GUC Berlin
German University in Cairo
الجامعة الألمانية بالقاهرة

German University in Cairo - GUC
Faculty of Information Engineering & Technology - IET
Department of Communication Engineering
Dr.-Ing. Heiko Schwarz

COMM901 – Source Coding and Compression
Winter Semester 2013/2014

Midterm Exam

Bar Code

Instructions: **Read Carefully Before Proceeding.**

1- Non-programmable calculators are allowed
2- Write your solutions in the space provided
3- The exam consists of **4 questions**
4- This exam booklet contains **13 pages** including this page
5- Total time allowed for this exam is **(120) minutes**
6- When you are told that time is up, stop working on the test

Good Luck!

Question	**1**	**2**	**3**	**4**	**∑**
Possible Marks	21	20	20	23	80 + 4 bonus
Final Marks					

Question 1: Coding of IID Sources (21 Marks)

Given is a stationary discrete random process $\boldsymbol{S} = \{S_n\}$ with independent and identically distributed discrete random variables S_n. The alphabet $\mathcal{A}_S = \{a_i\}$ for the random variables S_n and the associated probability mass function $p(a_i) = P(S_n = a_i)$ are given in the table below.

(a) In the following, three codes are given. State for each of these codes whether the code is uniquely decodable (for arbitrarily long messages) and whether the code is a prefix code (i.e., it is instantaneously decodable). Write your answers ("yes" or "no") directly into the table. **[6 Marks]**

Solution:

a_i	$p(a_i)$	code A	code B	code C
$=$	0.70	1	00	1
$+$	0.15	10	01	00
$-$	0.05	100	10	010
$\times$	0.05	1000	110	011
$\div$	0.05	10000	111	100
uniquely decodable ?		yes **[1]**	yes **[1]**	no **[1]**
prefix code ?		no **[1]**	yes **[1]**	no **[1]**

(b) Assign a binary codeword to each element a_i of the alphabet $\mathcal{A}_S$ in a way that the resulting code is ***uniquely decodable*** for arbitrary long messages and the ***smallest possible average codeword length*** per symbol is obtained. Write the codewords directly into the table. **[4 Marks]**

Solution:

a_i	$p(a_i)$	codeword
$=$	0.70	0
$+$	0.15	10
$-$	0.05	110
$\times$	0.05	1110
$\div$	0.05	1111

[4]

(c) Determine the average codeword length $\bar{\ell}$, in bit per symbol, for the code you developed in sub-question (b). Do not round the result. **[2 Marks]**

Solution:

The average codeword length $\bar{\ell}$ is given by

$$\begin{aligned}\bar{\ell} &= \sum_{\mathrm{i=0}}^{4} p(a_i) \cdot \ell(a_i) \quad [\mathbf{1}] \\ &= 0.70 \cdot 1 + 0.15 \cdot 2 + 0.05 \cdot (3 + 4 + 4) \\ &= 0.70 + 0.30 + 0.55 \\ &= 1.55 \quad \text{bit per symbol}\end{aligned}$$

The average codeword length $\bar{\ell}$ for the developed code is 1.55 bit per symbol. **[1]**

(d) Determine the entropy rate $\bar{H}(\boldsymbol{S})$, in bit per symbol, for the given source $\boldsymbol{S}$. The final result should be stated with three digits after the decimal point (for intermediate results a higher precision may be required). **[3 Marks]**

Solution:

Since the source represents an iid process, the entropy rate $\bar{H}(\boldsymbol{S})$ is equal to the marginal entropy $H(S_n)$.

Hence, the entropy rate is given by

$$\begin{aligned}\bar{H}(\boldsymbol{S}) &= H(S_n) \quad [\mathbf{1}] \\ &= -\sum_{i=0}^{4} p(a_i) \cdot \log_2 p(a_i) \quad [\mathbf{1}] \\ &= -0.70 \cdot \log_2 0.70 - 0.15 \cdot \log_2 0.15 - 3 \cdot 0.05 \cdot \log_2 0.05 \\ &\approx 0.360201 + 0.410545 + 3 \cdot 0.216096 \\ &\approx 1.419 \quad \text{bit per symbol}\end{aligned}$$

The entropy rate $\bar{H}(\boldsymbol{S})$ for the given source is 1.419 bit per symbol. **[1]**

(e) In the following, we consider the coding of long messages (more than 10 000 symbols) for the given source. State ***two practical lossless coding techniques*** by which the coding efficiency can be increased relative to that of the code developed in (b). An explanation/justification is not required. **[2 Marks]**

Solution:

The coding efficiency can be increased by:

- **Block Huffman coding**: **[1]**
  Codewords are assigned to groups of N successive symbols. Unless all probability masses for an iid process are negative integer powers of 2 (which is not the case), increasing the size N of the symbol sequences to which codewords are assigned, decreases the average codeword length per symbol.

