Information Theory

Chapter3: Source Coding

Rudolf Mathar

Lehrstuhl für Theoretische Informationstechnik
RWTH AACHEN UNIVERSITY

WS 2018/19

Outline Chapter 2: Source Coding

Variable Length Encoding

Prefix Codes

Kraft-McMillan Theorem

Average Code Word Length

Noiseless Coding Theorem

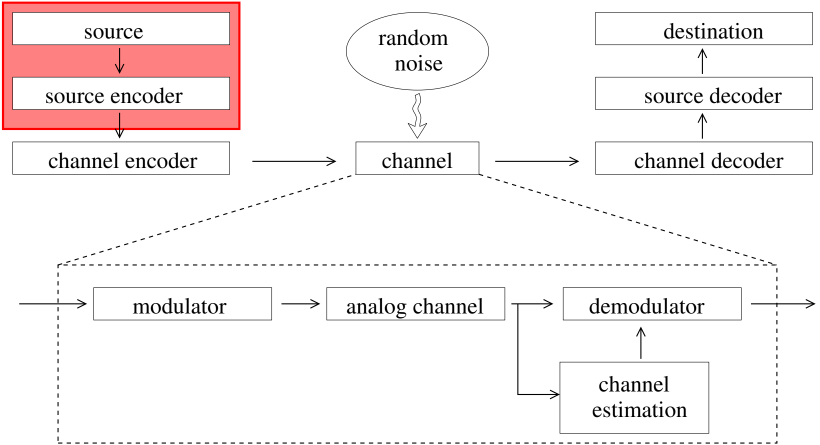
Huffman Coding

Block Codes for Stationary Sources

Arithmetic Coding

Communication Channel

from an information theoretic point of view



Variable Length Encoding

Given some
source alphabet $\mathcal{X} = \{x_1, \ldots, x_m\}$,
code alphabet $\mathcal{Y} = \{y_1, \ldots, y_d\}$.

Aim:
For each character $x_1, \ldots, x_m$ find a code word formed over $\mathcal{Y}$.

Formally:
Map each character $x_i \in \mathcal{X}$ uniquely onto a "word" over $\mathcal{Y}$.

Definition 3.1.
An injective mapping

$$g : \mathcal{X} \to \bigcup_{\ell=0}^{\infty} \mathcal{Y}^\ell : x_i \mapsto g(x_i) = (w_{i1}, \ldots, w_{in_i})$$

is called *encoding*. $g(x_i) = (w_{i1}, \ldots, w_{in_i})$ is called *code word* of character x_i, n_i is called *length* of code word i.

Variable Length Encoding

Example:

	g_1	g_2	g_3	g_4
a	1	1	0	0
b	0	10	10	01
c	1	100	110	10
d	00	1000	111	11
	no encoding	encoding, words are separable	encoding, shorter, words separable	encoding, even shorter, not separable

Hence, separability of concatenated words over $\mathcal{Y}$ is important.

Variable Length Encoding

Definition 3.2.
An encoding g is called *uniquely decodable (u.d.)* or *uniquely decipherable*, if the mapping

$$G : \bigcup_{\ell=0}^{\infty} \mathcal{X}^{\ell} \to \bigcup_{\ell=0}^{\infty} \mathcal{Y}^{\ell} : (a_1, \dots, a_k) \mapsto (g(a_1), \dots, g(a_k))$$

is injectiv.

Example:
Use the previous encoding g_3

	g_3
a	0
b	10
c	110
d	111

1 1 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0
1 1 1|1 0 0 0 1 1 0 1 1 1 0 0 0 1 0
1 1 1|1 0 |0 0 1 1 0 1 1 1 0 0 0 1 0
1 1 1|1 0 |0|0 |1 1 0|1 1 1|0| 0|0|1 0
d b a a c d a a a b

(g_3 is a so called prefix code)

Prefix Codes

Definition 3.3.
A code is called *prefix code*, if no complete code word is prefix of some other code word, i.e., no code word evolves from continuing some other.

