Efstratios Demertzoglou - TH20580

Thursday, March 13, 2025 4:32 PM

Άσκηση 1

$$H(x) = -\sum_{i=1}^{n} P(x_i) \cdot \log_2 P(x_i)$$

Στην περίπτωσή μας: $P=0.5$ (είτε μαύρο είτε άσπρο)

$$H(x) = -(0.5 \log_2 0.5 + 0.5 \cdot \log_2 0.5) = -(0.5 \cdot (-1) + 0.5(-1)) = 1 \text{ bit per pixel}$$

Άσκηση 2

A:(1) B:(1) C:(1) D:(2) E:(3) F:(5) G:(5) H:(10)

H: 1
G: 01
F: 00
E: 001
D 0001
C 00001
B 000001
A: 000000

Steps:
1) Δημιουργούμε σειρά προτεραιότητας με κάθε χαρακτήρα ως κόμβο φύλλου
2) Συνδυάζουμε τους δύο κόμβους χαμηλότερης συχνότητας επαναληπτικά για να δημιουργήσουμε το δέντρο
3) Αντιστοίχιση στον αριστερό κλάδο και στα δεξιά 0 1

Άσκηση 3

{0, 20, 50, 99}

99	99	99	99	99	99	99	99
20	20	20	20	20	20	20	20
0	0	0	0	0	0	0	0
0	0	50	50	50	50	0	0
0	0	50	50	50	50	0	0
0	0	50	50	50	50	0	0
0	0	50	50	50	50	0	0
0	0	0	0	0	0	0	0

$P(99) = 8/64 = 1/8 = 0.125$
$P(20) = 8/64 = 1/8 = 0.125$
$P(50) = 16/64 = 2/8 = 1/4 = 0.25$
$P(0) = 32/64 = 4/8 = 1/2 = 0.5$

$H(x) = -\sum P(x_i) \cdot \log_2 P(x_i)$ $\quad \log_2(1/8) = -3 \quad \log_2(1/4) = -2 \quad \log_2(1/2) = -1$

$\Rightarrow H(x) = -(0.125(-3) + 0.125(-3) + 0.25 \cdot (-2) + 0.5(-1)) = -(-0.75 - 0.5 - 0.5) = 1.75$ bits/pixel

b) Merge(99, 20) → New node (0.125 + 0.125 = 0.25)

Merge(50, New node) → Newer node (0.25 + 0.25 = 0.5)

Merge (0, Newer node) → Final Node (0.5 + 0.5 = 1)

Άρα: 99 → 000 20 → 001 50 → 01 0 → 1

c) $L = 3/8 + 3/8 + 2/4 + 1/2 = 1.75$ bits per pixel

d) $\text{Efficiency} = \dfrac{\text{Entropy}}{\text{Huffman Length}} \cdot 100\% = \dfrac{1.75}{1.75} \cdot 100\% = 100\%$

Άσκηση 6

a) $P(0) = 7/8 \qquad P(1) = 1/8$

$H(x) = -\left(\frac{7}{8}\log_2(7/8) + \frac{1}{8}\log_2(1/8)\right) = -\left(\frac{7}{8}(-0.2) + \frac{1}{8}(-3)\right) = 0.175 + 0.375 = 0.55$ bits/symbol

b) $0 \to 0$ & $1 \to 1$ αφού είναι binary source average bitrate $\to$ 1 bit/symbol

c) Ζευγάρια 00, 01, 10, 11

00	$\to$ 0
01	$\to$ 10
10	$\to$ 110
11	$\to$ 111

$P(00) = P(0) \cdot P(0) = 49/64 \simeq 0.77$

$P(01) = P(10) = P(1) \cdot P(0) = 7/64 \simeq 0.11$

$P(11) = P(1) \cdot P(1) = 1/64 \simeq 0.016$

Average bitrate: $L = 0.77 + 0.11 \cdot 2 + 0.11 \cdot 3 + 0.016 \cdot 4 = 1.36$ bits per 2-symbol pair

