

Információelmélet

Informatikai rendszerek alapjai

Horváth Árpád

2015. október 29.

1. Információelmélet alapfogalmai

Információelmélet

Egy jelsorozat esetén vizsgáljuk, mennyi információt tartalmaz.

Nem érdekel minket a jelek tényleges jelentése.

Ahonnán a jelek jönnek, *forrásnak* nevezzük. A jelek összességét a *forrás jelkészletének* (szimbólumkészletének, ábécéjének) nevezzük.

Egyszerűség kedvéért feltételezzük, hogy véges számú jel van (diszkrét jelkészlet). Ezt nevezzük *diszkrét forrásnak*.

1.1. Információtartalom

Mekkora egy jel információtartalma?

Mennyi igen-nem válasszal határozhatunk meg egy kártyát a magyarkártya-pakliból?

Makk VIII-as (M,VIII)

M,VII	M,VIII	M,IX	M,X	M,A	M,F	M,K	M,Á
T,VII	T,VIII	T,IX	T,X	T,A	T,F	T,K	T,Á
Z,VII	Z,VIII	Z,IX	Z,X	Z,A	Z,F	Z,K	Z,Á
P,VII	P,VIII	P,IX	P,X	P,A	P,F	P,K	P,Á

M, VII	M, VIII	M, IX	M, X	M, A	M, F	M, K	M, Á
T, VII	T, VIII	T, IX	T, X	T, A	T, F	T, K	T, Á
Z, VII	Z, VIII	Z, IX	Z, X	Z, A	Z, F	Z, K	Z, Á
P, VII	P, VIII	P, IX	P, X	P, A	P, F	P, K	P, Á

M, VII	M, VIII	M, IX	M, X	M, A	M, F	M, K	M, Á
T, VII	T, VIII	T, IX	T, X	T, A	T, F	T, K	T, Á
Z, VII	Z, VIII	Z, IX	Z, X	Z, A	Z, F	Z, K	Z, Á
P, VII	P, VIII	P, IX	P, X	P, A	P, F	P, K	P, Á

M, VII	M, VIII	M, IX	M, X	M, A	M, F	M, K	M, Á
T, VII	T, VIII	T, IX	T, X	T, A	T, F	T, K	T, Á
Z, VII	Z, VIII	Z, IX	Z, X	Z, A	Z, F	Z, K	Z, Á
P, VII	P, VIII	P, IX	P, X	P, A	P, F	P, K	P, Á

M,VII	M,VIII	M,IX	M,X	M,A	M,F	M,K	M,Á
T,VII	T,VIII	T,IX	T,X	T,A	T,F	T,K	T,Á
Z,VII	Z,VIII	Z,IX	Z,X	Z,A	Z,F	Z,K	Z,Á
P,VII	P,VIII	P,IX	P,X	P,A	P,F	P,K	P,Á

M,VII	M,VIII	M,IX	M,X	M,A	M,F	M,K	M,Á
T,VII	T,VIII	T,IX	T,X	T,A	T,F	T,K	T,Á
Z,VII	Z,VIII	Z,IX	Z,X	Z,A	Z,F	Z,K	Z,Á
P,VII	P,VIII	P,IX	P,X	P,A	P,F	P,K	P,Á

Kétféle modellt vizsgálunk a források esetén

Legyen egy diszkrét forrás esetén egy jel előfordulási valószínűsége független az előző jel(ek) értékétől. (független=independent)

- *IID modell*: A jelek egyforma valószínűséggel jelennek meg, tehát ha 32 jelem van, mindegyik $1/32$ valószínűséggel. (independent, identically distributed).
- *IRD modell*: A jelek tetszőleges valószínűséggel jelennek meg, tehát hiába van 32 jelem (kártyám), lehet az egyik (pl. piros király) valószínűsége $1/2$ is. (Fontos, hogy ebben az esetben minden pillanatban $1/2$ lesz a valószínűsége az adott jelnek, akármilyen jelek is voltak előtte.) (independent, randomly distributed).

Random jelentései

- véletlen
- *tetszőleges*
- rendszertelen
- találmra történő

A random jelentése itt nem annyira véletlenszerűség, mint inkább tetszőlegesség. A valószínűségek tetszőlegesek lehetnek.

Kiegészítő megjegyzés: A RAM-nál (Random Access Memory) sem véletlenszerűen férek hozzá az egyes bájtokhoz. Inkább azt jeletni, hogy tetszőleges bájtjához könnyen hozzáférhetek. Nem kell végigolvasnom az összes bájtot, hogy hozzáférjek egy adathoz, könnyen ki tudok olvasni akármelyik helyről adatot.