Formally:
$\boldsymbol{a} \in \mathcal{Y}^k$ is called prefix of $\boldsymbol{b} \in \mathcal{Y}^l$, $k \leq l$, if there is some $c \in \mathcal{Y}^{l-k}$ such that $\boldsymbol{b} = (\boldsymbol{a}, \boldsymbol{c})$.

Theorem 3.4.
Prefix codes are uniquely decodable.

More properties:

- Prefix codes are easy to construct based on the code word lengths.
- Decoding of prefix codes is fast and requires no memory storage.

Next aim: characterize uniquely decodable codes by their code word lengths.

Kraft-McMillan Theorem

Theorem 3.5. (a) McMillan (1959), b) Kraft (1949))

a) All uniquely decodable codes with code word lengths $n_1, \dots, n_m$ satisfy

$$\sum_{j=1}^{m} d^{-n_j} \leq 1$$

b) Conversely, if $n_1, \dots, n_m \in \mathbb{N}$ are such that $\sum_{j=1}^{m} d^{-n_j} \leq 1$, then there exists a u.d. code (even a prefix code) with code word lengths $n_1, \dots, n_m$.

Example:

	g_3	g_4
a	0	0
b	10	01
c	110	10
d	111	11
	u.d.	not u.d.

For g_3: $2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 1$

For g_4:

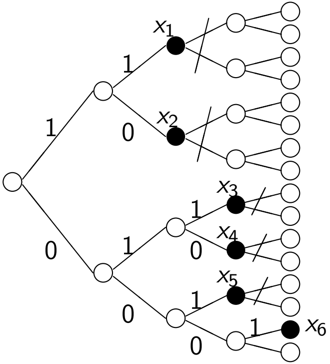
$$2^{-1} + 2^{-2} + 2^{-2} + 2^{-2} = 5/4 > 1$$

g_4 is not u.d., there is no u.d. code with code word lengths 1,2,2,2.

RWTH AACHEN UNIVERSITY

Kraft-McMillan Theorem, Proof of b)

Assume $n_1 = n_2 = 2,\ n_3 = n_4 = n_5 = 3,\ n_6 = 4.$
Then $\sum i = 1^6 = 15/16 < 1$
Construct a prefix code by a binary code tree as follows.



The corresponding code is given as

x_i	x_1	x_2	x_3	x_4	x_5	x_6
$g(x_i)$	11	10	011	010	001	0001

Average Code Word Length

Given a code $g(x_1), \dots, g(x_m)$ with code word lengths $n_1, \dots, n_m$.
Question: What is a reasonable measure of the "length of a code"?

Definition 3.6.
The *expected code word length* is defined as

$$\bar{n} = \bar{n}(g) = \sum_{j=1}^{m} n_j p_j = \sum_{j=1}^{m} n_j P(X = x_j)$$

Example:

	p_i	g_2	g_3
a	1/2	1	0
b	1/4	10	10
c	1/8	100	110
d	1/8	1000	111
$\bar{n}(g)$		15/8	14/8
$H(X)$	14/8		

Noiseless Coding Theorem, Shannon (1949)

Theorem 3.7.
Let random variable X describe a source with distribution $P(X = x_i) = p_i$, $i = 1, \dots, m$. Let the code alphabet $\mathcal{Y} = \{y_1, \dots, y_d\}$ have size d.

a) Each u.d. code g with code word lengths $n_1, \dots, n_m$ satisfies

$$\bar{n}(g) \geq H(X) / \log d.$$

b) Conversely, there is a prefix code, hence a u.d. code g with

$$\bar{n}(g) \leq H(X) / \log d + 1.$$

Proof of a)

