Digital Signal Processing for Music

Part 23: Source Coding

alexander lerch

source coding
introduction 1/3

- typical audio **bit rates**

$$16\,\mathrm{bit} \cdot 44100\,\mathrm{sps} \cdot 2\,\mathrm{chan} = 1411.2\,\mathrm{kbps}$$
$$24\,\mathrm{bit} \cdot 192000\,\mathrm{sps} \cdot 5\,\mathrm{chan} = 23040\,\mathrm{kbps}$$

- **reasons** for bit rate reduction
  - economical reasons: cheaper transmission/storage
  - technical reasons: restricted storage/transmission bandwidth
- **applications** for source coding
  - Internet: streaming, distribution, peer-2-peer, VoIP, ...
  - Media: DVD-V/A, ...
  - Portable Devices: MP3-Player, cell phones, Mini-Disc, ...
  - Broadcasting: (Digital) Radio, TV, ...
  - Cinema: DD, DTS, SDDS
  - ...

source coding
introduction 1/3

- typical audio **bit rates**

$$16\,\text{bit} \cdot 44100\,\text{sps} \cdot 2\,\text{chan} = 1411.2\,\text{kbps}$$
$$24\,\text{bit} \cdot 192000\,\text{sps} \cdot 5\,\text{chan} = 23040\,\text{kbps}$$

- **reasons** for bit rate reduction
  - economical reasons: cheaper transmission/storage
  - technical reasons: restricted storage/transmission bandwidth
- **applications** for source coding
  - Internet: streaming, distribution, peer-2-peer, VoIP, ...
  - Media: DVD-V/A, ...
  - Portable Devices: MP3-Player, cell phones, Mini-Disc, ...
  - Broadcasting: (Digital) Radio, TV, ...
  - Cinema: DD, DTS, SDDS
  - ...

source coding
introduction 1/3

- typical audio **bit rates**

$$
\begin{aligned}
16\,\mathrm{bit} \cdot 44100\,\mathrm{sps} \cdot 2\,\mathrm{chan} &= 1411.2\,\mathrm{kbps} \\
24\,\mathrm{bit} \cdot 192000\,\mathrm{sps} \cdot 5\,\mathrm{chan} &= 23040\,\mathrm{kbps}
\end{aligned}
$$

- **reasons** for bit rate reduction
  - economical reasons: cheaper transmission/storage
  - technical reasons: restricted storage/transmission bandwidth
- **applications** for source coding
  - Internet: streaming, distribution, peer-2-peer, VoIP, ...
  - Media: DVD-V/A, ...
  - Portable Devices: MP3-Player, cell phones, Mini-Disc, ...
  - Broadcasting: (Digital) Radio, TV, ...
  - Cinema: DD, DTS, SDDS
  - ...

source coding

introduction 2/3

How can the bitrate be reduced

How can the bitrate be reduced

1. **lossless**:
   remove *redundant* information (unnecessary to reconstruct the signal)
   - entropy coding
   - (linear predictive coding)

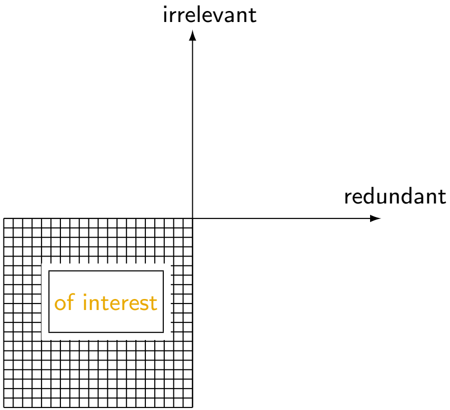
How can the bitrate be reduced

1. **lossless**:
   remove *redundant* information (unnecessary to reconstruct the signal)
   - entropy coding
   - (linear predictive coding)

2. **lossy**:
   remove *irrelevant* information (not "missed" by the recipient)
   - waveform coding
   - perceptual coding

source coding

introduction 3/3



source coding
fundamentals: definitions

note: words to be transmitted are referred to as *symbols*

information content

The less frequent a symbol, the higher its *information content*, *self-information*, *surprisal*.

$$I_n = \log_2\left(\frac{1}{p_n}\right)$$

entropy

The entropy is the *Expected Value* of the information content. It is the *theoretic minimum of bits* required for transmission.

$$H = \sum_{n=0}^{N-1} p_n \cdot I_n$$

source coding

fundamentals: definitions

note: words to be transmitted are referred to as *symbols*

information content

The less frequent a symbol, the higher its *information content*, *self-information*, *surprisal*.

