---

posledná hodnota (tag) $T(a_{s_1} \dots a_{s_n})$

$$T(\dots) = \frac{u_n - l_n}{2}$$

dekódovanie: $t_k = \dfrac{tag - l_{k-1}}{u_{k-1} - l_{k-1}}$

$F(s_k) \leq \dots \leq F(s_k + 1)$

hľadáme $a_{s_1} \dots a_{s_k}$

finty:

$$t_{k+1} = \frac{tag - \overbrace{(l_{k-1} + (u_{k-1} - l_{k-1}) \cdot F(s_k - 1))}^{l_k}}{u_k - l_k}$$

uvážime

$$u_k - l_k = (u_{k-1} - l_{k-1})\, P(a_{s_k})$$

$$t_{k+1} = \frac{t_k - F(s_k - 1)}{F(a_{s_k})}$$

PR dekódovanie 0,55

$t_1 = \dfrac{0{,}55 - 0}{1} = 0{,}55 = tag \in \langle 0; 0{,}7) \Rightarrow a_1$

$t_2 = \dfrac{0{,}55 - 0}{0{,}7} = 0{,}786 \in \langle 0{,}7; 0{,}8) \Rightarrow a_2$

$u_k - l_k = (u_{k-1} - l_{k-1})\, P(a_{s_k})$
$= (u_{k-2} - l_{k-2})\, P(a_{s_{k-1}})\, P(a_{s_k})$
$= \prod_{i=1}^{n} P(a_{s_i})$

$l_n = \sum_{j=1}^{n} \prod_{i=1}^{j-1} P(a_{s_i})\, F(s_j - 1)$

$u_n =$

$a_1\ a_2\ a_3\ a_1$

$+0{,}7($

---

3. $\boxed{IV.}$ (v Huff $\leq H(A)+1$)

platí $H(A) \leq L(A) \leq H(A)+2$ pre binár. aritm. kód.
ak každému 1 znaku pridelím kód
ak 2 znakové a viacznak. kódy
$H(A^m) \leq L(A^m) \leq H(A^m)+2$
$m \cdot H(A) \leq L(A^m) \leq m \cdot H(A) + \frac{2}{m}$

čím viac znakov, tým bližšie k entropii
Huffmanovi sa mi ale rozširuje a komplikuje sa strom

je problém: treba mi presné aritmetiku (viac m-tíc .. väčšia presnosť)

ďalšie finty pre binárne aritmetic. kódovanie

0 0,10000 1

ak sú obe hranice vľavo, prvý bit bude 1, môžem poslať 1 a SHL
ak obe hranice vpravo, pošlem 0 a SHR

$\langle 0; 0{,}5)$: urob SHR (vynásob $2\times$)
$\langle 0{,}5; 1)$: SHR $(2(x-0{,}5))$

3	2
0,18	0,02

chcem kód 1 3 2 1

$l_0 = 0$ $u_0 = 1$
$l_1 = 0$ $u_0 = 0{,}8$

$l_2 = 0{,}8 \cdot 0{,}82 = 0{,}656$ $u_2 = 0{,}8$ | už sú v $\langle 0{,}5; 0{,}8\rangle$ odošlem 1, SHL

upravený interval
$(0{,}656 - 0{,}5) \cdot 2 = l_2$
$l_2 = 0{,}312$ $u_2 = 0{,}6 = (0{,}8-0{,}5) \cdot 2$

prechádza 2/
$l_3 = 0{,}312 + (0{,}6 - 0{,}312)\, 0{,}8 = 0{,}5424$ $u_3 = 0{,}312 + 0{,}288\ 0{,}8 = 0{,}54816$

naše sme v hornej polovici | odošlem 1
upravený interval:
$l_3' = 0{,}0424 \cdot 2 = 0{,}0848$ $u_3' = 0{,}09632$

naše je to v dolnej polovici | odošlem 0
$l_3'' = 0{,}1696$ $u_3'' = 0{,}19264$

naše | odošlem 0
$l_3''' = 0{,}3392$ $u_3''' = 0{,}38528$

a naše | odošlem 0
$l_3^{(4)} = 0{,}6784$ $u_3^{(4)} = 0{,}77056$

a naše | odošlem 1
$l_3^{(5)} = 0{,}3568$ $u_3^{(5)} = 0{,}5412$

prechádza 1:
$l_4 = 0{,}3568 + (0{,}5412 - 0{,}3568) \cdot 0 = 0{,}3568$ $u_4 = 0{,}3568 + 0{,}1844 \cdot 0{,}8 = 0{,}504256$

končíme, vezmeme číslo $\langle 0{,}3568; 0{,}504256)$
napr.: $\underbrace{10000}_{\text{0 do vyjdenia na koniec}}$

$\underbrace{110|011|0}_{\text{pred}}$ } tag

---

dekódovanie: 0 0,8 0,82 1

$T > 0{,}8$ → 1 $1 \in \Sigma_S$
$l_1 = 0$ $u_1 = 0{,}8$
$l_2 = 0{,}8 \cdot 0{,}82 = 0{,}656$ $u_2 = 0{,}8 \cdot 1 = 0{,}8$
$l_2' = 0{,}312$ $u_2' = 0{,}6$
musím shiftnúť aj vstup. $T_2 = 0{,}10011$
$l_3 = 0{,}312 + 0{,}288\ 0{,}8$ $u_3 =$

celočíselné aritmetické kódovanie

majme m-bitové čísla: je ich 2^m

00...00 | 0   prvý rozsah m-bitov

11...11 | 1

nám Σ_S a pravdepodob. rozdelenie

$\Sigma_S = \{a_1, \dots, a_k\}$ n-tica znakov zo vstupu abecedy

n_i ... počet výskytov a_i v danej n-tici

$n = \sum_{i=1}^{k} n_i$. Máme ešte distribučnú funkciu $D(s) = \sum_{i=1}^{s} n_i$ $D(k) = n$

chcem len nad celočísel. aritmetikou

$l_j = l_{j-1} + \frac{1}{n}\left((u_{j-1} - l_{j-1}) + 1) \cdot D(x_j - 1)\right)$ znak pred x_j v n-tici

$x_1 \dots x_m$ nech je jedna n-tica

$x_i \in \{a_1, \dots, a_n\}$

$u_j = \lfloor \quad '' \quad \cdot D(x_j) \rfloor + 1$

aby sa to dalo implementovať, tak by ste to museli zistiť to

ak sa písmeno nevyskytuje, tak ho nerozrátavam.

aby sa to dalo nakódovať, musí platiť

$2^m \cdot \frac{n_{min}}{n} > 4$ $2^m > \frac{4n}{n_{min}}$ $m > \log_2 \frac{4n}{n_{min}}$

$\boxed{R}$

(1) = 40   potrebujem aspoň 200 čísel v rozsahu

(2) = 1   teda stačí 8 bitov

(3) = 9 $m > \log_2 \frac{200}{1}$ $m \doteq 8$ bitov

n = 50

aritmetický vs Huffman

- netreba posielať stromy

výhodnejší, ak je nerovnomerné rozdelenie

ale vzhľadom presnosť aritmetiky, je to lepšie

odoramme texty lepšie využiť usporiadanie v zdroji

ako frekvenčnú analýzu písmen

ako by som kódoval po digramoch

slovníková kompresia

- vytvára sa slovník

- statický kódujem písmená, alebo digramy

napr. kódujem písmená, voľné kódy slova používam

napr. na digramy   (26 b na A–Z

26–32 na digramy)

digramové a trigramové

prechádzam, hľadám trigramy, potom digramy, písmená

napr. kľúčové slová v program. jazyku

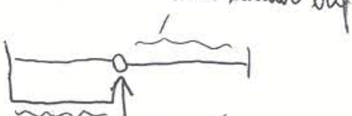
- môžem urobiť frekvenčnú analýzu digramov a trigramov

ale musím ju posielať

- dynamický slovník: vytvára sa priebežne, mení sa (adaptívne)

pravidlá také, aby sa to na príjme zrekonštruovalo

LEMPEL – ZIV (LZ77)



dívam sa na kus textu

všetko čo je pred hlavou považujem za slovník

chcem pomocou slovníka zakódovať, čo nasleduje

- ak sa písmeno nachádza v slovníku, ~~[illegible]~~

- ak sa nenachádza   zistím, či aj nasledovné

pošlem kód <0, 0, kód písmena>   písmená mi nevytvoria

niečo v slovníku

vezmem najdlhší podreťazec LABuffera a pošlem

<offset, dĺžka, kód nasled. písmena>

$\boxed{PR}$

CABRACA|DABRA|R|RARRAD

7 znakový SB ...   6 znak LAB

<0, 0, kód (C)>
<0, 0, (A)>
<0, 0 (B)>
<0, 0, (R)>

D: nie je ⇒ <0, 0, kód(D)>, posuniem okno

A: v SB mám najdlhšie "ABRA"

offset = 7, dĺžka = 4   <7, 4, kód(R)>

idem kódovať druhé R

môže sa SB prekrývať v LAB

RARRA   <3, 5, kód(D)>

koniec

nevýhoda: mnoho kódov typu <0, 0, ?>

ulelo rozlišujúci 1 bitom: 0 .. kód písmena,
1 .. ale offset, dĺžka

ulelo: sa najprv kód ďalšieho znaku (LZSS)