- **Arithmetic Coding (fixed-precision variant of Elias coding)**: **[1]**
  Codewords are iteratively constructed based on the joint probability mass function. When ignoring the impact of rounding, the average codeword length for coding N symbols of an iid process is guaranteed to be less than $\bar{H}(\boldsymbol{S}) + 1/N$. The impact of rounding is typically very small.

Note: Conditional Huffman coding does not increase the efficiency, since we consider an iid source.

(f) Consider the coding of long messages for the given source. What is the maximum ***percentage of average bit rate that can be saved*** if the code you developed in (b) is replaced by a more efficient lossless coding technique? State the result in percent, with a precision of two digits after the decimal point. **[4 Marks]**

Hint: *Re-use results that you have already derived.*

Solution:

The greatest lower bound for the average codeword length for lossless coding of very long messages is given by the entropy rate $\bar{H}(\boldsymbol{S})$ of the source. **[1]**

For iid sources and long messages, this bound can be at least asymptotically achieved (e.g., by using high-precision arithmetic coding). Hence, the maximum relative amount of bit rate that can be saved on average is given by

$$\varrho = \frac{\bar{\ell} - \bar{H}(\boldsymbol{S})}{\bar{\ell}} = 1 - \frac{\bar{H}(\boldsymbol{S})}{\bar{\ell}} \qquad [2]$$

$$\approx 1 - \frac{1.419}{1.55} \approx 0.0845 \approx 8.45\ \%$$

At maximum 8.45% of the bit rate can be saved on average if the developed code is replaced by a more efficient lossless coding technique. **[1]**

Question 2: Conditional Huffman Coding (20 Marks)

Given is a stationary Markov process $\boldsymbol{S} = \{S_n\}$ with the ternary symbol alphabet $\mathcal{A}_S = \{x, y, z\}$. The conditional symbol probabilities $p(s_n|s_{n-1}) = P(S_n = s_n|S_{n-1} = s_{n-1})$ are given in the left table below. Due to the symmetry of the transition probabilities, the marginal probability mass function is uniform; the corresponding marginal symbol probabilities $p(s_n) = P(S_n = s_n)$ are given in the right table below.

s_n	$p(s_n\|s_{n-1} = x)$	$p(s_n\|s_{n-1} = y)$	$p(s_n\|s_{n-1} = z)$
x	3/4	1/8	1/8
y	1/8	3/4	1/8
z	1/8	1/8	3/4

s_n	$p(s_n)$
x	1/3
y	1/3
z	1/3

(a) Develop a conditional Huffman code for which the chosen code table for a symbol s_n depends on the value of the preceding symbol s_{n-1}. Write the codewords directly into the table below. **[6 Marks]**

Solution:

current symbol s_n	codewords (depending on previous symbol s_{n-1})		
	$s_{n-1} = x$	$s_{n-1} = y$	$s_{n-1} = z$
x	0	10	10
y	10	0	11
z	11	11	0
	[2]	**[2]**	**[2]**

(b) Now, we want to use the developed conditional Huffman code for encoding a message. For the first symbol s_0 of the message we do not have a preceding symbol s_{-1}. Here, we use the convention $s_{-1} = x$, i.e., we always use the first codeword table (the one for $s_{n-1} = x$) for the first symbol.
Write down the bitstream for the symbol sequence "$xyyz$". **[4 Marks]**

Solution:

The bitstream is created by concatenating the codewords for all symbols of the symbol sequence "$xyyz$". In the following, the selected tables ([a] stands for "$s_{n-1} = a$") and the chosen codewords are shown:

"[x]0 [x]10 [y]0 [y]11"

Hence, the resulting bitstream is

"010011" **[4]**

(c) Now, we want to use the developed conditional Huffman code for decoding. As in the previous sub-question, we use the convention $s_{-1} = x$, i.e., we use the codeword table for $s_{n-1} = x$ for the first symbol. Given is a bitstream that starts with "$00110011 \cdots$". Decode the first 4 symbols. **[4 Marks]**

Solution:

The decoding process can be described as follows:

1st symbol:	table for "$s_{n-1} = x$";	codeword "0";	symbol "x"
2nd symbol:	table for "$s_{n-1} = x$";	codeword "0";	symbol "x"
3rd symbol:	table for "$s_{n-1} = x$";	codeword "11";	symbol "z"
4th symbol:	table for "$s_{n-1} = z$";	codeword "0";	symbol "z"

The first four symbols of the symbol sequence represented by the bitstream are

"$xxzz$" **[4]**

(d) Determine the average codeword length, in bit per symbol, for the developed conditional Huffman code (for coding very long messages). Do not round the result. **[6 Marks]**

<u>Hint:</u> *There are at least two different ways for determine the average codeword length. It might be advantageous to start with determining the average codeword lengths for the three different conditions:* $\bar{\ell}(s_{n-1} = x)$, $\bar{\ell}(s_{n-1} = y)$ *and* $\bar{\ell}(s_{n-1} = z)$.