For any u.d. code it holds by McMillan's Theorem that

$$
\begin{aligned}
\frac{H(X)}{\log d} - \bar{n}(g) &= \frac{1}{\log d} \sum_{j=1}^{m} p_j \log \frac{1}{p_j} - \sum_{j=1}^{m} p_j n_j \\
&= \frac{1}{\log d} \sum_{j=1}^{m} p_j \log \frac{1}{p_j} + \sum_{j=1}^{m} p_j \frac{\log d^{-n_j}}{\log d} \\
&= \frac{1}{\log d} \sum_{j=1}^{m} p_j \log \frac{d^{-n_j}}{p_j} \\
&= \frac{\log e}{\log d} \sum_{j=1}^{m} p_j \ln \frac{d^{-n_j}}{p_j} \\
&\leq \frac{\log e}{\log d} \sum_{j=1}^{m} p_j \left(\frac{d^{-n_j}}{p_j} - 1 \right) \\
&\leq \frac{\log e}{\log d} \sum_{j=1}^{m} \left(d^{-n_j} - p_j \right) \leq 0
\end{aligned}
$$

Proof of b) Shannon-Fano Coding

W.l.o.g. assume that $p_j > 0$ for all j.

Choose integers n_j such that $d^{-n_j} \leq p_j < d^{-n_j+1}$ for all j.
Then

$$\sum_{j=1}^{m} d^{-n_j} \leq \sum_{j=1}^{m} p_j \leq 1$$

such that by Kraft's Theorem a u.d. code g exists. Furthermore,

$$\log p_j < (-n_j + 1) \log d$$

holds by construction. Hence

$$\sum_{j=1}^{m} p_j \log p_j < (\log d) \sum_{j=1}^{m} p_j(-n_j + 1),$$

equivalently,

$$H(X) > (\log d) \left(\bar{n}(g) - 1\right).$$

Compact Codes

Is there always a u.d. code g with

$$\bar{n}(g) = H(X)/\log d?$$

No! Check the previous proof. Equality holds if and only if $p_j = 2^{-n_j}$ for all $j = 1, \ldots, m$.

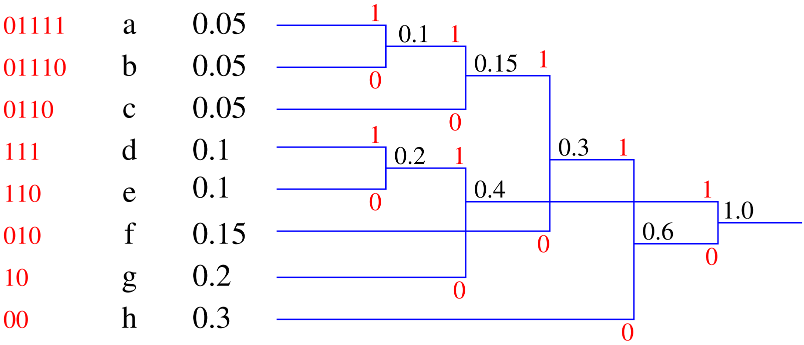
Example. Consider binary codes, i.e., $d = 2$. $\mathcal{X} = \{a, b\}$, $p_1 = 0.6,\ p_2 = 0.4$. The shortest possible code is $g(a) = (0),\ g(b) = (1)$.

$$\begin{aligned} H(X) &= -0.6 \log_2 0.6 - 0.4 \log_2 0.4 = 0.97095 \\ \bar{n}(g) &= 1. \end{aligned}$$

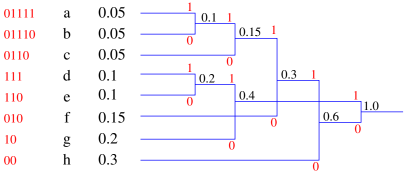
Definition 3.8.
Any code of shortest possible average code word length is called *compact*.

How to construct compact codes?