LZ78

vytvára sa neraviadny slovník (v LZ77 strácam už spracované údaje mimo SB)

WABBA ₿WABBA₿WABBA₿WABBA₿WABBA WOO₿ WOO₿

na začiatku: W: ⟨0, kód (W)⟩ slovník: 1→W

A: ⟨0, kód(A)⟩ 1→W 2→A

B: ⟨0, kód B⟩ 1→W 2→A 3→B

B: ⟨3, kód (A)⟩ 1→W 2→A 3→B 4→BA

následujúce písmeno

₿: ⟨0, kód (₿)⟩ —11— 5→₿

W: ⟨1, kód (A)⟩ —11— 6→WA

B: ⟨3, kód(B)⟩ —11— 7→BB

A: ⟨2, kód (₿)⟩ —11— 8→A₿

WA: ⟨6, kód (B)⟩ —11— 9→WAB

3A: ⟨B, kód (₿)⟩ —11— 10→BA₿

: ⟨9 B⟩ —11— 11→WABB

: ⟨8, kód (W)⟩ —11— 12→A₿W

:

toto je kód

ekódovanie: beriem a budujem slovník

ovanie ale dvojíc písmen (v slovníku aj na výstupe)

LZW

ohodnú sa obe strany na abecede {₿, A, B, O, W} (1 2 3 4 5)

bude posielať len čísla, namiesto kódov

W: 5 do slovníka o písmeno viac 6→WA

A: 2 7→AB (ešte AB v slovníku nie je)

B: 3 8→BB

B: 3 9→BA

A: 2 10→A₿

₿ 1 11→₿W

WA 6 12→WAB

BB 8 13→BBA

A₿ 10 14→A₿W

na každom znaku musím niečo dať do slovníka

dekódovanie

5 2 3 3 2 1 6 8 10

₿ A B O W
1 2 3 4 5

5→W

2→A | v slovníku 6→WA

najprv zapíšem potom výstup

3→B | 7→AB

3→B | 8→BB

2→A | 9→A

1→₿ | 10→A₿

6→WA | 11→₿W

8→BB | 12→WAB

10→AB | 13→BBA

| 14→A₿

15.3.2004 V: VSETCI/GRAFIKA/COMPRESS

ZIP: 10 metód (pre každý súbor zvlášť)

0.. stored

1.. shrinking (LZW) bajty: priamo 9 bitov: 256|1

plním tabuľku, ak je plná, príde špec. znak a tabuľka sa zväčší na 10b

ak nestačí, opäť rozšírim.

max: 13 bitov

o špec. znak redukcie odrež všetky listy v strome, na nich sa môžu pridávať nové sekvencie (redukcia je nekonštruktívna) 256|2

2–5: Huffmann (rastie frekvencie znakov pre pravdep.

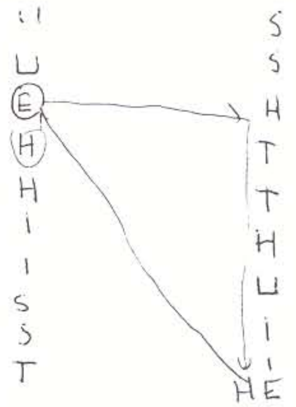
6: imploding: modifik. LZ77, na konci binárny + Shannon. strom

7: deflating: sliding directory (32KB) + Huffman (expanding modif.)

THIS␣IS␣THE␣

0. rád	
t	1
h	1
i	2
s	1
␣	1

1. rád		
t	h	1
h	i	1
i	s	1
s	␣	1
␣	i	1



porovnám si kontext s najlepšou pravdepodobnosťou

s: v 0: 1/5   v 1: 1/6   v −1: 1/26

vyberiem teda 0. rád a upravím kontexty

kontexty sa menia priebežne (rovnako pri dekód.)

potom: aritmetic. kódovanie podľa najlepš. kontextu

je to ako samotné aritm. (blíži sa k slovník. technikám)

do pravdepodobné dvojice sa môžu vyhodiť podľa nejakých pravidiel

---

TRANSFORMÁCIA VSTUPU + KÓDOV. ALG.

(preusporiada znaky na vstupe)   BURROWS-WHEELER

BWT: prehodí znaky tak, aby ich bolo veľa rovnakých pri sebe.

(BZ2: robili ste s tým niečo?)
BZ2 (na BZ2 sme mali analýzu.")

THIS␣IS␣THE — urobí sa SHL (všetky rotácie)
ETHIS␣IS␣TH — urobí sa lexikografické
HETHIS␣IS␣T — usporiadanie
THIS␣IS␣
THIS␣IS — posled. stĺpec má dobré štatistic.
THIS␣I — vlastnosti (zvíce písmen sú
THIS␣ — častejšie pri sebe)
THIS — na dlhom texte časté sekvencie rovnakých písmen.

potom posledný stĺpec a číslo riadku s pôvodn. textom

dekódovanie: dostanem stĺpec a riadok. vezmem stĺpec a usporiadam písmená podľa abecedy do 1. stĺpca textu

na to nahodíme RLE

S2 H1 T2 H1 ␣1 I2 ␣1 E1

alebo MTF (move-to-front)

písmená zoradené podľa abecedy ␣EHIST
priradím čísla 0 1 2 3 4 5

S=4 | prehodím dopredu   S␣EHIT (0 1 2 3 4 5)
S=0 | ďalšie
H=3   HS␣EIT
T=5   THS␣EI

často sa vyskytujúce písmená majú nízke čísla

403 ~~5~~ 5 0 1 3 5 0 1

FAX   CCITT GROUP 1 – 4
jednoduch. zložité

1. špeciál. RLE: strieda počet bielych a čiernych
1-D kódovanie políčok (na zač. je čierna)

2- ... nejaké predikcie

3-4. predikcia v závislosti od predch. riadku (rozdiely od predch. riadku)

JPEG: bezstratový typ (JPEG-LS)   „do bloku"

predikcia typ 0 – 7   vyberie sa najlepšia, kód sa zapíše do hlavičky

0. bez predikcie
1. z predch. pixelu
2. z vyššieho pixelu
3. z [i−1, j−1]
4. $\begin{array}{|c|c|}\hline c & a \\ \hline b & \\ \hline\end{array}$ (a+b)−c resp. (N+W)−NW
5) W+ priemer (N+NW)
6. priemer (W+NW)+N
7.) priemer (N+W)

v JPEG 2000 navyše PPM

možno aj progresívne

GIF (LZW)

často sú medzi bajtami súboru nejaké závislosti
(vi slovníkové kódy to nejakým spôsobom využívajú)

5 5 4 2 1 7 6 5

viem predpovedať (napr. ak dáta sú nejaký signál)

+2 −1 −2 −1 +6 −1 −1 | zmeny medzi jednotlivými hodnotami

menšie rozdiely hodnôt (rozptyl)

PREDIKČNÉ METÓDY KOMPRIMÁCIE $y_1, \dots, y_n, \dots$

$\overline{y_n} = y_{n-1}$ (rovnaké znaky)

$x_n = y_n - \overline{y_n}$ viem pôvodnú postupnosť rekonštruovať

Predikcia len z už spracovaných hodnôt

$y_n = x_n + \overline{y_n}$

kódovanie: malé čísla kratšími reťazcami
veľké dlhšími reťazcami

problém: + / − : riešenie: na koniec pridám 0 alebo 1 (RISE - GOLOMB)

kódovanie čísla ako dvojica $qm + r$: <m, r>

q 0 r, q .. počet jednotiek udáva q zvyšok

$q = 3$ $\underbrace{111}_{q}$

pre $m = 5$

4 ↦ 01111 ($q = 0$)
3 ↦ 110
2 ↦ 10
1 ↦ 01
0 ↦ 00

„na 2 bity to asi ako nerozhodujem, tak skúsim teda na 2"

ako v adaptívnom Huffmanovom kódovaní)

12 ↦ 11 0 10

2 ako podiel
2 ako zvyšok (počet jednotiek)

optimálne ak $P(n) = p^{n-1}(1-p)$

potom je vhodné $m := \left\lceil \frac{-1}{\log_2 p} \right\rceil$

Kódovanie ČB obrázkov (binárne)

JBIG (joint bilevel processing group) ⊆ ISO

berie sa okolie bodu 

10 okolitých bodov určí kódy: tabuľka 1024 riadkov

určíme pravdepodobnosť výskytu 1 (prechodom celého obrázka)

potom aritmet. kódovanie: pre celý obrázok budeme jeden kód: pre p_i delím interval podľa kontextu.

| p_i | $1-p_i$ |

príklad pre 3: 

000	0,2
001	0,8
010	0,5
011	0,4
100	0,3
101	0,9
110	0,1
111	0,2

0 0 1 0 1
0 1 1 0 1
0 1 0 1 0

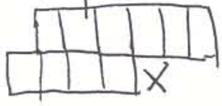
mám hranice L H
L =
kontext: (0, 1, 0)
„čítam po riadkoch

0 0 1 0 1
0 1 1 0 1
0 1 0 1 0

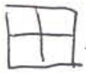
kontext (1, 0, 1)
P("1") = 0,9
rozdelím interval a pre 0 beriem dolnú hranicu

na začiatku a mimo obrázku sú nuly

mám rozdelenia pravdepodobnosti pre kontexty.