<u>Solution:</u>

For all three possible conditions "$s_{n-1} = x$", "$s_{n-1} = y$", and "$s_{n-1} = z$", we have the same set of probability masses and associated codeword lengths. Hence, all three conditional average codeword lengths $\bar{\ell}(s_{n-1} = x)$, $\bar{\ell}(s_{n-1} = y)$ and $\bar{\ell}(s_{n-1} = z)$ are the same. **[2]**

With b representing x, y, or z, we obtain

$$\begin{aligned}\bar{\ell}(s_{n-1} = b) &= \sum_{i=0}^{2} p(a_i|b) \cdot \ell(a_i|b) \quad [\mathbf{1}] \\ &= \frac{3}{4} \cdot 1 + \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 2 = \frac{3+1+1}{4} \\ &= \frac{5}{4} = 1.25 \quad \text{bit per symbol} \quad [\mathbf{1}]\end{aligned}$$

The average codeword length $\bar{\ell}$ is then given by the expectation value of the conditional average codeword lengths $\bar{\ell}(s_{n-1} = b)$. Hence, we have

$$\begin{aligned}\bar{\ell} &= \sum_{i=0}^{2} p(b_i) \cdot \bar{\ell}(s_{n-1} = b_i) \quad [\mathbf{1}] \\ &= \bar{\ell}(s_{n-1} = b) \cdot \left(\sum_{i=0}^{2} p(b_i) \right) = \bar{\ell}(s_{n-1} = b) \\ &= \frac{5}{4} = 1.25 \quad \text{bit per symbol}\end{aligned}$$

The average codeword length $\bar{\ell}$ is 1.25 bit per symbol. **[1]**

Question 3: Elias and Arithmetic Coding (20 Marks)

Given is a Bernoulli process (binary iid process). The symbol alphabet is given by $\mathcal{A}_S = \{x, o\}$.
The probability of the symbol x is $p(x) = P(S_n = x) = 0.1$.
For Elias coding, which we consider in the following, we use the convention "$x < o$".

(a) Determine the Elias codeword for the symbol sequence "oxo". **[15 Marks]**

Solution:

The following table shows the pmf $p(s_n)$ as well as the cmf $c(s_n)$ excluding the current symbol.

s_n	$p(s_n)$	$c(s_n)$
x	0.1	0.0
o	0.9	0.1

[2]

We use the iterative encoding procedure:

- Initialization:

$$W_0 = 1 \quad \textbf{[1]}$$
$$L_0 = 0 \quad \textbf{[1]}$$

- Iteration for the first symbol "o":

$$W_1 = W_0 \cdot p(o) = 1 \cdot 0.9 = 0.9 \quad \textbf{[1]}$$
$$L_1 = L_0 + W_0 \cdot c(o) = 0 + 1 \cdot 0.1 = 0.1 \quad \textbf{[1]}$$

- Iteration for the second symbol "x":

$$W_2 = W_1 \cdot p(x) = 0.9 \cdot 0.1 = 0.09 \quad \textbf{[1]}$$
$$L_2 = L_1 + W_1 \cdot c(x) = 0.1 + 0.9 \cdot 0.0 = 0.1 \quad \textbf{[1]}$$

- Iteration for the third symbol "o":

$$W_3 = W_2 \cdot p(o) = 0.09 \cdot 0.9 = 0.081 \quad \textbf{[1]}$$
$$L_3 = L_2 + W_2 \cdot c(o) = 0.1 + 0.09 \cdot 0.1 = 0.109 \quad \textbf{[1]}$$

- Termination:

$$K = \lceil -\log_2 W_3 \rceil = \lceil -\log_2 0.081 \rceil = \lceil 3.63 \rceil = 4 \quad \textbf{[2]}$$
$$v = \lceil L_3 \cdot 2^K \rceil \cdot 2^{-K} = \lceil 0.109 \cdot 16 \rceil \cdot 2^{-4} = \lceil 1.744 \rceil \cdot 2^{-4} = 2 \cdot 2^{-4} = 2^{-3}$$
$$= 0.0010_{(2)} \quad \textbf{[2]}$$

- The codeword is given by the first K bits after the binary point of the binary representation of v.