$$I_n = \log_2\left(\frac{1}{p_n}\right)$$

entropy

The entropy is the *Expected Value* of the information content. It is the *theoretic minimum of bits* required for transmission.

$$H = \sum_{n=0}^{N-1} p_n \cdot I_n$$

source coding
fundamentals: definitions

note: words to be transmitted are referred to as *symbols*

information content

The less frequent a symbol, the higher its *information content*, *self-information*, *surprisal*.

$$I_n = \log_2\left(\frac{1}{p_n}\right)$$

entropy

The entropy is the *Expected Value* of the information content. It is the *theoretic minimum of bits* required for transmission.

$$H = \sum_{n=0}^{N-1} p_n \cdot I_n$$

source coding

fundamentals: information content and entropy examples

- **dice**: $p_n = \frac{1}{6}$

$$
\begin{aligned}
I_n &= \log_2\left(\frac{1}{p_n}\right) = 2.58\ bit \\
H &= 6 \cdot \frac{1}{6} \cdot 2.58\ bit
\end{aligned}
$$

source coding

fundamentals: information content and entropy examples

- **dice**: $p_n = \frac{1}{6}$

$$\begin{aligned} I_n &= \log_2\left(\frac{1}{p_n}\right) = 2.58\ bit \\ H &= 6 \cdot \frac{1}{6} \cdot 2.58\ bit \end{aligned}$$

- **imperfect dice**: $p_1 = \frac{1}{2},\ p_{2\ldots6} = \frac{1}{10}$

source coding

fundamentals: information content and entropy examples

- **dice**: $p_n = \frac{1}{6}$

$$\begin{aligned} I_n &= \log_2\left(\frac{1}{p_n}\right) = 2.58\ bit \\ H &= 6 \cdot \frac{1}{6} \cdot 2.58\ bit \end{aligned}$$

- **imperfect dice**: $p_1 = \frac{1}{2},\ p_{2\dots6} = \frac{1}{10}$

$$\begin{aligned} I_1 &= \log_2(2) &&= 1\ bit \\ I_{2\dots6} &= \log_2(10) &&= 3.32\ bit \\ H &= \tfrac{1}{2}\cdot 1 + \tfrac{5}{10}\cdot 3.32 &&= 2.16\ bit \end{aligned}$$

source coding

entropy coding: example 1

idea: use shorter words for frequent symbols

- 3 possible symbols

symbol	probability	word
A	$p = 0.5$	
B	$p = 0.25$	
C	$p = 0.25$	

- entropy

$$H = \sum_{n=0}^{N-1} p_n \log_2 \left(\frac{1}{p_n}\right) = 1.5$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ optimal transmission

source coding

entropy coding: example 1

idea: use shorter words for frequent symbols

- 3 possible symbols

symbol	probability	word
A	$p = 0.5$	
B	$p = 0.25$	
C	$p = 0.25$	

- entropy

$$H = \sum_{n=0}^{N-1} p_n \log_2\left(\frac{1}{p_n}\right) = 1.5$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ optimal transmission

source coding

entropy coding: example 1

idea: use shorter words for frequent symbols

- 3 possible symbols

symbol	probability	word
A	$p = 0.5$	0
B	$p = 0.25$	10
C	$p = 0.25$	11

- entropy

$$H = \sum_{n=0}^{N-1} p_n \log_2\left(\frac{1}{p_n}\right) = 1.5$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ optimal transmission

source coding
entropy coding: example 1

idea: use shorter words for frequent symbols

- 3 possible symbols

symbol	probability	word
A	$p = 0.5$	0
B	$p = 0.25$	10
C	$p = 0.25$	11

- entropy

$$H = \sum_{n=0}^{N-1} p_n \log_2\left(\frac{1}{p_n}\right) = 1.5$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ optimal transmission

source coding

entropy coding: example 1

idea: use shorter words for frequent symbols

- 3 possible symbols

symbol	probability	word
A	$p = 0.5$	0
B	$p = 0.25$	10
C	$p = 0.25$	11

- entropy

$$H = \sum_{n=0}^{N-1} p_n \log_2\left(\frac{1}{p_n}\right) = 1.5$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ optimal transmission

source coding
entropy coding: example 1

idea: use shorter words for frequent symbols

- 3 possible symbols

symbol	probability	word
A	$p = 0.5$	0
B	$p = 0.25$	10
C	$p = 0.25$	11

- entropy

$$H = \sum_{n=0}^{N-1} p_n \log_2\left(\frac{1}{p_n}\right) = 1.5$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ optimal transmission