2-riadkovo 

PROGRESÍVNE KÓDOVANIE

- pre JBIG sa berú štvorce . V 1. priblížení sa berie 1 hodnota pre 4 štvorce, ďalej sa berú rozdiely v štvorcoch od veľkého štvorca. Obrázok sa „zaostruje".

PPM (prediction with partial match)

často v kombinácii s LZW

„pri výskyte LZW zlyhaní"

môže sa stať, že je lepšie zakódovať viacero kratších po sebe idúcich rovnakých reťazcov než hľadať najdlhší reťazec

berú sa výskyty v kontextoch

(−1) - rád ... rovnomerné rozdelenie (znak sa nevyskytol)
0 ... výskyt jednotl. znakov

1. 3 2004 VI

teraz-
kódér | ——— kanál ——— | dekóder |

Σ_S Σ_C

zavádzali sme chyby

uvažujme teraz chyby pri prenose (šum)

ak $W = \{w_1, \dots, w_m\} \subseteq \Sigma_C^*$

$a_1 \mapsto w_1, a_2 \mapsto w_2, \dots, a_m \mapsto w_m$

úplný kód: každé slovo v kanál abecede zodpovedá nejakému slovu v Σ_S

vieme rozpoznávať čo je slovo (kód) a čo chyba v Σ_C

kódové slová: $W' = \Sigma_C - W$ └ nekód. slová

chybami/šumom z kód. slov sa stávajú nekódové a naopak

odhaľovanie chyby

odstraňovanie chýb

nejaké predpoklady (najviac k chýb ...)

blok. kódmi: rovnaké kód. slová: takto sa pracuje

$l \subseteq \Sigma^k$ slová dĺžky k

$|\Sigma^k| = n^k$ slov dĺžky k

hammingova vzdialenosť na reťazcoch

$\Delta(v,w)$ $v = b_1 \dots b_s$, $w = b'_1 \dots b'_s$ | $d(v,w)$ = počet indexov i, kde $b_i \neq b'_i$, $\forall i \in [1, \dots s]$

$\Delta(00000, 11111) = 5$

. je metrika

D = minimálna vzdialenosť blok. kódu W

$= \min\{d(v,w) : v, w \in W, v \neq w\}$

úplnom kóde $D = 1$

$D = 2$ 2 kód. slová sa líšia aspoň na 2 miestach

odhaluje chybu pri prenose

$\underbrace{b_1 \dots b_s}_{b} \xrightarrow{\text{chyba}} \underbrace{b_1 \dots b_i \dots b_s}_{b'}$ ↑ chyba $d(b,b') = 1$ potom ale b' nemôže byť kód. slovo

k D nejakého kódu vie odhaľovať D-1 chýb

$\Sigma_S = \{a,b,c,d\}$, kód opravujúci 3 chyby (opakovacie kódovanie)

a ↦ 0000 0000
b ↦ 0000 1111
c ↦ 1111 0000
d ↦ 1111 1111

vychádzam z úplného kódovania, rozšírim paritný-počet-kód

dvojnásobný ?

odhalenie 1 chyby

			počet 1-tiek	
a ↦ 0000	celkom	00	0	párny počet 1-čiek
b ↦ 0011	pridám	01	1	D=2
c ↦ 1100	paritný	10	1	
d ↦ 1111	bit	11	0	

ISBN: 10 cifier
0-13-061814-4
$a_1 \dots a_9$
$\sum_{i=0}^{10} i a_i \equiv 0 \pmod{11}$

kód „2 z 5" opravuje 1 chybu

a 11000
b 00011
c 01100
d 00110
e 10100
f 10010
g 10001
h 01010
i 01001
j 00101

posled. cifra: 0.. 9, X
• detekuje zámenu 2 sused. čísel
• detekuje 1 chybu

nech $i \cdot x + (i+1)y = (i+1)x + iy \pmod{11}$

$y = x \pmod{11}$

$y = x + 11t$, ale $x, y \in \{0, \dots 9, X\}$

EAN-13 (čiar. kód)

$a_{13} = -(a_1 + 3a_2 + a_3 + 3a_4 + \dots + a_{11} + 3a_{12}) \bmod 10$

8588001 028817

~~5+24+8+0+0+3+0+6~~
~~8+24+1+24~~
~~= 29+8+3+0+14+46~~
~~= 37+3+60=100 mod~~

• odhaluje jednoduché chyby
• odhaluje zámeny niekt. chýb

2rozmerné kódy: odhaľujú 2 chyby

SAMOOPRAVNÉ CHYBY

- vie odhaliť a opraviť chybu
- ku každému kód. slovu nemám slovo so vzdialenosťou 1

- samoopravný kód opravuje t-násobné chyby, ak vzdialenosť $(v,w) \le t$ (pošlem v, prijímam w) a $\exists u \in W: d(u,w) < d(v,w)$

kontrolný odhaľuje, ale neopravuje 1 chybu, ale vie odstrániť 1 chybu

a ↦ 000 111
b ↦ 000 000
c ↦ 111 000
d ↦ 111 111

ak D je min. vzdialenosť dvoch kódov W, tak kód

vie opraviť $\left\lfloor \frac{D-1}{2} \right\rfloor$ chýb

„tam by som tím dostal, hej ako nejako"

pre opravu 1 chyby:

vieme $2^k \geq k \cdot |W|$

ak $|V| = \frac{2^k}{k}$ perfekt. kódy

ak tu nejaký vzťah je neviem,

čo je čo, $\frac{\log_m |W|}{k}$ .. redundancia

kód: informač. časť | kontrol. časť

napr. (6, 2) kód — inf. — kontrol

kód (7, 4)

0000
0010
0011
0100
0101
0110
0111

príde $\overset{b_1 \dots}{0010110}{}^{b_7}$ .. bity

chcem nájsť chybný bit

8 výsledkov (na ktor. mieste, alebo nikde)

$$\left.\begin{array}{ll} b_1 \oplus b_3 \oplus b_5 \oplus b_7 = S_0 & (1\text{-ice}) \\ b_2 \oplus b_3 \oplus b_6 \oplus b_7 = S_1 & (2\text{-ice}) \\ b_4 \oplus b_5 \oplus b_6 \oplus b_7 = S_2 & (4\text{-ice}) \end{array}\right\} (\square)$$

viem, že ak $S_0 = S_1 = S_2 = 0$, potom je to kód. slovo

S_0 určí pár/nepár, S_1 druhý, S_2 stredný

týmito 3 rovnicami to viem odkontrolovať.

konštruujme taký kód:

$b_1 = b_3 \oplus b_5 \oplus b_7$ .. platí S_0

$b_2 = b_3 \oplus b_6 \oplus b_7$ .. platí S_2

$b_4 = b_5 \oplus b_6 \oplus b_7$ .. platí S_3

0000	
$b_3 b_5 b_6 b_7$	0000000
0001	1101001
1101	1010101
111	1111111

majme kód. slovo 1010101 : chybné bude : 1011101

rátame súčty podľa $(\square)$

S_0: 0

S_1: 0

S_2: 1

$100 = (4)_2$ .. na 4 mieste

malo 1101001, chyba 1111001

$S_0 = 1 \quad S_1 = 1 \quad S_2 = 0 \quad \} \Rightarrow 011 = (3)_2$.

chyba na 3. mieste

perfektný kód pre 1 chybu -

Hammingove kódy. $(2^m - 1, 2^m - 1 - m)$

$$(a_1, a_2, a_3, a_4)\left(E_4 \cdot \begin{matrix} 110 \\ 101 \\ 011 \\ 111 \end{matrix}\right) = \begin{pmatrix} a_1' \\ a_2' \\ a_3' \\ a_4' \\ \vdots \\ a_7' \end{pmatrix}$$

4-1 stĺpcov

dekódovanie

$$\underbrace{(\dots)}_{\text{sednica}} \begin{pmatrix} 110 \\ 101 \\ 011 \\ 111 \\ \hline 100 \\ 010 \\ 001 \end{pmatrix} = (\dots)$$

~~pozadovaba~~

ak je to OK, je tu nul. vektor.

tu musí vyjsť binár. zap. miesta chyby

GOLAYOV KÓD (23, 12) 3-perfektný

HAMMINGOV KÓD 1-perfektný

neexistujú iné perfektné kódy

29. 3. 2004 VII.

