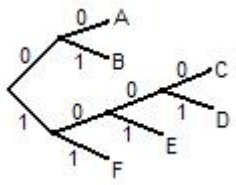
Datu saspiešana II

(konspekta autors: Mārcis Pinnis)

Hofmana kodēšana – paveids kodēšanai ar prefiksu koku.



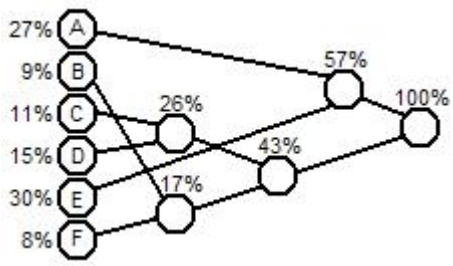
Prefiksu kokā burti tiek kodēti ar simbolu virkni ceļā no saknes uz doto burtu. Attēlā pa kreisi redzamajā kokā „A" tiek kodēts ar simbolu virkni „00", savukārt „D" tiek kodēts ar simbolu virkni „1001" utt.

Hofmana kodēšanā biežāk sastopamajiem burtiem tiek izmantotas mazākas bitu virknes, līdz ar to sakodētā informācija aizņem mazāk vietas.

Hofmana algoritms – izveido katram burtam lapu.

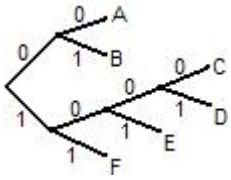
Pieņemsim, ka ir dots katram burtam sastopamības biežums tekstā:

A. 27%
B. 9%
C. 11%
D. 15%
E. 30%
F. 8%



Izmantojot burtus izveidojam nesaistītus koka mezglus katram burtam.

Kamēr ir vairāk par diviem kokiem – atrod divus burtus ar vismazākajiem biežumiem, apvieno tos jaunā kokā (biežums jaunajai virsotnei vienāds ar sākotnējo divu summu).



Kad visas virsotnes apvienotas, varam pārzīmēt koku, lai tā šķautnes nepārklātos:

Iegūtais ir **Hofmana koks**.

Teorēma: Hofmana algoritms vienmēr dod optimālo prefiksu koku.

Ieviešam apzīmējumus: burta x biežums – P_x, bitu virknes garums l_x.

Vidējais bitu skaits uz vienu simbolu ir:

$$\sum_x P_x l_x$$

Šo lielumu vēlamies pēc iespējas mazāku, līdz ar to cenšamies to minimizēt.

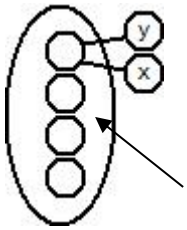
Teorēmas pierādījums (ar matemātisko indukciju):

Bāze: ja alfabētā ir 1 burts, ir tikai viens koks 

, kas ir gan optimālais, gan Hofmana koks.

Indukcijas pāreja (gadījumam, ja ir vismaz divi burti).

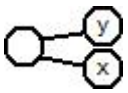
Pieņemam, ka Hofmana algoritms vienmēr dod optimālu koku priekš k-1 burtiem; dots alfabēts A ar k burtiem; x un y ir divi visretāk sastopamie burti.



Pirmajā solī tiek apvienotas virsotnes x un y. Izveidojas jauna virsotne, kuras biežums ir P_x+P_y.

Tālāk ir jāpielieto Hofmana algoritms k-1 burtam.

Pēc indukcijas pieņēmuma Hofmana algoritms k-1 burtam dod optimālo koku. Tas nozīmē, ka Hofmana algoritms dod optimālo koku starp kokiem, kuros x un y atrodas blakus.

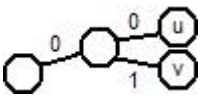


Lai pabeigtu teorēmas pierādījumu, jāpierāda, ka citus kokus var pārveidot par ekvivalentiem kokiem, kur x un y ir blakus.

Optimālā kokā izpildās sekojoši nosacījumi:

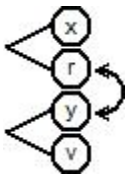
1) Ja $P_x<P_y$, tad $l_x \geq l_y$ (Ja $l_x<l_y$, tad apmaina *x* un *y* vietām kokā. Līdz ar to x piešķir l_y un y- l_x. Tas samazina vidējo bitu skaitu uz burtu un tas nozīmē, ka sākotnējais koks nebija optimāls.)
2) Apskatīsim maksimālo virknes garumu $l_{max}=\max_x l_x$.

   Ir ≥ 2 burti u, v, kuriem $l_u=l_v=l_{max}$.



Lai to pierādītu, pieņemsim, ka $l_u=l_{max}$. Ja šķautnes uz v nebūtu, tad varētu saīsināt u kodējumu par vienu šķautni. Tātad arī $l_v=l_{max}$.

1+2) $l_x=l_y=l_{max}$



x, y – 2 retāk sastopamie burti.

Apmainot y vietām ar burta x kaimiņu, no jebkura optimāla koka var iegūt citu optimālu koku, kuram x un y ir blakus.

Entropija.