source coding

entropy coding: example 2

- 3 possible symbols

symbol	probability	word
A	$p = 0.7$	
B	$p = 0.2$	
C	$p = 0.1$	

- entropy

$$H = \sum_{n=0}^{N} p_n \log_2\left(\frac{1}{p_n}\right) = 1.11$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ *non*-optimal transmission

source coding

entropy coding: example 2

- 3 possible symbols

symbol	probability	word
A	$p = 0.7$	0
B	$p = 0.2$	10
C	$p = 0.1$	11

- entropy

$$H = \sum_{n=0}^{N} p_n \log_2\left(\frac{1}{p_n}\right) = 1.11$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ *non*-optimal transmission

source coding

entropy coding: example 2

- 3 possible symbols

symbol	probability	word
A	$p = 0.7$	0
B	$p = 0.2$	10
C	$p = 0.1$	11

- entropy

$$H = \sum_{n=0}^{N} p_n \log_2 \left(\frac{1}{p_n}\right) = 1.11$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\mathit{transmitted\ bits}}{\mathit{transmitted\ symbols}} = \frac{6}{4} = 1.5$$

$\Rightarrow$ *non*-optimal transmission

source coding

entropy coding: example 2

- 3 possible symbols

symbol	probability	word
A	$p = 0.7$	0
B	$p = 0.2$	10
C	$p = 0.1$	11

- entropy

$$H = \sum_{n=0}^{N} p_n \log_2\left(\frac{1}{p_n}\right) = 1.11$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{transmitted\ bits}{transmitted\ symbols} = \frac{6}{4} = 1.5$$

$\Rightarrow$ *non*-optimal transmission

source coding
entropy coding: example 2

- 3 possible symbols

symbol	probability	word
A	$p = 0.7$	0
B	$p = 0.2$	10
C	$p = 0.1$	11

- entropy

$$H = \sum_{n=0}^{N} p_n \log_2\left(\frac{1}{p_n}\right) = 1.11$$

- transmit the following group of symbols: $ABCA \rightarrow$ 010110
- required bits:

$$\frac{\textit{transmitted bits}}{\textit{transmitted symbols}} = \frac{6}{4} = 1.5$$

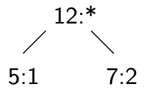
$\Rightarrow$ *non*-optimal transmission

source coding
huffman coding: tree construction 1/2

sort symbols acc. to frequency

frequency	symbol
5	1
7	2
10	3
15	4
20	5
45	6

combine two lowest symbols into new entry (sum)