$\Rightarrow 0.68$ bits per symbol

Άρα πιο αποτελεσματικό το extended Huffman coding από το κανονικό

Άσκηση 7

Πλεονεκτήματα της αριθμητικής κωδικοποίησης:

1. **Καλύτερη συμπίεση:**
   - Μπορεί να επιτύχει συμπίεση πιο κοντά στην εντροπία εκχωρώντας κλασματικά μήκη bit.
2. **Αποτελεσματικό για ασύμμετρες πιθανότητες:**
   - Λειτουργεί καλά όταν οι πιθανότητες συμβόλων ποικίλλουν σημαντικά.
3. **Δεν χρειάζονται προκαθορισμένοι κωδικοί:**
   - Δεν χρειάζεται να κατασκευάσετε ένα δέντρο Huffman εκ των προτέρων.

Μειονεκτήματα της αριθμητικής κωδικοποίησης:

1. **Βραδύτερη κωδικοποίηση/αποκωδικοποίηση:**
   - Πιο υπολογιστικά εντατική από την κωδικοποίηση Huffman.
2. **Χωρίς στοίχιση byte:**
   - Οι κώδικες Huffman λειτουργούν καλά με συστήματα προσανατολισμένα σε byte, ενώ η αριθμητική κωδικοποίηση μπορεί να απαιτεί ειδικό χειρισμό.
3. **Πιθανά ζητήματα διπλωμάτων ευρεσιτεχνίας:**
   - Ιστορικά, η αριθμητική κωδικοποίηση αντιμετώπιζε νομικούς περιορισμούς.

Arithmetic:

A: [0, 0.05)

B: [0.5, 0.9)

C: [0.9, 1.0)

Βήμα 1ο $\to$ [0, 1)

B $\to$ [0.5, 0.9)

BB $\to$ [0.5 + 0.4(0.9 − 0.5), 0.5 + 0.9(0.4)]

$\to$ [0.66, 0.86]

BBB $\to$ [0.66 + 0.4(0.86 − 0.66), 0.66 + 0.9(0.2)]

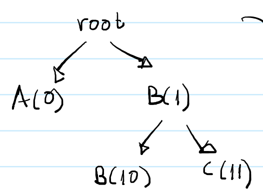
$\to$ [0.74, 0.84]

3.32 bps · 3 ≃ 10 bits

Huffman

$P(A) = 0.5$, $P(B) = 0.4$, $P(C) = 0.1$

$\to$ (BC, 0.5)



A = 0

B = 10

C = 11

BBB: 10 10 10 σύνολο 6-bit

Huffman: 6 bit

Arithmetic: 10 bit

Άσκηση 9

Βήμα 1: Εκχώρηση εύρους

A	:	$P(A) = 0.8$	$[0.00, 0.80)$
B	:	$P(B) = 0.02$	$[0.80, 0.82)$
C	:	$P(C) = 0.18$	$[0.82, 1.00)$

Βήμα 2 Κωδικοποίηση ACBA

Πρώτο σύμβολο "A" : εύρος εκκίνησης $[0, 1]$

πέφτει σε $[0, 0.8)$

νέο εύρος → $[0, 0.8]$

"C" : πέφτει στο $[0.82, 1)$ εύρος του προηγούμενου εύρους

νέο εύρος → $[0 + 0.8 \cdot 0.82,\ 0 + 0.8 \cdot 1] = [0.656,\ 0.8]$

"B" : πέφτει σε $[0.8, 0.82)$

νέο εύρος → $[0.656 + (0.8 - 0.656) \cdot 0.8,\ 0.656 + (0.8 - 0.656) \cdot 0.82] = [0.786, 0.7882]$

"A": Το εύρος παραμένει $[0.786, 0.7882]$

Άσκηση 12

"0" → 0 , "1" → 1

step	input	update	output
1	0	–	0
2	01	"01" → 2	1
3	1	–	1
4	10	"10" → 3	3
5	0	–	0
6	00	"00" → 4	0
7	11	"11" → 5	1

Encoded Output: [0, 1, 1, 3, 0, 0, 1]

Final Dictionary: {"0": 0, "1": 1, "01": 2, "10": 3, "00": 4, "11": 5}