Huffman Coding



Huffman Coding



A compact code g^* is given by:

Character:	a	b	c	d	e	f	g	h
Code word:	01111	01110	0110	111	110	010	10	00

It holds (log to the base 2):

$$\bar{n}(g^*) = 5 \cdot 0.05 + \cdots + 2 \cdot 0.3 = \mathbf{2.75}$$
$$H(X) = -0.05 \cdot \log_2 0.05 - \cdots - 0.3 \cdot \log_2 0.3 = \mathbf{2.7087}$$

Block Codes for Stationary Sources

Encode blocks/words of length N by words over the code alphabet $\mathcal{Y}$.
Assume that blocks are generated by a stationary source, a stationary sequence of random variables $\{X_n\}_{n\in\mathbb{N}}$.
Notation for a block code:

$$g^{(N)} : \mathcal{X}^N \to \bigcup_{\ell=0}^{\infty} \mathcal{Y}^\ell$$

Block codes are "normal" variabel length codes over the extended alphabet $\mathcal{X}^N$.

A fair measure of the "length" of a block code is the average code word length per character

$$\bar{n}\big(g^{(N)}\big)/N.$$

The lower Shannon bound, namely the entropy of the source, is asymptotically ($N \to \infty$) attained by suitable block codes, as is shown in the following.

Noiseless Coding Theorem for Block Codes

Theorem 3.9.
Let $\boldsymbol{X} = \{X_n\}_{n \in \mathbb{N}}$ be a stationary source. Let the code alphabet $\mathcal{Y} = \{y_1, \dots, y_d\}$ have size d.

a) Each u.d. block code $g^{(N)}$ satisfies

$$\frac{\bar{n}(g^{(N)})}{N} \geq \frac{H(X_1, \dots, X_N)}{N \, \log d}.$$

b) Conversely, there is a prefix block code, hence a u.d. block code $g^{(N)}$ with

$$\frac{\bar{n}(g^{(N)})}{N} \leq \frac{H(X_1, \dots, X_N)}{N \, \log d} + \frac{1}{N}.$$

Hence, in the limit as $N \to \infty$:
There is a sequence of u.d. block codes $g^{(N)}$ such that

$$\lim_{N \to \infty} \frac{\bar{n}(g^{(N)})}{N} = \frac{H_\infty(\boldsymbol{X})}{\log d}.$$

Huffman Block Coding

In principle, Huffman encoding can be applied to block codes. However, problems include

- The size of the Huffman table is m^N, thus growing exponentially with the block length.
- The code table needs to be transmitted to the receiver.
- The source statistics are assumed to be stationary. No adaptivity to to changing probabilities.
- Encoding and decoding only per block. Delays occur at the beginning and end. Padding may be necessary.

"Arithmetic coding" avoids these shortcomings.

Arithmetic Coding

Assume that

- Message $(x_{i_1}, \ldots, x_{i_N})$, $x_{i_j} \in \mathcal{X}$, $j = 1, \ldots, N$ is generated by some source $\{X_n\}_{n \in \mathbb{N}}$.
- All (conditional) probabilities

$$P(X_n = x_{i_n} \mid X_1 = x_{i_1}, \ldots, X_{n-1} = x_{i_{n-1}}) = p(i_n \mid i_1, \ldots, i_{n-1}),$$

$x_{i_1}, \ldots, x_{i_n} \in \mathcal{X}$, $n = 1, \ldots, N$, are known to the encoder and decoder, or can be estimated.

Then,

$$P(X_1 = x_{i_1}, \ldots, X_n = x_{i_n}) = p(i_1, \ldots, i_n)$$

can be easily computed as

$$p(i_1, \ldots, i_n) = p(i_n \mid i_1, \ldots, i_{n-1}) \cdot p(i_1, \ldots, i_{n-1})$$

Arithmetic Coding

Iterately construct intervals

Initialization, $n = 1$: $\big(c(1) = 0,\ c(m+1) = 1\big)$

$$I(j) = \big[c(j), c(j+1)\big), \quad c(j) = \sum_{i=1}^{j-1} p(i),\ j = 1, \dots, m$$

(cumulative probabilities)