Nem olyan mint a veremtár, amelynél mindig az utoljára berakott értéket érem el először. (LIFO - Last In First Out) A veremtárnál ha egy korábbi érték kell, akkor minden felette levőt ki kell olvasni.

Nem is olyan, mint egy magnószalag, ahol az adatsor elejétől végig kell olvasnom a szalagot.

IID

A 32 kártyából, ha mindegyiket egyforma valószínűséggel választhatják ki, akkor 5 igen-nem válaszból kitalálhatom a kártyát.

Általában igaz, ha 2^m lehetőségem van, akkor m kérdést szükséges feltennem.

$X = \{x_1, x_2, \dots, x_n\}$ jelhalmaz esetén $\log_2 n$ kérdés kell (felfele kerekítve a következő egészre).

Egy jel (pl. véletlen választott kártya) információtartalma az IID modellben

$$I = \log_2 n$$

IRD

Ha nem egyforma valószínűsége van egy-egy kártyának (jelnek), akkor lehet, hogy érdemesebb a kártyaszám-felezéses taktika helyett másikat választani. Inkább a valószínűségeket kell felezni, mint a kártyák (jelek) számát.

Ha pl. a Piros VIII-as jelenik meg az esetek felében, akkor érdemes lehet arra rákérdezni, így azt egy kérdéssel kitaláljuk. Ha így teszünk, akkor lesznek olyan kártyalapok, amelyhez 5-nél több kérdés kell majd, de ha ezek elég ritkán jelennek meg, akkor az átlagos kérdésszám lehet öt alatti.

A ritkábban megjelenő kártya megjelenésének az információtartalma nagyobb. Az, hogy 4 lapom közül mind a négy ász, az nagyobb információtartalmú, mint hogy minden kártyám számos (VII-es, VIII-as, IX-es vagy X-es),

IRD

Ha nem egyforma valószínűsége van egy-egy jelnek, akkor a kisebb valószínűségűnek nagyobb az információtartalma.

$X = \{x_1, x_2, \dots, x_n\}$ jelekhez tartozzanak $P(X) = \{p_1, p_2, \dots, p_n\}$ valószínűségek.

Legyen a rendszer teljes:

$$p_1 + p_2 + \dots + p_n = \sum_{k=1}^n p_k = 1 = 100\%$$

Ekkor minden x_k értékhez valamilyen p_k -től függő információtartalom tartozik.

Az x_k jel információtartalma az IRD modellben:

$$I_k = \log_2 \frac{1}{p_k} = -\log_2 p_k$$

A kisebb p_k valószínűséghez nagyobb információtartalom tartozik, mivel a 2-es alapú logaritmusfüggvény monoton növekvő.

Ha mind egyforma valószínűségű, akkor

$$p_k = 1/n \Rightarrow I_k = \log_2 \frac{1}{p_k} = \log_2 n,$$

visszkapjuk az IID modell képletét.

Az órán vizsgáltuk a következő eseteket

$$X = \{A, B, C, D\}$$

jelhalmaz esetén.

- $P(X) = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$
- $P(X) = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$
- $P(X) = \{0.8, 0.1, 0.05, 0.05\}$

Egy-egy kódot adtam meg az egyes betűkhöz, amivel vizsgáltuk a jelenként felhasználandó bitek számának várható értékét.

Az első esetben az alábbi első eloszlás, a második kettőben az alábbi második kódokkal vizsgáltuk.

$$A = 00, B = 01, C = 10, D = 11$$

$$A = 1, B = 01, C = 001, D = 000$$

Az utolsó esetben a következő táblázatot állítottuk fel:

k	p_k	I_k $\log_2 \frac{1}{p_k} = -\log_2 p_k$	$p_k I_k$	kód	d_k	$p_k d_k$
A	0,8	0,3219 bit	0,2575 bit/jel	1	1 bit	0,8 bit/jel
B	0,1	3,3219 bit	0,3322 bit/jel	01	2 bit	0,2 bit/jel
C	0,05	4,3219 bit	0,2161 bit/jel	001	3 bit	0,15 bit/jel
D	0,05	4,3219 bit	0,2161 bit/jel	000	3 bit	0,15 bit/jel

Az entrópia a 4. oszlop értékeinek összege:

$$H(X) = \sum_{k=1}^n p_k I_k = 1,0219 \text{ bit/jel.}$$

Tetszőleges kódolással ennél csak nagyobb bit/jel arányt érhetünk el, az fenti esetben például átlagosan az utolsó oszlop értékeinek összegét:

1,75 bit/jel

Ha összefognánk több jelet, például három hosszúságú jelsorozatokhoz rendelnénk bitsorozatokat, akkor közelebb kerülhetnénk az entrópia értékéhez, de azalá nem mehetünk, ha a jelnél

feltételezzük, hogy a valószínűségek függetlenek (a modellek első I betűje) attól, hogy korábban milyen jelek jöttek a jelforrásból.

