LINKÖPING UNIVERSITY

Solutions to Written Exam in **Data compression** **TSBK08**

27th August 2022

1 a) See the course literature.

b) See the course literature.

c) See the course literature.

d) See the course literature.

e) See the course literature.

2 If X takes values in the alphabet $\{a_1, a_2, \ldots, a_L\}$ the entropy is given by

$$H(X) = -\sum_{i=1}^{L} p(a_i) \cdot \log p(a_i)$$

The left inequality comes from

$$-p(a_i) \cdot \log p(a_i) \begin{cases} = 0, & p(a_i) = 0 \\ > 0, & 0 < p(x_i) < 1 \\ = 0, & p(a_i) = 1 \end{cases}$$

Thus $H(X) \geq 0$ with equality if and only if $p(a_i)$ is either 0 or 1 for every i, but then we must have that $p(a_i) = 1$ for exactly one i.

The right inequality comes from

$$\begin{aligned} H(X) - \log L &= -\sum_{i=1}^{L} p(a_i) \log p(a_i) - \log L \\ &= \sum_{i=1}^{L} p(a_i) \log \frac{1}{L \cdot p(a_i)} \\ &\leq \sum_{i=1}^{L} p(a_i) (\frac{1}{L \cdot p(a_i)} - 1) \log e \\ &= (\sum_{i=1}^{L} \frac{1}{L} - \sum_{i=1}^{L} p(a_i)) \log e \\ &= (1-1) \log e = 0 \end{aligned}$$

with equality if and only if $p(a_i) = \frac{1}{L}$ for all $i = 1, \ldots, L$. This inequality can also be proven by regular Lagrange minimization techniques.

3 A Huffman code for the distribution gives the mean codeword length $\bar{l} = 2.64$ bits/codeword and average data rate $R = \bar{l} = 2.64$ bits/symbol.

For comparison, the entropy rate of the source is $H(X_n) \approx 2.5819$.

4 a) The stationary distribution of the Markov source is

$$\bar{w} = [w_a \;\; w_b \;\; w_c] = [\frac{23}{66} \;\; \frac{18}{66} \;\; \frac{25}{66}]$$

This gives us the memoryless entropy

$$H(X_n) = -\frac{23}{66} \cdot \log \frac{23}{66} - \frac{18}{66} \cdot \log \frac{18}{66} - \frac{25}{66} \cdot \log \frac{25}{66} \approx 1.5717$$

and the conditional entropy

$$\begin{aligned} H(X_n|X_{n-1}) &= \frac{23}{66} \cdot (-0.75 \cdot \log 0.75 - 0.15 \cdot \log 0.15 - 0.1 \cdot \log 0.1) + \\ &\quad \frac{18}{66} \cdot (-0.25 \cdot \log 0.25 - 0.6 \cdot \log 0.6 - 0.15 \cdot \log 0.15) + \\ &\quad \frac{25}{66} \cdot (-0.05 \cdot \log 0.05 - 0.15 \cdot \log 0.15 - 0.8 \cdot \log 0.8) \approx \\ &\approx 1.0712 \end{aligned}$$

Finally, the chain rule gives us

$$\begin{aligned} H(X_n, X_{n+1}, X_{n+2}) &= H(X_n) + H(X_{n+1}|X_n) + H(X_{n+2}|X_n, X_{n+1}) \\ &= H(X_n) + 2 \cdot H(X_{n+1}|X_n) \\ &\approx 3.7140 \end{aligned}$$

where we used the fact that the source is of order one and stationary.

b) Under the assumption that the subintervals are always ordered in the same order as in the alphabet, we will get the interval [0.48751875 0.49359375) with size 0.006075. (Simple check: $0.75 \cdot 0.75 \cdot 0.15 \cdot 0.6 \cdot 0.15 \cdot 0.8 = 0.006075$).

We will need at least $\lceil -\log_2 0.006075 \rceil = 8$ bits in our codeword, maybe one more.

Write the two limits as binary numbers:

$$\begin{aligned} 0.48751875 &= 0.01111100110011\ldots \\ 0.49359375 &= 0.01111110010111\ldots \end{aligned}$$

The smallest number with eight bits in the interval is 0.01111101. All numbers that start with these bits are also inside the interval (ie smaller than the upper limit). Eight bits will therefore be enough.

The codeword is thus **01111101**.

5 Inverse mtf gives the vector $L = [o\ o\ p\ p\ p\ o\ m\ m]$.

Sort the sequence to get the vector $F = [m\ m\ o\ o\ o\ p\ p\ p]$ which gives us the vector $T = [6\ 7\ 0\ 1\ 5\ 2\ 3\ 4]$ (the position in L where you find each symbol in F).

Inverse BWT gives the sequence *pompompo*.

6 a) If we only use fixed length codes, we need $\log_2 512 = 9$ bits for the offset and $\log_2 16 = 4$ bits to code a symbol. Including

the flag bit, we thus need 1+9+4=14 bits to code a match and 1+4=5 bits to code a single symbol. It is thus better to code matches of length 1 and 2 as single symbols. Given 4 bits for the length we can thus use the match lengths 3-18 instead of the lengths 1-16.

If flag 0 is used to indicate a single symbol and flag 1 is used to indicate a match, we get the codewords ($< 0, c >$ or $< 1, o, l >$):

$< 0, k >$ $< 0, a >$ $< 0, n >$ $< 1, 2, 5 >$ $< 0, p >$ $< 1, 7, 5 >$ $< 0, b >$ $< 1, 0, 3 >$ $< 1, 10, 4 >$ $\ldots$

The number of bits used to code the first 22 symbols of the sequence is 5+5+5+14+5+14+5+14+14 = 81.

b) The decoded sequence is

$$gagagagamigamigogogoge \ldots$$

and the dictionary looks like

index	word	index	word	index	word	index	word
0	a	8	i	16	ga	24	mig
1	b	9	j	17	ag	25	go
2	c	10	k	18	gag	26	og
3	d	11	l	19	$gaga$	27	gog
4	e	12	m	20	am	28	$goge$
5	f	13	n	21	mi	29	$e*$
6	g	14	o	22	ig		
7	h	15	p	23	gam		

where the last symbol of word 29 isn't known until we have decoded the next index.