STRATOVÁ KOMPRESIA

LOSSY compression

→ [encoder] — kanál — [decoder]

Σ_S Σ_c

$a \in \Sigma_S^* \rightsquigarrow b \in \Sigma_c^*$

namiesto b verim b'

dekódovanie: $b' \rightsquigarrow a'$

„je pokazený signál", kanál je spoľahlivý!

ten to kazí. chyba pri prenose $d(a, a')$. strata

d: distortion napríklad d ako kvadrat. u

$d(a, a') = (a - a')^2$ $\qquad \sigma_x = \frac{1}{N} \sum_{i=1}^{N} x_i^2$

$d(a, a') = |a - a'|$

$d(a, a') = a \oplus a'$ (ak a, a' sú vektory)

stredná kvadrat. chyba $\frac{1}{n} \sum_{i=1}^{n} (x_i - x_i') = \sigma^2$

miera kvality signálu

charakteristika signálu: pomer signálu k šumu

$\frac{\sigma_x^2}{\sigma^2} = SNR$ (signal noise ratio)

pomery sú dosť veľké čísla, meriame v dB

$10 \log \frac{\sigma_x^2}{}$

- môžem merať aj ďalšie „nie pomalé" pohyby

• oko : intenzita signálu $1 - 10^{10}$

vo sa rozlišovať len na rovnakých intenzitách
(ostré svetlo, šero, ...)
intenzívne farby s ... ,
nedaj ... a ...
chyby v pozadí

stred intenzita   intenzita

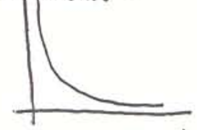
často sa vyjadrujú (...posuny ... na „slovar")

• ucho: 20 – 20000 Hz, intenzita (vnímaná citlivosť na rozsahu)

vnímaná frekvencia (menej strát vo vyšších frekvenciách)

distortion

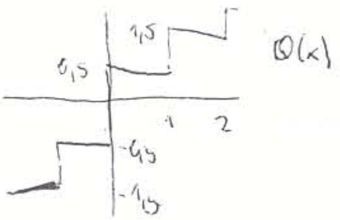
RD-function



kompres. pomer

$H \cdot \log(1 + SNR)$

„to sú nejaké integrály to by som tam aj tak nerozumel, to by som musel 2 dni vysvetľovať a by ste si to aj tak len odpísali... alebo na DÚ"



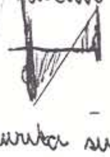
predp. rovnomerne rozdelíme hodnoty v $[-x_{max}, x_{max}]$

$\Delta = \frac{2x_{max}}{M}$ M úrovní kvantizácie

M intervalov

$\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$ $\sigma_q^2 = \frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} q^2\, dq = \ldots = \frac{\Delta^3}{12\Delta} = \frac{\Delta^2}{12}$

chcem rátať plochu pod krivkou



intenzita šumu

„tie integrály sa dá na mieste z rôznych strán ako dívať"

... $M = 2^n$ (počet bitov na úroveň)

$SNR\,[dB] = 10 \log \frac{\sigma^2}{(\sigma_q)^2} = 10 \log \left(\frac{2x_{max}}{2x_{max}}\right)^2 \cdot M^2$ „potom ďalej už to viem"

toto by som chápal

$\sigma^2 = \frac{(2x_{max})^2}{12}$ ale tomuto už nie

$= 10 \log M^2$

$\to 10.2 \log M$

pre σ^2 rátame integrál („integrál od rovnakého až do konca")

= n krát nejaká konštanta

= n · 6dB   pre odstup 30 dB potrebujem 5b

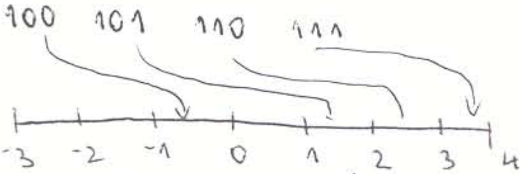
6dB zodpovedá 1b pri kvantizácii

kvantizačný šum $x - Q(x)$ vzniká pri kvantovaní

telefón : 30 dB odstup   CD . 16b

KVANTIZÁCIA ako metóda stratovej kompresie

Δ konšt. uniformná

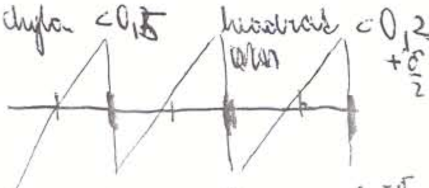


namerané hodnoty popíšem intervalom, v ktorom sa nachádza

000 .. najľavo   111 .. najpravo (mnoho intervalov)

−1,23	−0,13	1,18	2,22	Σ_g
010	011	101	110	
−1,5	−0,5	1,5	2,5	Σ_c
0,27	0,37	0,32	0,28	chyba

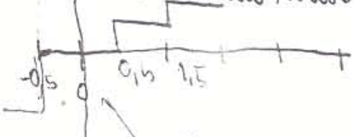
chyba < 0,5   hodnota < 0,25 $+\frac{\delta}{2}$



kvantizačná chyba funkcia

$q = x - Q(x)$

čo s 0? treba robiť buď 0,5 / −0,5



midrise quantizer
v 0 skáče

midtread quantizer
v 0 prechádza

ADAPTÍVNA (UNIFORMNÁ) KVANTIZÁCIA (FORWARD (OFFLINE))

- prejdem vstup a podľa neho nastavím intervaly

• zistím x_{min}, x_{max} a rovnomerne rozdelím

$\Delta = \frac{x_{max} - x_{min}}{M}$

„tiež sa tam dá niečo spraviť"

backward (online)

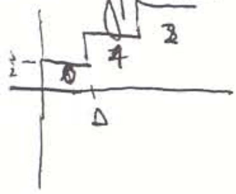
zdá na nastaviť adresné nastavujem intervaly

JAYANT kvantizácia

$\Delta_n = \Delta_{n-1} \cdot M(Q, n-1)$

k rovnosti od kvantovania v predch. cas

aj problémy, keď možno škála a zistím, že to zdá do nejakej časti ($\Rightarrow$ interval sa zmenší) alebo vypadáva ($\Rightarrow$ interval sa zväčší)



$M_0 = M_4 = 0{,}8$ .. zmenšuje sa Δ (priemerujeme)

$M_1 = M_5 = 0{,}9$

$M_2 = M_6 = 1$

$M_3 = M_7 = 1{,}2$

raz sa to zmenší, že väčší.

musia byť rovnaké hodnoty aj dĺžku

a nastavím prvé Δ

NEUNIFORMNÁ KVANTIZÁCIA (nerovnomerná kvant.)

výhody: menšie intervaly blízko stred. hodnoty

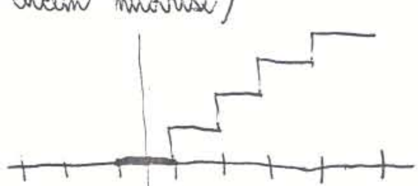
väčšie intervaly ďalej

PDF-kvantizácia (PDF = distrib. fcia)

"... ale ako ste to povedal?"

LLOYD-MAX algoritmus

- poznám distrib. fciu ("mne sa zdá, že to je dosť ťažké.") chcem minimalizovať)



$[b_2, b_3] = y_3$

$[b_1, b_2] = y_2$

$b_0 = 0$, M známe, $b_{\frac{M}{2}} = \max = \infty$

$$y_1 = \frac{\int_{b_0}^{b_1} x \, f_x(x)\, dx}{\int_{b_0}^{b_1} f_x(x)\, dx}$$

"nikdy nevieme vypočítať tento integrál"

b_1 nepoznám...

dist. fcia na vstupe

okolo stredu je to veľa.

$$b_j = \frac{y_{j+1} + y_j}{2}$$

"ale y_{j+1} nepoznáme..."