A valódi üzeneteknél a jelek valószínűsége függ az üzenetben szereplő korábbi jelektől (nem IID). Például egy levélben "info" karakterek után nagyobb valószínűséggel jön "r" karakter, mint általában. Ilyen esetek matematikai kezelése nehézkes, mi nem tárgyaljuk.

A kódot most én adtam meg, később majd tanulunk módszert hogyan lehet „jó” kódot készíteni.

A kódból fel kell tudni rajzolni a fát, és kiszámolni minden értéket, ha a kódot és a valószínűségeket megadom. Nem csak 4 jel esetén.

Az információ egységei

Az eddigiekben a kettes alapszám helyett mást is használhatunk:

alapszám (a)	egység
2	bit (Shannon)
10	Hartley
e	Nat

$$I_k = \log_a \frac{1}{p_k} = -\log_a p_k$$

2-es és más alapú logaritmus kiszámítása számológéppel:

$$\log_2 x = \frac{\lg x}{\lg 2} \quad \log_a x = \frac{\lg x}{\lg a}$$

$$p_k = 0,1 \Rightarrow I_k = \lg \frac{1}{0,1} = \lg 10 = 1 \text{ Hartley.}$$

$$p_k = 0,1 \Rightarrow I_k = \log_2 10 = \frac{\lg 10}{\lg 2} = 3,3219 \text{ bit.}$$

A $\lg = \log_{10}$ jele a számológépen **log**.

A $\ln = \log_e$ jele a számológépen **ln**.

Ha egy x_k jelhez $p_k = 0,1$ valószínűség tartozik, akkor annak az információtartalma

$$I_k = \lg \frac{1}{0,1} = \lg 10 = 1 \text{ Hartley,}$$

másképpen $I_k = -\lg 0,1 = 1 \text{ Hartley.}$

A $\lg 2 = 0,3010 \approx 0,3$ értéket érdemes megjegyezni, mert a villanytanban a törésponti erősítés (3 dB-es) értéke (a Bode-diagramon) is ebből következik.

Ennek ismeretében tehát a $p_k = 0,1$ valószínűséghez tartozó információtartalom bitekben fejben kiszámítható jó közelítéssel:

$$I_k = \log_2 10 = \frac{\lg 10}{\lg 2} \approx \frac{1}{0,3} = 3,3.$$

Hasonlóan a $p_k = 1/32$ valószínűséghez tartozó információtartalom Hartley-ban:

$$I_k = \log_2 32 = 5 \text{ bit} \approx 5 \cdot 0,3 \text{ Hartley} = 1,5 \text{ Hartley.}$$

1.2. Entrópia

Entrópia

Mekkora a várható értéke a következő jel információtartalmának?

A valószínűségekkel súlyozni kell az összes jel információtartalmát. Ezt nevezzük *entrópiának*:

$$H(X) = \sum_{k=1}^n p_k \cdot I_k = \sum_{k=1}^n p_k \cdot \log_2 \frac{1}{p_k} \quad \text{bit/jel}$$

IID modellnél:

$$H_{n,IID} = n \cdot \frac{1}{n} \cdot \log_2 n = \log_2 n \quad \text{bit/jel}$$

Mire jó ez nekünk? Meghatározhatjuk, mennyi bit szükséges feltétlenül az információ átviteléhez.

Példa

•

$$X = \{A, B, C, D, E\}, \quad P(X) = \left\{ \frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8} \right\}$$

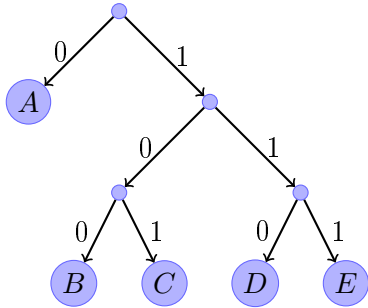
•

$$I_A = 1\text{bit}, \quad I_B = I_C = I_D = I_E = 3\text{bit},$$

$$H(X) = \left(\frac{1}{2} \cdot 1 + 4 \cdot \frac{1}{8} \cdot 3 \right) \quad \text{bit/jel} = 2 \quad \text{bit/jel}$$

Példa

- Legyen A=0, B=100, C=101, D=110, E=111.