Ja burta x biežums ir P_x, tad alfabēta entropija ir

$$H = -\sum_x P_x \log_2 P_x$$

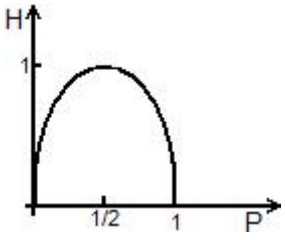
Piemērs:

$\{0,1\}$ – alfabēts

burta „0” biežums ir p

burta „1” biežums ir 1-p

$$H = -p\log_2 p - (1-p)\log_2(1-p)$$



Entropija piemēra gadījumā ir informācijas daudzums tekstā, ja katrs burts ir 0 ar varbūtību p un 1 ar varbūtību 1-p.

Ja $p=1/2$, tad $H=1$

Ja $p=0$, tad $H=0$

Priekš k burtiem:

H – **maksimālā** vērtība, ja burtu biežumi ir vienādi ar $1/k$

$$H = \log_2 k$$

H – **minimālā** vērtība, ja burtu biežumi ir $1,0,0,\ldots,0$

$$H = 0$$

Saikne starp prefiksu kodēšanu un entropiju.

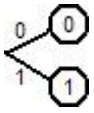
Teorēma: jebkuram prefiksu kokam

$$\sum_x P_x l_x \geq H$$

Tas ir, vidējais bitu skaits uz burtu ir vismaz tik liels, cik alfabēta entropija.

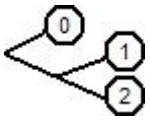
Gadījumi, kad Hofmana koks sasniedz entropiju.

1) $p_0 = p_1 = \frac{1}{2}$



Hofmana koks. Katrs simbols tiek kodēts ar 1 bitu.
$H=1$

2) Ja $p_i = \frac{1}{2^j}$, j atkarīgs no i.

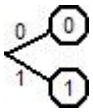


$p_0 = \frac{1}{2}, p_1 = \frac{1}{4}, p_2 = \frac{1}{4}$

Vidēji $1,5$ biti vienam simbolam.
H arī ir $1,5$.

Cik liela var būt atšķirība starp entropiju un Hofmana koku?

$p_0 = 1$



$p_1 = 0$

$$\sum_x p_x l_x = 1$$

$H = 0$

Gadījumā, kad alfabēts sastāv no vairākiem burtiem, tad vidējais bitu skaits uz burtu (ja vienam no burtiem sastopamības biežums ir 1, bet pārējiem 0) arī būs 1, un entropija būs 0.

$p_0 = 1 - \delta$
$p_1 = \delta$

Ja δ tuvs 0, tad $H \approx 0$.

Teorēma: ja $M = \sum_x p_x l_x$ – Hofmana kokam, tad $H \leq M \leq H + 1$

Teorēma: $M \leq H + max_x p_x + 0{,}086$.

Vai eksistē metode, ar kuru var sasniegt H?

Ja $l_x = -\log_2 p_x$, tad $\sum_x p_x = -p_x \log_2 p_x = H$

Problēma: l_x var nebūt veseli skaitļi.

Ko nozīmē: kodēt „a" ar $0{,}4$ bitiem?

Atbilde saistīta ar **aritmētisko saspiešanu**.

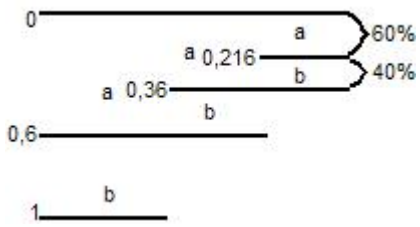
Teksts ▲ skaitlis starp 0 un 1 ▲ skaitļa pieraksts binārajā sistēmā ar noteiktu precizitāti.

Piemērs:

a $\quad p_a = 0{,}6$

b $\quad p_b = 0{,}4$

Pieņemsim, ka kodējam aab



Ja 1. burts ir a, tad simbolu virkne atradīsies zem $0{,}6$.

Līdz ar to aab atbilst jebkuram skaitlim intervālā $[2{,}16;0{,}36]$.

Piemēram, $0{,}25 = (0{,}01)_2$

Ja $[cx,dx]$ – atbilst simbolu virknei x, tad:

$[cx,0{,}6dx+0{,}4cx]$ atbilst xa;

$[0{,}6dx+0{,}4cx,dx]$ atbilst xb.

Atkodēšana:

Kādai simbolu virknei atbilst $0,4$?

Precīza atkodēšana, izmantojot tikai skaitli nav iespējama, jo $0,4$ var atbilst:

- $a\ [0;0,6]$
- $ab\ [0,36;0,6]$
- $aba\ [0,36;...]$
- $abaa\ [0,36;...]$
- ...

Līdz ar to, lai precīzi atkodētu, jāpārraida arī burtu skaits ziņojumā.

Viennozīmīgai atkodēšanai vajadzīgo precizitāti var sasniegt, izmantojot bitu skaitu, kas ir aptuveni vienāds ar vārda garumu reizinātu ar H.

Kā var pārsniegt entropiju?

- Datu saspiešanas metode nevar būt labāka par entropiju, ja katrs simbols izvēlēts neatkarīgs.
- Reālos datos nākošais simbols atkarīgs no iepriekšējā.
- Vienkāršākais veids ir definējot 2 burtu virknes kā simbolus $\{aa, ab, ac, ad, ...\}$ un pielietojam Hofmana vai aritmētisko metodi.
- Otra pieeja ir izmantojot to, ka ir simbolu virknes, kas atkārtojas. Ievieš jaunus simbolus priekš šādām virknēm.
- Tas dod Lempela-Ziva metodi, kas tiks apskatīta nākošajā lekcijā.