"to je ČO?"

$$y_j = \frac{\int_{b_{j-1}}^{b_j} x\, f(x)\, dx}{\int_{b_{j-1}}^{b_j} f_x(x)\, dx}$$

"nie čo to vlastne hovorí"

"minimalizuje sa preto čo sa hodí tak najviac"

$$\sigma_q^2 = \sum_{i=1}^{M} \int_{b_{i-1}}^{b_i} (x - y_i)^2 f_x(x)\, dx$$

reálna hod. kvantovaná hodnota

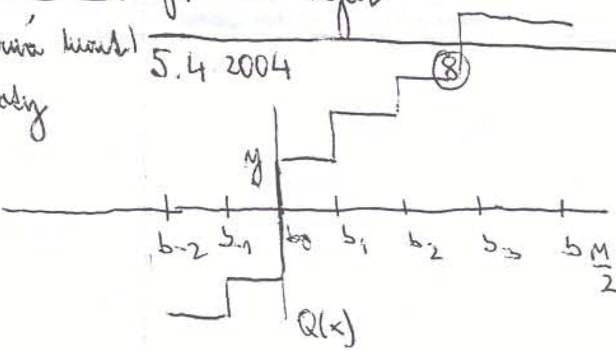
chceme minimalizovať šum σ_q^2, takú kvantizáciu chcem postaviť

"to odhadnete na DÚ nejakým lineárnym prediktorom (dajte to do série).."

$$b_1 = \frac{y_2 + y_1}{2} \Rightarrow y_2 = 2b_1 - y_1$$

y_1 zvolím nejak

5.4.2004 ⑧ VIII.



chceme nájsť kvantizačný šum (diskrét. prípad)

$$\sum_{i=-\frac{M}{2}}^{+\frac{M}{2}} \int_{b_i}^{b_{i+1}} (x - y_i)^2 \cdot f_x(x)\, dx \longrightarrow \min$$

derivácia podľa y_i, porovnáme s 0

$$y_i = \frac{\int_{b_{i-1}}^{b_i} x\, f_x(x)\, dx}{\int_{b_{i-1}}^{b_i} f_x(x)\, dx}$$

derivácia podľa b_i, porovnáme s 0

$$b_i = \frac{y_i + y_{i+1}}{2}$$

Kde bitka musí byť v strede y-niek, potom bude šum minimalizovaný ako nastaviť hranice, keď máme danú dĺžku poľa ?

LLOYD-MAX

$b_0 = 0$

y_1 = náhodne/odhadnem, potom viem vypočítať ostatné hodnoty

z optimál. vzťahu   keď DF => unif. kvant.

$$y_1 = \frac{\int_{b_0}^{b_1} x f_x(x)\,dx}{\int_{b_0}^{b_1} f_x(x)\,dx}$$, viem vyrátať b_1

$y_2 = 2b_1 - y_1$

b_2 zo vzťahu pre y_2

$\boxed{y_{\frac{M}{2}} \quad b_{\frac{M}{2}}}$

$\boxed{= 2b_{\frac{M}{2}-1} + y_{\frac{M}{2}-1}}$

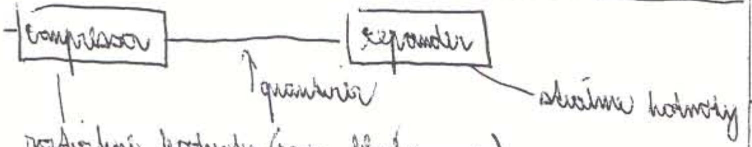
zároveň by malo platiť

$$\boxed{y_{\frac{M}{2}} = \frac{\int_{b_{\frac{M}{2}-1}}^{b_{\frac{M}{2}}} x f_x(x)\,dx}{\int f_x(x)\,dx}}$$

$b_{\frac{M}{2}}$ poznám (je maximálna hodnota, ktorú máme)

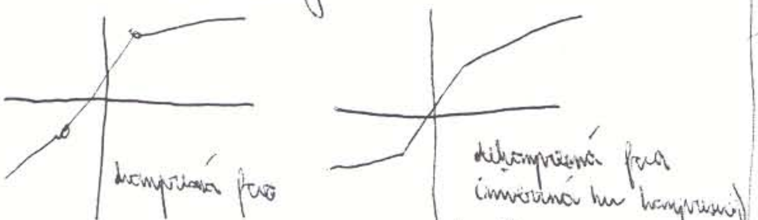
porovnáme, ak rozdiel $< \varepsilon$ .. OK

inak musím upraviť kvantizáciu



rozdielne hodnoty (napr. blízke nule)

rozšíri iné hodnoty

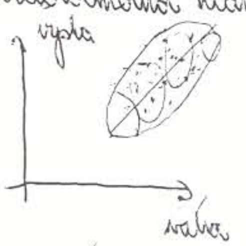


$Q(C(x)) \longrightarrow D(\tilde{Q}(2)) = x$

vektorová skalárna kvantizácia

VEKTOROVÁ KVANTIZÁCIA

viacrozmerná kvantizácia pre $f: X^n \to \mathbb{R}$



pri scalar kvantizácii výšky a váhy by sa nejaké extrémne hodnoty stratili

vytvoríme oblasti (klastre) oblasť má byť podľa

$Q(X) = Y_i$ ak $d(X_i, Y_i) \le d(X_i, Y_j) \quad \forall j \ne i$

najbližší reprezentant

kvantizačný región = „oblasť"

$V_i = \{X : \forall j \ne i \;\; d(X, Y_i) < d(X, Y_j)\}$

LBG kvantizácia (LINDE-BUZO-GREY) clustering

postupne sa približujeme $D^{(0)} = 0$

$Y_i^{(0)}$ $i \in [1, m]$ volíme ; ε.. presnosť ktorú chceme dosiahnuť

$V_i^{(0)} = \{X : d(X, Y_i) < d(X, Y_j), j \ne i\}$

$$D^{(k)} = \sum_{i=1}^{M} \int_{V_i^{(k)}} \| X - Y_i^{(k)} \|^2 \cdot f(x)\,dx$$

DF

výpočet kvant. chyby

ak $\frac{D^{(k)} - D^{(k-1)}}{D^{(k)}} < \varepsilon$ .. koniec

inak vyrátam nové hodnoty

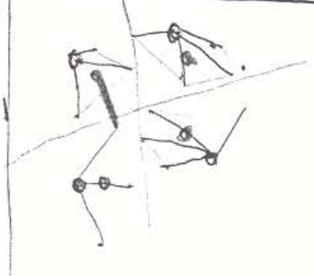
$k := k+1$

$Y_i^{(k)}$ = stredná hodnota z $V_i^{(k-1)}$

opäť vytvorím nové regióny a porovnám

problém: zložité výpočty integrálov

riešime - porovnáme s tréning. množinou.



spočítam čo chcem.

ak nie som spokojný

vyrátam stredné hodnoty

(„vektor priemery")

pri každom kroku by sa malo D minimalizovať

Diferenčné kódovanie (differential encoding)

x_1	x_2	x_3					
6,2	9,7	13,2	5,9	8	7,4	4,2	1,8
6,2	3,5	3,5	-7,3	2,1	-0,6	-3,2	-2,4
d_1	d_2	d_3					

$d_i = x_i - x_{i-1}$

k by sme kvantizovali na $-6, -4, 2, 0, 2, 4, 6$,

predikt by bol 6 | 4 | 4 | -6 | 2 | 0 | -4 | -2

d_1' | d_2' | d_3' | ...

dekódovanie

10 14 8 10 10 6 4

$x_i' = x_i - x_{i-1} + \delta_i$

$x_{i+1}' = x_{i+1} - x_i + \delta_{i+1}$

$x_i' = \sum_{i=0}^{i} d_g = x_i + \sum_{j=0}^{i} \delta_j$

chyba sa propaguje a hromadí, až je celá hodnota môže to vybehnúť úplne mimo

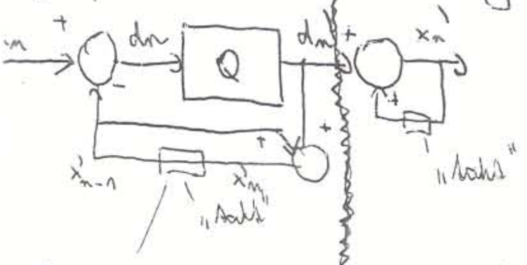
---

kompatibilné: diferencie rátame z kvantizovaných hodnôt

$d_i = x_i' - x_{i-1}'$

6,2	9,7	13,2	5,9	8	7,4	4,2	1,8
6,2	9,7-6	13,2-10	5,9-14	8-8	7,4-8	4,2-8	1,8-4
6	3,7	3,2	$\cancel{-8,1}$	0	-0,6	-3,8	-2,2
+	4	4	-6	0	0	-4	2
	10	14	8	8	8	4	2