- Minden bitsorozat egyértelműen visszafejthető pl: 1000001101110101
- 100,0,0,0,110,111,0,101 = BAADEAC
- Ekkor 8 karakterből átlagosan 4 db „A” (4 bit) a többi négy 3 bites (12 bit): ez összesen 16 bit \Rightarrow 2 bit/jel.

A 2 bit/jel értéket megkaphatjuk úgy is, hogy felírjuk a korábban példaként szereplő táblázatot erre az esetre. Mindkét forma elfogadható, ha a dolgozatban a választ egyértelműen feltünteti a hallgató.

Annak a feltétele, hogy egyértelműen visszafejthető legyen a kód, az, hogy semelyik kód *előtagja* ne egyezzen egy másik kóddal: pl. ha van 001 kód, akkor nem lehet 00 kód is

További mérőszámok

- Mindig az IID esetben a legnagyobb az entrópia azonos jelszám (n) esetén:

$$H_{n,max} = H_{n,IID} \geq H(X), \quad |X| = n$$

- Hatásfok: Az entrópia aránya az ugyanannyi jelet tartalmazó IID modelléhez viszonyítva.

$$\eta = \frac{H(X)}{H_{n,max}}$$

- Redundancia: (hétköznapi jelentés: terjengősség)

$$R = 1 - \eta$$

Gyakoriság, valószínűség helyett A következő gyakoriságokat mértem egy diszkrét forrás által kibocsátott jelek esetén. Határozzuk meg az egyes jelek információtartalmát, az forrás entrópiáját, hatásfokát és redundanciáját.

- A 13
- B 25
- C 12
- D 27
- E 23

Megoldás vázlata:

Összesen 100 jelet figyeltem meg, mert a gyakoriságok összege 100. Ebből a valószínűségek úgy kaphatóak, hogy a gyakoriságot elosztjuk az összegükkel. Pl. $p_1 = P(A) = 13/100 = 0,13$.

Ebből a többi tulajdonság az előzőekhez hasonlóan számolható, amit gyakorlásként érdemes is megtenni.

Példa

-

$$X = \{A, B, C, D, E\}, \quad P(X) = \left\{ \frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8} \right\}$$

- Hatásfok:

$$\eta = \frac{H(X)}{H_{n,max}} = \frac{2 \text{ bit/jel}}{\log_2 5 \text{ bit/jel}} = \frac{2 \text{ bit/jel}}{2,3219 \text{ bit/jel}} = 0,8614 \approx 86\%$$

- Redundancia:

$$R = 1 - \eta = 0,1386 \approx 14\%$$

Ajánlott segédlet

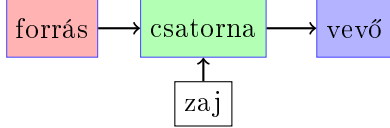
Dr. Tóth Mihály – Tóth Gergely: Az információ- és kódoláselmélet

Kidolgozott példák és feladatok: 1.13.1, 1.13.2, 1.14.28, 1.14.35, 1.14.36, 1.14.37, 1.14.38, 1.14.39,

A feladatok a jegyzet I. részének 24. oldalán kezdődnek.

2. Forráskódolás

A kommunikációs rendszerek vázlata



Példák:

Beszélgetés: **száj** ⇒ levegő ⇒ **fül**

Földfelszíni rádióadás: **rádióadó** ⇒ az „éter” ⇒ a rádió antennája

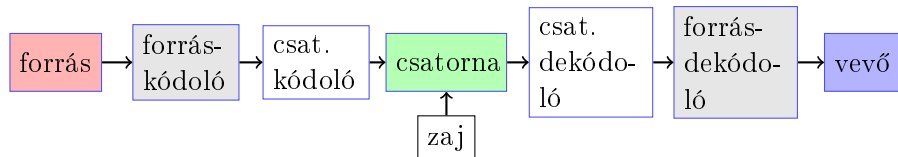
Internet: **router1** ⇒ üvegszál ⇒ **router2**

koncert ⇒ CD ⇒ **emberi fül**

A kommunikációs rendszerek bővebb vázlata

Forráskódolás: a forrás által küldött jelek kódolása legkisebb redundanciával („legkevesebb biten”)

Csatornakódolás: A jelet úgy kódolom, hogy a hibákat észre tudjam venni vagy javítani tudjam.