The Elias codeword for the symbol sequence "oxo" is "0010". **[1]**

(b) Elias coding provides the desirable property that the codewords for arbitrarily long symbol sequences can be iteratively constructed. For coding long symbol sequences and using the correct probabilities, the resulting coding efficiency is typically very close to the entropy rate. However, Elias coding is not used in practice. Instead, arithmetic coding is used in several applications.

Provide brief answers to the following questions: **[5 Marks]**

- Why is Elias coding not used in practice?
- How is arithmetic coding related to Elias coding?
- What are the main differences between Elias coding and arithmetic coding?

Solution:

In particular for long symbol sequences, Elias coding cannot be applied in practice due to the extremely large precision that would be required for calculating the interval width and lower interval boundary. **[1]**

Arithmetic coding is a variant of Elias coding that can be implemented with machine-precision integer arithmetic. **[1]**

The main differences to Elias coding are:

- In arithmetic coding, all probabilities are represented by fixed-precision numbers. **[1]**
- In arithmetic coding, the interval width is represented by a fixed-precision number. **[1]**
  In each iteration step, the interval width is down-rounded to meet the fixed-precision requirement.
- In arithmetic coding, due to the fixed-precision representation of the interval width, the bits of the binary representation of the lower interval boundary that can be modified in following iteration steps are represented by a fixed-precision number and a counter. **[1]**
  The bits of the binary representation of the lower interval boundary that cannot be modified in future iteration step are output as soon as possible (this can also be done in Elias coding).

Question 4: Bounds and Comparison of Lossless Coding Techniques (23 Marks)

Given is a stationary Markov process $\boldsymbol{S} = \{S_n\}$ with the binary symbol alphabet $\mathcal{A}_S = \{a_i\} = \{x, y\}$. The conditional symbol probabilities $p(s_n|s_{n-1}) = P(S_n = s_n|S_{n-1} = s_{n-1})$ are shown in the left table below.

s_n	$p(s_n\|s_{n-1} = x)$	$p(s_n\|s_{n-1} = y)$
x	0.8	0.2
y	0.2	0.8

s_n	codeword
x	0
y	1

Consider the transmission of long messages for this source. Given is a simple code, which is shown in the right table above; it assigns a bit equal to 0 to the symbol "x" and a bit equal to 1 to the symbol "y".

(a) With how many bits is a symbol on average represented using the given code? **[2 Marks]**

Solution:

Since we assign a codeword of 1 bit to each symbol, it is obvious that the average codeword length is one bit per symbol. This can also be shown by

$$\bar{\ell} = p(x) \cdot 1 + p(y) \cdot 1 = \big(p(x) + p(y)\big) \cdot 1 = 1 \cdot 1 = 1$$

On average, a source symbol is represented with one bit. **[2]**

(b) State the marginal probability masses $p(x) = P(S_n = x)$ and $p(y) = P(S_n = y)$. **[3 Marks]**

> Hint: *You can directly calculate the marginal probability masses. It may also be possible to derive them without any calculation (but then you should briefly explain how you did that).*

Solution:

Since both transition probabilities are the same, $p(x|y) = p(y|x) = 0.2$, and are unequal to zero, due to reasons of symmetry, we can directly conclude $p(x) = p(y) = 0.5$. **[3]**

This can also be shown as follows

$$\begin{aligned} p(x) &= p(x,x) + p(x,y) = p(x|x) \cdot p(x) + p(x|y) \cdot p(y) \\ &= 0.8 \cdot p(x) + 0.2 \cdot \big(1 - p(x)\big) \\ p(x) \cdot (1 - 0.8 + 0.2) &= 0.2 \\ \Longrightarrow \quad p(x) &= \frac{0.2}{0.4} = \frac{1}{2} \\ \Longrightarrow \quad p(y) &= 1 - p(x) = 1 - \frac{1}{2} = \frac{1}{2} \end{aligned}$$

(c) Determine the marginal entropy $H(S_n)$, in bit per symbol, for the given source. **[2 Marks]**

Solution:

The marginal entropy $H(S_n)$ is given by

$$\begin{aligned} H(S_n) &= -\sum_{i=0}^{1} p(a_i) \cdot \log_2 p(a_i) \qquad [1] \\ &= -p(x) \cdot \log_2 p(x) - p(y) \cdot \log_2 p(y) \\ &= -2 \cdot \frac{1}{2} \cdot \log_2 \frac{1}{2} = \log_2 2 \\ &= 1 \quad \text{bit per symbol} \end{aligned}$$