-y neprekročia veľkosť kvantizačnej úrovne



môžem dať zložitejšie a nie komplikovanie

často lineárna fcia

odhadom $P_n = f(x_n', x_{n-1}', \ldots x_1')$

pravidelný/jednoduchý priebeh

jednoduchý prípad $f(x_n') = x_{n-1}$

DPCM - differential pulse code modulation (vzorec)

statický výpočet z vopred známej vzorky

musím poslať dopredu parametre predikčnej funkcie

---

delta na základe charakteristiky vstupného signálu

P: prenesť / dohodnúť

ADAPTÍVNE DPCM : ADPCM

- kvantizátor: forward (offline): dopredu spočítam hodnoty kvantizačnej fcie (minim chyby, optimalizované hodnoty)

backward (online) - Jayant

- prediktor: forward (ADPCM-APF)

„adaptívna predikcia forwardná"

často sa ráta forward po blokoch (nie z celého)

zvuk: 128 vzoriek (16 ms)

obraz: 8×8

viac vzoriek = viac parametrov (sústava 100×100, musím ju vyriešiť za 16 ms vyrátať + musím ju poslať)

dekódovanie je veľmi rýchle

- prediktor: backward (ADPCM-APB)

priebežne sa upravujú koeficienty polynómu predikč. fcie

je dobré vhodne navrhnúť z druhej strany

úroveň predikcie $x_n' = \sum_{i=1}^{k} a_i x_{n-i}$ k úroveň predikcie

1 úroveň predikcie $x_n' = A_n x_{n-1}'$ chcem A meniť dynamicky priebežne

najjednoduchšie $A_n = A_{n-1} + \alpha d_n x_{n-1}'$

(derivácia prirodzene)

predch. diferencia

$d_n = x_n - A x_{n-1}$

$d_n^2 = (x_n - A x_{n-1}) \Rightarrow \min$ (súčet štvorcov čo najmenší)

$A^{(n+1)} = A^{(n)} - \alpha \left(\frac{\partial d_n^2}{\partial A} \right) = -2 d_n x_{n-1}'$

z druhej strany viem $A_n = A_{n-1} \alpha d_n' x_{n-1}'$

$x_{n+1}' = A_{n+1} x_n$

$A_{n+1} = A_n + \alpha d_n' x_n'$ do ďalšieho kroku

meniť rýchlosť zmenou jemnosti kvantovania

DELTA MODULÁCIA!

kvantizátor: ± 1

ak sú prudké skoky, nestíham modulovať

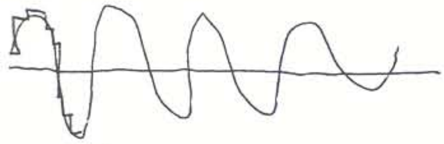
jednoduché kódovanie

stačí vzorkovať dvojnásobnou frekvenciou, tu ale

musím vzorkovať častejšie (100×)

rovný signál ——— na výstupe 1010101010.

ADAPTÍVNA Δ-MODULÁCIA (CFDM)



pohnúť iba o 2 kroky nahor / nadol, ďalší krok sa

zdvojnásobí

$M \in \langle 1,2 \rangle \quad \Delta_m = M \cdot \Delta_{m-1} \quad \begin{matrix} ++ \\ -- \end{matrix}$

$\Delta_m = \frac{1}{M} \cdot \Delta_{m-1} \quad \begin{matrix} -+ \\ +- \end{matrix}$

$sg_{M-2} = sg_{M-1} = sg_M \Rightarrow$ väčší koeficient $\Delta := \Delta * 2$

$sg_{M-2} \neq sg_{M-1} = sg_M$ 1,5

$=$ $\neq$ 0,94

$\neq$ $\neq$ 0,9

koeficienty / kvantizačné hodnoty: máme 2 typy

„to je všetko, čo som si nedávno prečítal"

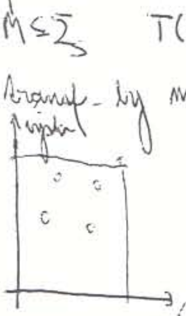
TRANSFORMÁCIA   26.4.2004

úprava vstupných údajov, aby sa s nimi lepšie

pracovalo ~~[illegible]~~

$M \subseteq \Sigma \quad T(M) \rightsquigarrow T^{-1}(M') = M$

transf. by mala byť jednoznačná



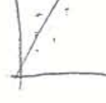
obrázok s reprezentantom pre blízke

body vo vektor. kvantiz.

v dátach ex. asi závislosť, ktorú nevidíme

napr. priamka reprezentujúca závislosť

pre $y \doteq 2x$



otočím obrázok aby priamka bola rovnobežná

y-ové súradnice po otočení majú hodnoty

x-ové súradnice veľké hodn.

$x' = 2x$ transf. otočenia!

transf. danou maticou

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos x & \sin x \\ -\sin y & -\sin y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

(kolmé vekt., jednotkové)

musím poznať uhol

kvôli zachovaniu

metric. vlastností berieme

ortonormálne matice

$\Downarrow$

najprv potom $A^{-1} = A^T$

- FORWARD TRANSFORM

- transf. matice sa predkladajú na množinu dát

$x_1 \dots x_n \mapsto x'_1 \dots x'_n$

$x'_1 = \sum_{j=1}^{n} a_{j} x_j$ koeficienty sa vhodne navolia

chceme ortonormálne vektory a_j

$\underbrace{(\ \ )}$

matica $n \times n$ pre a-čkové vektory

alebo sa matica volí najprv (podľa charakteru dát

volím transformáciu

- V ZÁVISLOSTI OD POČTU ÚDAJOV

napr. $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ sa používa u dát, kt. sa príliš často nemenia

kvôli ortonormalite

$(x, x) \mapsto \frac{1}{\sqrt{2}}(2x, 0)$ rovnaké hodnoty $\Rightarrow$ malý 2. člen

súčet   rozdiel

$(3, 1) \mapsto (4/\sqrt{2}, 2/\sqrt{2})$ väčší rozdiel medzi hod. =

$(3, -1) \mapsto (2/\sqrt{2}, 4/\sqrt{2})$ mnoho sa ľ. strana, mnoho

P. strana

pri rovnakých údajoch môžem zanedbávať 2. zložku

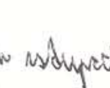
(ušetrím kódovanie jednoduchšie), zameriam sa na 1. zložku

rekonštrukcia v prípade vynechanej 2. zložky:

$\frac{x}{2}$ $4/\sqrt{2} \mapsto (2, 2)$ bližšie hodnoty =

$2/\sqrt{2} \mapsto (1, 1)$ menšia chyba

2D údaje:



transpon.   kvôli orto.

$$\begin{bmatrix} x_{00} & x_{11} \\ x_{10} & x_{11} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x'_{00} & x'_{01} \\ x'_{10} & x'_{11} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}}$$

inverz. transf.

$\frac{1}{2} \Big(x_{00} + x_{01} + x_{10} + x_{11} \quad x_{00} + x_{01} + x_{00} + x_{0}$

$x_{..} + x_{..} + x_{..} + x_{..}$

$n \times n$ ortonormálna. Na inverznú stačí transponovať

$$\left(C_{n\times n} \right)\begin{pmatrix} x_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ \lambda_n \end{pmatrix} DC$$

prvá sa berie presne, ostatné sa nemusia až tak

2D verzia

DC ...

napr. JPEG DC sa pošle presne, ostatné koeficienty cik cak — čím ďalej sme v matici od DC

zložky, tým menší vplyv majú jednotlivé zložky

---

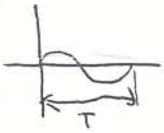
chceme to teda ešte raz, ako to bolo, ako sme to pochopili a ako to je" $\omega = \frac{2\pi}{T}$ 3.5.04.

funkcia $f(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{in\omega t}$ Fourier

koef. sa dá zapísať takto (f periodická s periódou T)

$$c_n = \frac{1}{T}\int_0^T f(t)\, e^{-in\omega t}\, dt$$

$c_n = g(\omega)$



Fourierov rozvoj periodickej funkcie

Fourier transf. je niečo iné, to len aby sa to nepletilo ... „mne sa to pletie"

$$f(t) = \sum_{n=-\infty}^{\infty} \left(\frac{1}{T}\int_0^T f(t)\, e^{-in\omega t}\, dt \right) \cdot e^{in\omega t}$$

môžeme sa dívať na $e^{in\omega t}$ ako na bázu

nekonečnou lineár. kombináciou $e^{in\omega t}$ vieme popísať funkciu

$e^{in\omega t}$ sú navzájom kolmé v nekonečnorozmernom priestore

Fourier. transformácia

definujem skalár. súčin $\langle f(t), g(t)\rangle =$

$$\langle f(t), g(t)\rangle = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)\cdot \overline{g(t)}\, dt$$

komplex. združené

$$\langle e^{in\omega t}, e^{im\omega t}\rangle = \frac{1}{T}\int_0^T e^{in\omega t}\, e^{-im\omega t}\, dt = \frac{1}{T}\int_0^T e^{i(n-m)\omega t}\, dt = \begin{cases} 1 & \text{ak } m = n \\ 0 & \text{ak } m \neq n \end{cases}$$

periodické pre n veľa integrál som napísal 2

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt \qquad f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(t)\, e^{i\omega t}$$