2.1. Huffman-kódolás

Feladat

Kódoljuk az alábbi eloszlású IRD forrást Huffman-kódolással!

k karakter	n_k darabszám
A	45
B	13
C	12
D	16
E	9
F	5

A Huffman-kódolás algoritmus

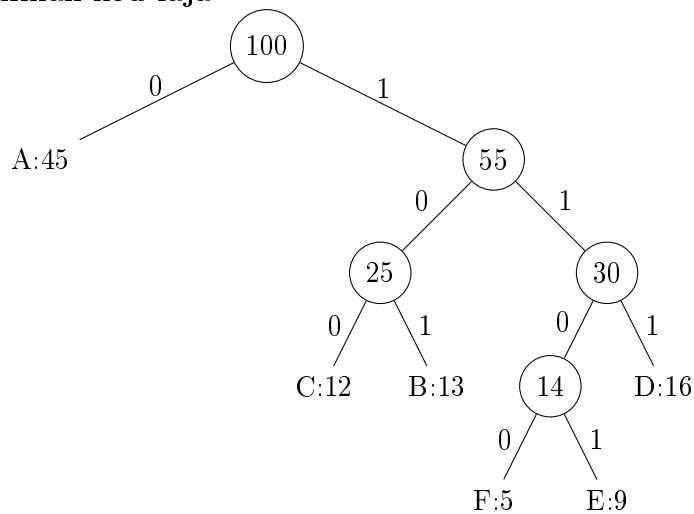
1. Növekvő gyakoriságú sorrendbe rakom a jeleket.
Azonos gyakoriságnál ABC-rendbe (vagy pl. ASCII-kódbeli rendbe).
Kezdetben 1 jel 1 fának számít.
2. Ciklus, amíg van legalább két fa
 - (a) Az első kettő fát összekötöm egy új gyökérrel.
Gyakorisága a két fa gyakoriságának összege lesz.
A baloldalt 0-val, a jobbot 1-gyel címkézem.
 - (b) A következő ábrán növekvő gyakorisági sorrendbe rakom a fákat. (Felül hagyok helyet az összekötésnek.)
Azonos értékeknél az újat lehető leghátulra rakom.

Ciklus vége

3. Leolvasom a jelek kódjait.

A két részektől nem függ a kódolás hatékonysága, de egyezően kell a forrásnál és a vevőnél kódolni.

Huffman-kód fája



B kódja: 101, F kódja: 1100

ASCII-tábla, 7 bites

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	'	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Q: 0x51

Szóköz (space=SP), szá-

9: 0x39

szóköz: 0x20

WALL: 0x57414C4C

mok, nagybetűk, kisbetűk helye

Amit érdemes megjegyezni A szóköz (0x20) után állnak a műveleti jelek és az írásjelek (.,!?:;) a számok (0x30–0x39) körül, majd először a nagy (0x41-től), majd kisbetűk (0x61-től). A 7 bites ASCII-ban nincsenek ékezetes betűk; a 8 bitesben valahol a 128=0b1000 0000=0x80 érték felett, ahol a legértékesebb bit (MSB) 1-es értékű.

A pontos intervallumot nem kell megjegyezni, a sorrendet, ha egy feladatban kell, megadom.

Ha a következő karaktersorozatot a Huffman-algoritmussal tömöríttem, akkor mi lesz az egyes jelekhez rendelt bitsorozat?

BENEDEK_ELEK

Az üzenethez tartozó bitsorozat:

A Huffman-kódban hány bit jut átlagosan egy jelre? ($\langle d_k \rangle$)

Mekkora lesz az egyes karakterek egyedi információtartalma (I_k)?

Milyen alsó határt számolhatunk a szükséges bitek számára (H)?

Mi lesz a 10101001110110011110 bitsorozathoz tartozó üzenet?