The marginal entropy $H(S_n)$ is 1 bit per symbol. **[1]**

(d) Let $H(S_n)$ denote the marginal entropy and let $H(S_n|S_{n-1})$ denote the conditional entropy given the last symbol. How are the marginal entropy $H(S_n)$ and the conditional entropy $H(S_n|S_{n-1})$ related for the given stationary Markov source? Insert the correct relational sign ("<" or "=" or ">") into the following expression and briefly explain your choice. **[2 Marks]**

Solution:

$$H(S_n|S_{n-1}) \quad < \quad H(S_n) \qquad [1]$$

The condition

$$H(S_n|S_{n-1}) \le H(S_n)$$

is always true. Conditioning does not increase the uncertainty about a random variable. Equality is only achieved if the random variables S_n and S_{n-1} are independent, i.e., if $p(s_n|s_{n-1}) = p(s_n)$, which is not the case for the given source. **[1]**

(e) Let $H(S_n)$ denote the marginal entropy, let $H(S_n|S_{n-1})$ denote the conditional entropy given the last symbol, and let $\bar{H}(\boldsymbol{S})$ denote the entropy rate for a source $\boldsymbol{S} = \{S_n\}$. Furthermore let $\bar{\ell}$ denote the average codeword length per symbol for some ***unknown lossless coding technique***.
Now, we consider three different sources:

- an iid source (independent and identically distributed random variables)
- a stationary Markov source (which is not iid)
- a general stationary source with memory, which does not fulfil the Markov property.

For each of the three sources, indicate in the following table (by writing "true" or "false" into the table cells), which of the statements are always true. Note that multiple statements can be true for some or for all sources. An explanation/justification is not required. **[6 Marks]**

Hints: *Read the question carefully and think carefully!*

Solution:

statement	iid source	stationary Markov source	general stationary source (not Markov)	
$\bar{\ell} \ge H(S_n)$	true	false	false	**[1.5]**
$\bar{\ell} \ge H(S_n|S_{n-1})$	true	true	false	**[1.5]**
$\bar{\ell} \ge \bar{H}(\boldsymbol{S})$	true	true	true	**[1.5]**
$\bar{\ell} < H(S_n) + 1$	false	false	false	**[1.5]**

(f) By which of the following lossless coding techniques is it possible to increase the coding efficiency for the given source in comparison to the simple code given at the beginning of this question?
State for each of the techniques that are listed below whether an increase in coding efficiency can be obtained and briefly justify/explain your statement. **[8 Marks]**

Hints: *Think carefully about the techniques and keep in mind what source we consider.*

Solution:

Conditional Huffman coding: A codeword is assigned to each symbol s_n; the selection of the codeword table for a symbol s_n is done on the basis of the previous symbol s_{n-1}.

Since the source is a binary source and all conditional probability masses are greater than zero, a conditional Huffman code assigns exactly one bit to each symbol. Hence, the average codeword length is equal to 1 bit per symbol. Thus, no increase in coding efficiency can be obtained. **[2]**

Block Huffman coding: A message is partitioned into blocks of constant size N, with $N \geq 2$, and a codeword is assigned to each possible block "$s_n s_{n+1} \cdots s_{n+N-1}$".

With block Huffman coding, an increase of the block size always leads to an improved coding efficiency, except the entropy rate is already achieved. Since the conditional probability masses are not the same, the entropy rate is less than 1. Hence, an increase of the coding efficiency is possible for the given source. **[2]**

Marginal arithmetic coding: The arithmetic coding is done using the marginal symbol probabilities, which are given by $p(s_n) = P(S_n = s_n)$.

The minimum achievable average codeword length per symbol is given by the marginal entropy. Hence, we have $\bar{\ell} \geq H(S_n) = \log_2 2 = 1$ bit per symbol. An increase of the coding efficiency is not possible. **[2]**

Conditional arithmetic coding: The arithmetic coding is done using the conditional symbol probabilities, which are given by $p(s_n|s_{n-1}) = P(S_n = s_n|S_{n-1} = s_{n-1})$.

For the coding of long messages, it is possible to achieve an average codeword length per symbol that is very close to the conditional entropy $H(S_n|S_{n-1})$. Since the conditional symbol probabilities are not the same, the conditional entropy $H(S_n|S_{n-1})$ is less than 1. Hence, an increase of the coding efficiency is possible for the given source. **[2]**