Fourier. transf. tam aj späť

---

máme funkciu typu (skoro nulovú)

vyrobíme z nej periodickú skokovú funkciu (=vzorky po úsekoch)

máme (impulz šírky Δ) a $\Delta \to 0$ Diracova funkcia δ

$\delta(t-t_0)$ všade má hodnotu 0, v t_0 má hodn.

majme $f(t)\,\delta(t-t_0) = f(t_0)$.. samplovaná funkcia

naším cieľom je diskrét. Fourier transf.

majme N nameraných bodov

$$F_K = \frac{1}{T}\int_0^T f(t) \cdot \underbrace{\sum_{n=0}^{N-1} \delta\left(t - \frac{n}{N}t\right)}_{} e^{iKt\omega}\, dt$$

K.. koeficient — postupnosť N meraní v tvare funkcie

koeficient rozvoja

$$= \frac{1}{T}\sum f\left(\frac{n}{N}t\right) \cdot e^{\frac{i2\pi Kn}{N}}$$

$$= \sum_{n=0}^{N-1} f_n\, e^{\frac{i2\pi Kn}{N}}$$

$f_0, f_1, \dots f_n \dots$ merania (alebo frekvencie normalizované)

naspäť: „to už nebudem opisovať .. ak ti čo sú podstatnejšie v analýze .."

$$f_n = \frac{1}{N}\sum_{k=0}^{N-1} F_k\, e^{\frac{i2\pi k.n}{N}}$$

FAST FOURIER TRANSFORMATION − FFT

− rýchle vyhodnocovanie polynómov

$A_K = \sum_{i=0}^{n-1} a_i \cdot \omega^{Ki}$, $\omega^n = 1$

rýchle spočítame A_K ak máme dané a_i

buď spočítam normálne, alebo napíšem ako maticu

$$(A_1, \dots, A_n) = (a_1, \dots, a_n) \begin{pmatrix} 1 & \cdots & \omega^{n-1} \\ \vdots & & \\ 1 & & \omega^{k(n-1)} \end{pmatrix}$$

stačí $n \log n$ násobení

...rmaké hodnoty : vľavo-hore veľké čísla
vpravo-dole hodnoty blízke 0

$(x, x, x, x) \mapsto \begin{pmatrix} 2x & 0 \\ 0 & 0 \end{pmatrix}$

...ový roh = DIRECT CURRENT jednosmer. zložka
...tatné AC = striedavá zložka (udávajú rozdiely od DC zložky)
...rovnou sa robí priemer z okol. stĺpcov.

...de o vymyslenie ortonormal. matíc, ktoré sa rýchlo ...átajú.

DISCRETE WALSH HADAMARD TRANSF: DWHT

zovšeobecnenie predch. postupu

$H_1 = 1$ $\quad H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$H_{2^k} = \begin{pmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{pmatrix}$ $\quad H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$

$H_8 = \cdots$

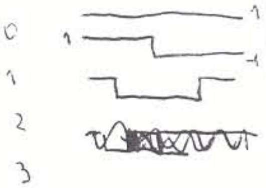
H_k obsahuje (okrem prvého) všetkých možných 2^{k-1} rozdelov
v každom riadku sú polovica +1-tiek a polovica -1-tiek
zpermutované na všetky spôsoby
Hadamard. matica vznikne reorganizáciou (permut.) riadkov, aby počet zmien znamienok v riadku

$\overline{H_n} = H_n \cdot P \cdot \frac{1}{\sqrt{n}}$ kvôli norme

$$H_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & +1 & +1 \\ 1 & -1 & -1 & 1 & -1 & +1 & +1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 7 \\ 3 \\ 4 \\ 1 \\ 6 \\ 2 \\ 5 \end{pmatrix}$$

počet zmien znamienok 0
mali by vzniknúť rôzne čísla

$$\overline{H_8} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ & & & \vdots & & & & \end{pmatrix} \cdot \frac{1}{\sqrt{8}}$$



transformáciou n-tice dostaneme n-ticu
prvý člen: súčet všetkých
2. člen: rozdiely medzi prvou $\frac{n}{2}$ ticou a druhou $\frac{n}{2}$ ticou
čím ďalšie členy, tým informovanie o jemnejších vlastnostiach
vľavo hore bude DC ale menej podstatné

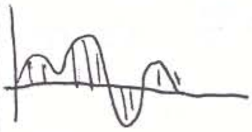
jemné zmeny môžem zanedbávať, alebo aplikovať nejaký filter
výhoda: rýchly výpočet
ale musia presnosť

rovnako lineárny
obraz: 2D: násobenie transf. maticou kvôli horiz. i vert. výpočtom

---

DCT: discrete cosine transform

zmiešanie 2x častí aho frekv. amplitúdy



$f(x) = \sum_{i=1}^{N} \cos \frac{i}{n} \cdot x \cdot C_i$

v súlade: pre komplex. fcn
discrete konečný počet aproximácií

namiesto všetkých fcie, nakopírujem ju aby sme dostali periodickú fcn

z N nameraných hodnôt vyrobím N koeficientov

chceme ale spojitosť: vezmem symetrickú

tak to vieme vyjadriť iba pomocou cosínusov (netreba nám Fouriera ...)

cos() je symetrický

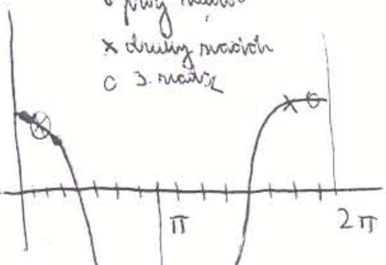
mám nameraných 2N hodnôt, ale stačí brať N (ostatné ruším symetriou)

$$C_n = \begin{cases} \frac{1}{\sqrt{n}} \dfrac{\cos (2j+1) i \cdot \pi}{2n} & i = 0 \\ \sqrt{\frac{2}{n}} \dfrac{\cos (2j+1)\, i\, \pi}{2n} & i \neq 0 \end{cases}$$

i.. riadky
j.. stĺpce

matica (4×4)

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ \sqrt{2} \cos \frac{\pi}{8} & \sqrt{2} \cos \frac{3\pi}{8} & & \\ \vdots & & & \end{pmatrix}$$



1 riadok jednotlivé
2. riadok klesá
3 riadok pol v plusoch, pol v mínusoch

$_k = a_0 \omega^0 + a_2 \omega^{2k} + \ldots + a_{n-2}\, \omega^{(n-2)k}$

$+ \omega^k (a_1 \omega^0 + a_3 \omega^{2k} + \ldots + a_{n-1} \omega^{(n-2)k})$

$_{k+\frac{n}{2}} = a_0 \omega^0 + a_1 \omega^{k+\frac{n}{2}} + a_2 \omega^{2(k+n/2)} + \ldots a_{n-1} \omega^{(n-1)(k+n/2)}$

$a_0 \omega^0 + a_2 \omega^{2k+n\omega} + \ldots + a_{n-2} \omega^{(n-2)k} -$

$\omega^k (a_1 \omega^0 + a_3 \omega^{2k} + \ldots + a_{n-1} \omega^{(n-2)k})$

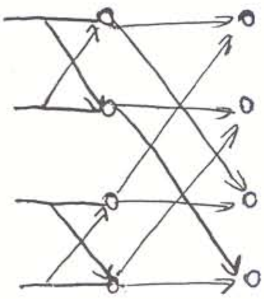
$$\boxed{\begin{aligned} _K &= B_k + \omega^k C_K \\ _{K+\frac{n}{2}} &= B_k - \omega^k C_K \end{aligned}}$$

často po mocninách 2 by sa robi

$_{10} = a_0 \qquad B_K = a_0 + a_2 (\omega^2)^K$

$C_K = a_1 + a_3 (\omega^2)^K$

ute



analóg. obvody

na cvičení

na skúške ...