A Huffman-kóddal kapcsolatos eredmények

k	n_k	kód	d_k kódhossz [bit]	$n_k d_k$
⌊	1	1100	4	4
B	1	1101	4	4
D	1	1110	4	4
L	1	1111	4	4
N	1	100	3	3
K	2	101	3	6
E	5	0	1	5
Σ	12 jel			30 bit

$$\langle d_k \rangle = \frac{30 \text{ bit}}{12 \text{ jel}} = 2,5 \text{ bit/jel}$$

A bitsorozathoz tartozó üzenet: **KEND_LE**

Ha egy forrásban olyan valószínűségekkel jönnek a jelek, mint a **BENEDEK_ELEK** szóban az arányuk, mindig az előző jeltől függetlenül valószínűséggel (IRD forrás), akkor nem lehet az entrópiánál kevesebb átlagos bittel kódolni akárhogy trükközök.

Általános információelméleti eredmények: entrópia

k jel	n_k	p_k	$I_k = \log_2(1/p_k)$ [bit]	$p_k I_k$
┘	1	1/12	$\log_2 12/1 = 3,585$	0,2987
B	1	1/12	3,585	0,2987
D	1	1/12	3,585	0,2987
L	1	1/12	3,585	0,2987
N	1	1/12	3,585	0,2987
K	2	2/12	$\log_2 12/2 = 2,585$	0,4308
E	5	5/12	$\log_2 12/5 = 1,263$	0,5262
össz	$n = 12$			$H=2,451$ bit/jel

$$H = \sum_k p_k \log_2 \frac{1}{p_k} = \sum_k p_k I_k$$

$$H = 4 \cdot 0,2987 + 0,4308 + 0,5262 = 2,451 \text{ bit/jel}$$

Általános információelméleti eredmények: hatásfok

Az IRD forrás entrópiája

$$H = 2,451 \text{ bit/jel}$$

7 jel esetén a maximális entrópia (amikor minden jel 1/7 valószínűséggel jön)

$$H_{7,max} = \log_2 7 = 2,807 \text{ bit/jel}$$

A hatásfok:

$$\eta = \frac{H}{H_{7,max}} = 0,873 \approx 87\%$$

A redundancia

$$R = 1 - \eta \approx 13\%$$

Adaptív Huffman-kódolás

A Huffman-kódolás esetén előre kellene tudnom, hogy milyen a *forrás statisztikája*. Gyakorlatban gyakran nem tehetem meg, hogy végigvárom a jelsorozatot, és csak utána kezdek el kódolni.

Sőt ez a statisztika *időben változhat*. Van olyan változata a Huffman-kódolásnak, amely a kód-szavakat időben változtatva hozzáigazítja a közelmúltbeli statisztikához. Tehát a kódhosszak úgy változnak, hogy várhatóan egyre hatékonyabb lesz a kódolás.

Általában *adaptív kódolásnak* nevezzük az olyan kódolásokat, amelynél a forrás tulajdonságainak figyelembe vételével egyre hatékonyabban tudom kódolni a szöveget. A hatékonyabb alatt azt értem, hogy átlagosan kevesebb bittel jelenként.

2.2. Aritmetikai kódolás

Aritmetikai kód: kódolás

Az aritmetikai kódolásnál a $[0; 1[$ intervallumot szűkítgetjük az egymás utáni jelekkel.

A fixpontos szám alakja $0,bbbbbb_b_2$

ahol a b-k bináris számjegyeket (1 vagy 0) jelölnek (itt például 8 darabot).

(A kezdő nullát felesleges tárolni.)

Az aritmetikai kód hatékonyabb...