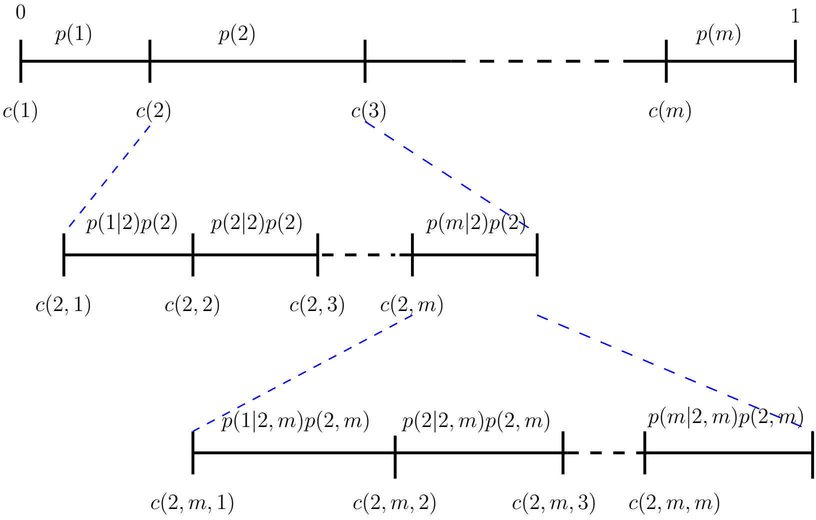
Recursion over $n = 2, \dots, N$:

$$\begin{aligned} &I(i_1, \dots, i_n) \\ &= \Big[c(i_1, \dots, i_{n-1}) + \sum_{i=1}^{i_n - 1} p(i_n \mid i_1, \dots, i_{n-1}) \cdot p(i_1, \dots, i_{n-1})\Big) \\ &\qquad c(i_1, \dots, i_{n-1}) + \sum_{i=1}^{i_n} p(i_n \mid i_1, \dots, i_{n-1}) \cdot p(i_1, \dots, i_{n-1})\Big) \end{aligned}$$

Program code available from Togneri, deSilva, p. 151, 152

Arithmetic Coding

Example.



Arithmetic Coding

Encode message $(x_{i_1}, \ldots, x_{i_N})$ by the binary representation of some binary number in the interval $I(i_1, \ldots, i_n)$.

A scheme which usually works quite well is as follows.
Let $l = l(i_1, \ldots, i_n)$ and $r = r(i_1, \ldots, i_n)$ denote the left and right bound of the corresponding interval. Carry out the binary expansion of l and r until until they differ. Since $l < r$, at the first place they differ there will be a 0 in the expansion of l and a 1 in the expansion of r. The number $0.a_1 a_2 \ldots a_{t-1} 1$ falls within the interval and requires the least number of bits.

$$(a_1 a_2 \ldots a_{t-1} 1) \text{ is the encoding of } (x_{i_1}, \ldots, x_{i_N}).$$

The probability of occurrence of message $(x_{i_1}, \ldots, x_{i_N})$ is equal to the length of the representing interval. Approximately

$$-\log_2 p(i_1, \ldots, i_n)$$

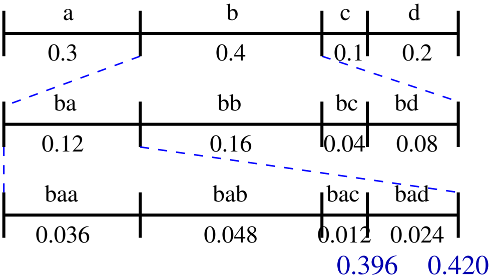
bits are needed to represent the interval, which is close to optimal.

Arithmetic Coding

Example. Assume a memoryless source with 4 characters and probabilities

x_i	a	b	c	d
$P(X_n = x_i)$	0.3	0.4	0.1	0.2

Encode the word (bad):



$$(bad) = [0.396, 0.42)$$

$$0.396 = 0.01100\ldots \qquad 0.420 = 0.01101\ldots$$

$$(bad) = (01101)$$