JPEG – „expertná skupina nejlepších fotografov"

„umelci pracujú na joints"

sekvenčné DCT

progresívne DCT

losless

hierarchické

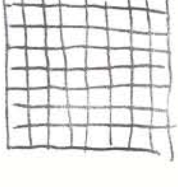
matica celých čísel $\in \langle 0, 2^c \rangle$

často $\langle 0, 255 \rangle$

) posun do rozsahu $\langle -128, +127 \rangle$

) rozdelíme obrázok na rámce 8 × 8 pixelov

doplní sa 0 ami alebo náhodné alebo konc. farby



3) DCT

vonkajšie riadky, urobím DCT

ďalší riadok DCT

⋮

potom 1. stĺpec, DCT

2. stĺpec, DCT

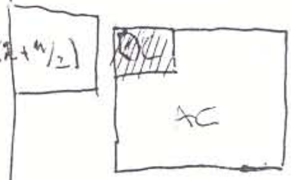
Kcem ked dostanem DC, ak sú dobré hodnoty

C je priemerom

$a_{xy} = \frac{c_{ij}}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f_{orig}(x,y) \cos \frac{(2x+1) i \pi}{16} \cos \frac{(2y+1) j \pi}{16}$

$\sqrt{\frac{2}{N}} \cos \frac{(2j+1) i}{2N}$

malo by byť $a_{00} = \frac{\sum farba}{N^2}$



4) DC .. kódovanie diferenčnou metódou

AC .. kvantizátor

$b_{ij} = \left\lfloor \frac{a_{ij}}{Q_{ij}} + 0{,}5 \right\rfloor$ mid-thread quantizer

Q_{ij} – uniformný skokový krok

pevný

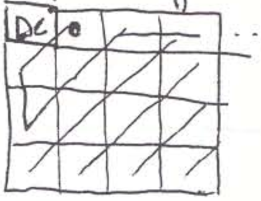
kvantizačná tabuľka: vlastná (dodaná so súbor

štandardná

hodnoty menšie v rohu, smerom vpravo dole sa zväčšujú (100 115)

11-15

naspäť: b_{ij} len prenásobím kvantizač. koeficientom



ako prechádzam maticu a kódujem

aj DC sú kvantované (tú mám ani reál. č)

(dĺžka, amplitúda)

DC	5	3	1	0	0
−4	0	0	5		
2	1	3			
−6	0				
0					

(0,5) (0,−4),

(0,3) (1,2),

(0,1) (1,1),

(0,1) (0,−6),

(2,3) (0,5),

⋮

EOB = ďalej idú samé 0

5) tabuľka kódov

(0,1) ↦ (0,0)	neexamplované do 16
(0,2) ↦ (0,1)	na sekvenciu len 16 núl
(0,3) ↦ (100)	
EOB ↦ 1010	

(Huff kód)

DC: aritmetické kódovanie $DC^{(n)} - DC^{(n-1)}$

$c_{ij} = \begin{cases} 1/2 & i=j=0 \\ \frac{1}{\sqrt{2}} & i=0 \vee j=0,\ i \neq j \\ 1 & \ldots \text{inokedy} \end{cases}$

cos sa často tabelluje

„keby ten kód skúšaný matematik tak by ste si písali vzorčeky" progresívne

spektrálny výber: najprv dekódujem DC mám 8×8 jednej farby, potom vezmem ďalšie členy, ďalšie. postupne sa to vylepšuje

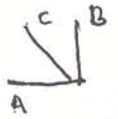
postupná aproximácia

všetky na začiatku pošlem s hrubou kvantizáciou, postupne spresňujem kvantizáciu

$Q_1(DC, AC, ..)$ | $Q_2(DC, AC)$

hrubá kvant. jemnejšia kvant.

[illegible] predikcie



- Bez
- A
- B
- C
- A+B−C
- $A+\frac{B-C}{2}$

⋮

po 8×8 predikcia

$DC \in \langle -2048, 2048 \rangle$

17. 5. 2004

$$\mathcal{F}\left(\sum_{-\infty}^{\infty}\delta(t-nT)\right) = \sigma_0 \sum_{-\infty}^{\infty}\delta(\omega - n\sigma_0) \qquad \sigma_0 = \frac{2\pi}{T}$$

JPEG2000 dôležitejšie frekvencie, smery

obrázok → LL, 4 podmatice

4 štvorčeky

rozdiely: dvojné rozdiely, malé hodnoty

vezmem opäť LL a rozdelím

asi 4 stránky (10 úrovňová hierarchia)

na konci čísla sú stále polovičné

1	1/2	1/4
1/2	1/2	
	1/4	1/4

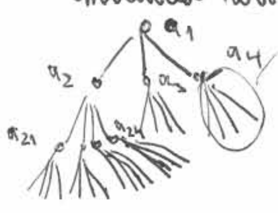
kódovanie: vezme sa prah

(pre kvantizačnú fciu) a stále sa berie 1/2 pre

ďalšiu úroveň prahu

t_0 { $-t_0$ .. pod; medzi; t_0 ... nad polovicou, filter sa zmenšuje — ďalej sa interval zmenší na

EZW – Embedded zerotree wavelet



podmatice

$T_0 = 2^{\lfloor \log_2 c_{max} \rfloor}$ — 1. prah; c_{max}: max. koef. v tabuľke

a_1	a_2	a_{21}	a_{22}
a_3	a_4	a_{23}	a_{24}
a_{31}	a_{32}	a_{41}	a_{42}
a_{33}	a_{34}	a_{43}	a_{44}

ak sú v nejakom podstrome sú < ako hranica, poviem, že nasleduje prázdny podstrom

posielam

p ak $> T_0$

n ak $< T_0$

z $(-T_0, +T_0)$ príliš malá hodnota, ale v potomkoch ešte sú dostatočne veľké hodnoty

t aj potomkovia $\in (-T_0, T_0)$ (zero tree)

podobné SPIHT: podstromy

26	6	13	0
-7	7	6	4
4	-4	4	-3
2	-2	2	0

7-úrovňový EZW

max = 26 $\quad T_0 = 2^{\lfloor \log_2 26 \rfloor} = 16$

p, t, t, t (26, 6, -7, 7) pre T_0 stačí 8b (2b/znak)

$T_1 = 8$ 26 je hotové, môžno tie, čo nie sú zero tree nechám

6: z t t (-7, 7)

p p z z

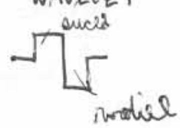
13 0 6 4

plus 6 b

z t t p p z z

„mohli by sa to ešte raz zopakovať?" -- to keby som ja vedel..

HAAR BASE WAVELET



$T_2 = 4$

*	6	*	*
-7	7	6	4
4	-4	4	-3
2	-2	2	0

$(-2T_2, +2T_2)$

p n p p p p n z z p z z z

p n p

6 -7 7

p p p n z z p z z z

6 4

stačilo 48b pre maticu 4×4

DEKÓD:

Stačí asi T_0 poslať

ak p: beriem polovicu $\frac{3}{2}T_0$

hodnoty $\in \langle T_0, 2T_0)$

$T_1 = 8$

z.. nejaký podstrom

t, t: } nič nepovie

8 - 16

$T_2 = 4$

p.. $\langle 4; 8) = 6$

n.. -6

p

p +6

p +6

n

z

z

24	6	12	12
-6	+6	6	6
6	-6	6	0
0	0	0	0

„podlaha" vyššie čísla... presnejšie

„malé bity = bity"

viac hierarchií = viac kvantizátorov

MPEG

snímky idú za sebou, môžem vobrať ako ďalší rozmer

film: 24

TV: 25

luminancia ... „luminancia ... jas"

namiesto RGB: jas $R\,0{,}3 + 0{,}6\,G + 0{,}1\,B = Y$

C_B chrominancia: $B-Y$

C_R: $R-Y$

$3{,}725\ MHz$ /

(Y, C_B, C_R) 4:2:2 : frekvencia samplovania

obraz ... buď menej časté samplovanie / kvôli kombinácii

4 údaje o jase + údaj o farbách

formáty: CIF: Common Interface Format

norma H.261 / H.263 streaming video:

symetrické algoritmy

kódovanie trvá rovnako dlho ako dekódovanie

360 × 240

obraz → blok 8×8 → DCT → kvantizač filter → prenos

Q^{-1}

DCT^{-1}

predikcia pohybu

doteraz počítam rozdiely od predch. obrazu

okolo na 64 kb/s

MPEG: asymetrické „lepšie na prenos obrazu"

MPEG-1: 1,5 Mbit/s   360 × 240 Y 180 × 120 C_B, C_R

MPEG-2: profily

makroblok

4 jasové

1 rozdielová B

1 rozdielová R

6 × 8 × 8 údajov

„ja by som na to potreboval tak 2 dni...

ja som to minule vedel, ale keď mám také blbé poznámky"

Group of Blocks /

1:10

I-Frame: normálne, bez predikcie cez DCT, Q

spojím prediktor P-Frame: rozdiely od predch. ~~frame~~

B-Frame: obojsmerná predikcia

spraví sa predikcia dopredu aj dozadu a vezmem lepšiu (keď mám napr. strih medzi framami, rozdiel bude veľmi veľký pri P-frame)

1:80 kompresia s 2smer. predikciou

schémy

I B B P B B P B B P B B I

bitstream order: prekódované framy v súbore

I P B B P B B P B B I B B

MPEG-4: hýbajúce sa objekty na predikovanie

ako popísať objekty v scéne? pozície!

ako ich vyhľadávať?

raz za čas dám I-Frame

predikovaných hodnôt ... 1:40