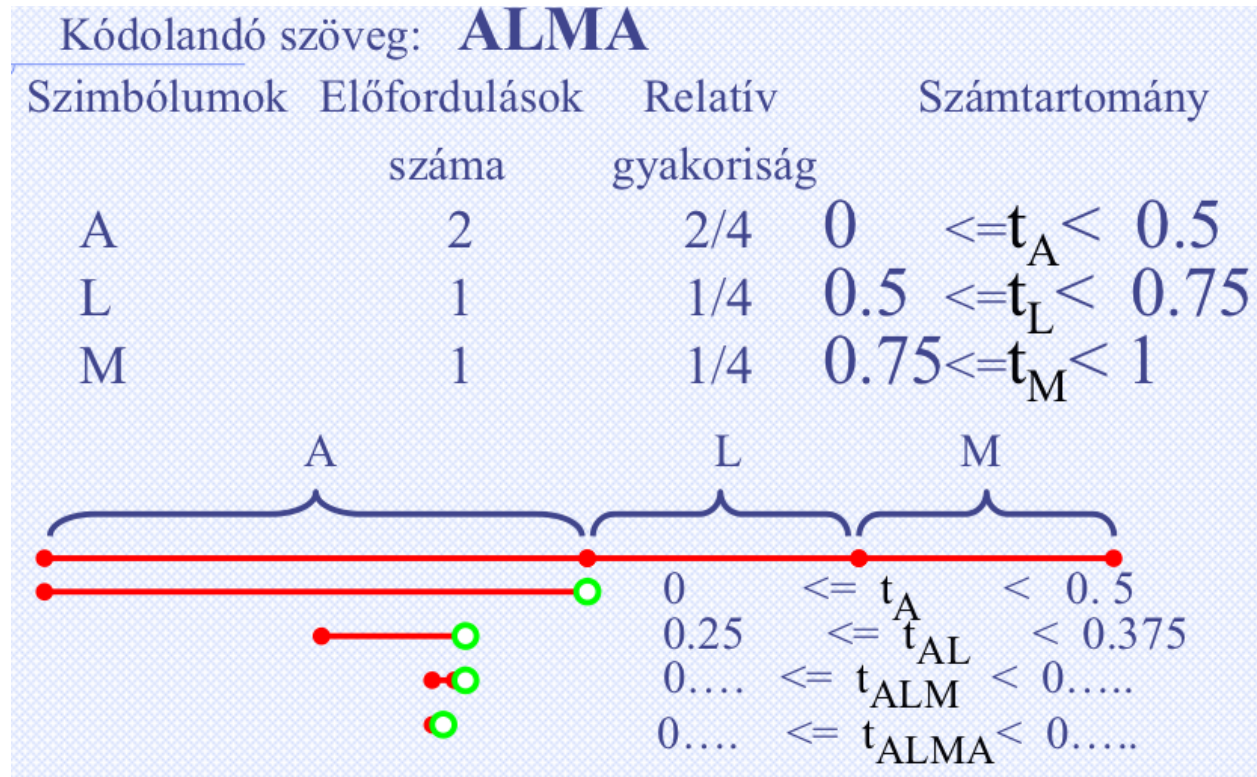
A nulla utáni 8 bináris számjegy esetén például a következő értékek tárolhatóak:

$$0, \frac{1}{256}, \frac{2}{256}, \frac{3}{256}, \dots, \frac{254}{256}, \frac{255}{256}$$

A következő példában ennek a lépésköznek az eléréséig A-ból 8-at, L-ből csak 4-et kódolhatunk (8 bit szükséges az AAAAAAAA és az LLLL üzenethez is), tehát egy A-ra 1 bit, egy L-re 2 bit jut, mint a Hamming-kódolásnál.

Olyan valószínűségeknél viszont, ahol az információtartalom nem egész bit, *az aritmetikai kód kevesebb bitből áll, mint a Hamming-kód.*

Egyszerű példa aritmetikai kódolásra



- A fenti példában az intervallumok hossza annak felel meg, hogy az adott üzenet milyen valószínűséggel áll elő, ha a forrásban a jelek az előző jelektől függetlenül mindig a fenti valószínűségekkel érkeznek (IRD forrás).

-

$$p_A = 1/2$$

$$p_{AL} = 1/2 \cdot 1/4 = 1/8$$

$$p_{ALM} = 1/2 \cdot 1/4 \cdot 1/4 = 1/32$$

$$p_{ALMA} = 1/2 \cdot 1/4 \cdot 1/4 \cdot 1/2 = 1/64$$

$$p_{LMLM} = 1/4 \cdot 1/4 \cdot 1/4 \cdot 1/4 = 1/256$$

- A továbbiakban a szűkítés során az üzenet feldolgozott szakaszának valószínűségét fogjuk $p_{\text{üzenet}}$ -tel jelölni. Ha egy lépés során a k jel jön, akkor $p_{\text{üzenet}}$ értéke a k jel valószínűségével fog szorozódni: $p_{\text{üzenet}} \leftarrow p_{\text{üzenet}} \cdot p[k]$

Kódolás algoritmus (egyszerűsített)

Be: üzenet

az üzenet k jeleinek ABC-sorrendbe rakása

$p[k]$ valószínűségek meghatározása

$a[k]$ alsó határok meghatározása
 $\text{alsó} \leftarrow 0$ # kezdőintervallum alsó határa
 $p_{\text{üzenet}} \leftarrow 1$ # kezdőintervallum hossza
 Ciklus az összes k jelre az üzenetben:
 $\text{alsó} \leftarrow \text{alsó} + p_{\text{üzenet}} * a[k]$
 $p_{\text{üzenet}} \leftarrow p_{\text{üzenet}} * p[k]$
 $\text{felső} \leftarrow \text{alsó} + p_{\text{üzenet}}$
 Ki: $k, \text{alsó}, \text{felső}$
 Ciklus vége
 Ki: kód \leftarrow egy szám a legutolsó intervallumból

Kódolás algoritmusának rövidebb változónevekkel

Be: üzenet
 az üzenet k jeleinek ABC-sorrendbe rakása
 $p[k]$ valószínűségek meghatározása
 $a[k]$ alsó határok meghatározása
 $\text{ah} \leftarrow 0$ # kezdőintervallum alsó határa
 $p_{\text{ü}} \leftarrow 1$ # kezdőintervallum hossza
 Ciklus az összes k jelre az üzenetben:
 $\text{ah} \leftarrow \text{ah} + p_{\text{ü}} * a[k]$
 $p_{\text{ü}} \leftarrow p_{\text{ü}} * p[k]$
 $f \leftarrow \text{ah} + p_{\text{ü}}$
 Ki: k, ah, f
 Ciklus vége
 Ki: kód \leftarrow egy szám a legutolsó intervallumból

Egyes jelek kódjai

A $[0; 1[$ intervallumot osztjuk fel az egyes jelek között. A jeleket ABC-rendbe (adott kódlap szerinti rendbe) rakjuk, és mindegyiknek a valószínűségével egyező nagyságú intervallum jut az egységnyi hosszúságú intervallumon.

Az ALMA illetve MIKKAMAKKA üzenetek esetén itt láthatóak az egyes k jelekhez tartozó p_k valószínűségek és a_k alsó határok.

k	A	L	M
p_k	1/2	1/4	1/4
a_k	0	1/2	3/4

k	A	I	K	M
p_k	0,3	0,1	0,4	0,2
a_k	0	0,3	0,4	0,8

Jelsorozat kódja

Az aritmetikai kódolásnál a $[0; 1[$ intervallumot szűkítgetjük az egymás utáni jelekkel.

M	→ [0,8	; 1	[
MI	→ [0,86	; 0,88	[
MIK	→ [0,868	; 0,876	[
MIKK	→ [0,8712	; 0,8744	[
MIKKA	→ [0,8712	; 0,87216	[
MIKKAM	→ [0,871968	; 0,87216	[
MIKKAMA	→ [0,871968	; 0,8720256	[
MIKKAMAK	→ [0,87199104	; 0,87201408	[
MIKKAMAKK	→ [0,872000256	; 0,872009472	[
MIKKAMAKKA	→ [0,872000256	; 0,8720030208	[
MIKKAMAKKA	→ 0,872002		

Aritmetikai kód: kódolt üzenet visszafejtése

Be: jelek és valószínűségek (k, p[k])

a[k] alsó határok kiszámítása

Be: kód # az üzenet kódja

Be: hossz # az üzenet hossza

Ciklus hossz-szor:

k ← a jel, aminek az intervallumában van a kód

Ki: k

kód ← (kód - a[k]) / p[k]

Ciklus vége

Visszafejtés

```
0,872002 M
0,360010 I
0,600100 K
0,500250 K
0,250625 A
0,835417 M
0,177083 A
0,590278 K
0,475694 K
0,189236 A
```

Egy gyakorlati alkalmazás

A képek JPEG kódolásánál választható az aritmetikai és Huffman-kódolás az adatok egyik tömörítési lépéséhez.

A Huffman-kódolás gyorsabb, de a tömörítés mértéke elmarad az aritmetikai kódoláshoz képest.

Lackfi János: Parabola

Kitekintés

Parabola, parabola
antenna,
nézzünk tévét
éppen ma!

Parabola, parabola
futbalmeccs,
lassított gól,
sípcsont reccs!

Parabola, parabola
sminkreklám.
Bőröd fittyed?
Kend ezt rá!

Parabola, parabola
bankrablók,
lő, fut, robban,
égből lóg.

Parabola, parabola
popcsillag,
rázós ritmus
popsidnak.

Parabola, parabola
antenna,
úrlény caplat
álmodba!

Ezt már azért nagyobb falat lenne kódolni, de erre lesznek hatékonyabb módok is, amelyek kihasználják, hogy a szövegekre nem igaz az IRD-modell, és sok ismétlődő jelsorozat van benne. Ezek lesznek a szótár alapú tömörítések.