add new entry to list

frequency	symbol
10	3
12	*
15	4
20	5
45	6

repeat until only one element left in the list

source coding

huffman coding: tree construction 2/2

5:1 7:2 10:3 15:4 20:5 45:6

source coding
huffman coding: tree construction 2/2

5:1 7:2 10:3 15:4 20:5 45:6

source coding

huffman coding: tree construction 2/2

10:3 12:* 15:4 20:5 45:6

5:1 7:2

source coding

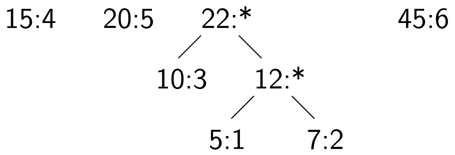
huffman coding: tree construction 2/2

10:3 12:* 15:4 20:5 45:6

5:1 7:2

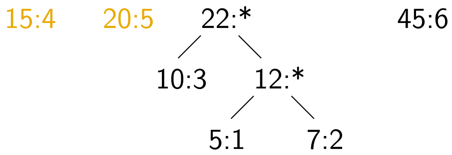
source coding

huffman coding: tree construction 2/2



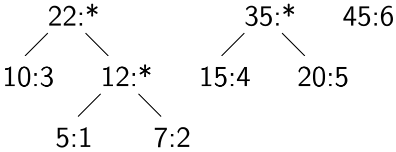
source coding

huffman coding: tree construction 2/2



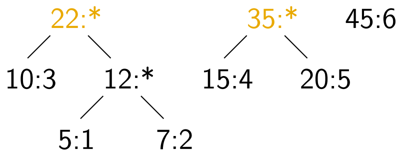
source coding

huffman coding: tree construction 2/2



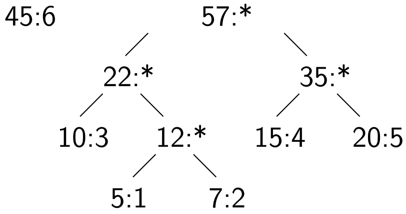
source coding

huffman coding: tree construction 2/2



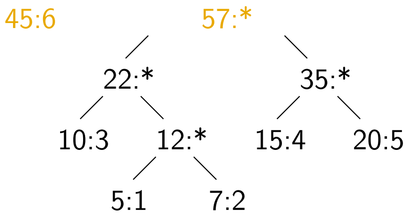
source coding

huffman coding: tree construction 2/2



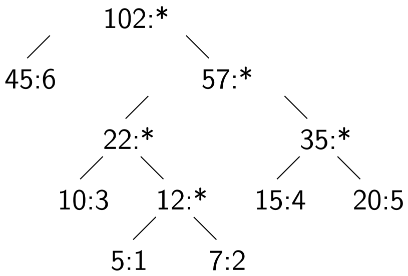
source coding

huffman coding: tree construction 2/2



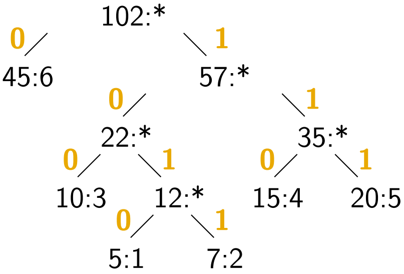
source coding

huffman coding: tree construction 2/2



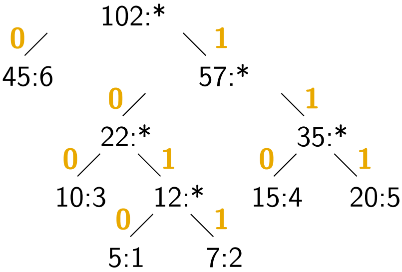
source coding

huffman coding: tree construction 2/2



source coding

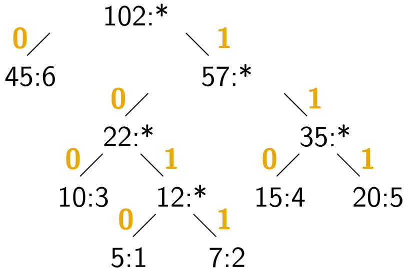
huffman coding: tree construction 2/2



frequency	**symbol**	**code**
5	1	1010
7	2	1011
10	3	100
15	4	110
20	5	111
45	6	0

source coding

huffman coding: tree construction 2/2

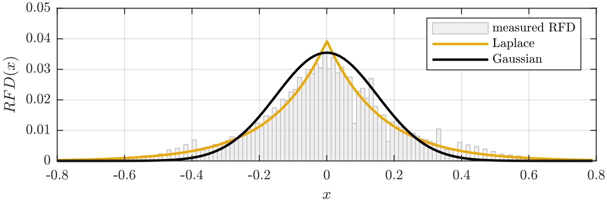


frequency	**symbol**	**code**
5	1	1010
7	2	1011
10	3	100
15	4	110
20	5	111
45	6	0

note: no code is prefix of another code!

source coding
huffman coding for audio signals

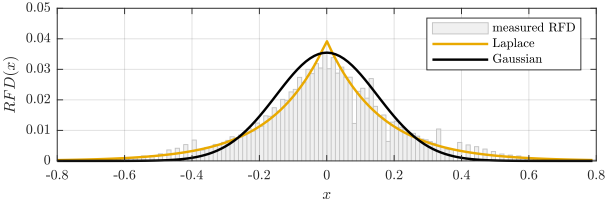
- Symbole: 2^w
- PDF indicates probability per symbol



matlab source: plotRfd.m

source coding
huffman coding for audio signals

- Symbole: 2^w
- PDF indicates probability per symbol



matlab source: plotRfd.m

source coding

arithmetic coding

- **Huffman coding is only optimal if $p_n = \frac{1}{2^k}$**

- alternative: **arithmetic coding**
    - allows other probability distributions
    - encodes the whole sequence in one fractional number $0.0 \leq f < 1.0$
    - **principle**:
        1. assume initial interval of $[0, 1[$
        2. assign interval segments to all symbols, e.g. $A = [0, 0.7[$, $B = [0.7, 0.9[$, $C = [0.9, 1[$
        3. select interval based on current symbol
        4. go to Step 2. or terminate

source coding
arithmetic coding

- **Huffman coding is only optimal if $p_n = \frac{1}{2^k}$**
- alternative: **arithmetic coding**
  - allows other probability distributions
  - encodes the whole sequence in one fractional number $0.0 \leq f < 1.0$
  - **principle**:
    1. assume initial interval of $[0, 1[$
    2. assign interval segments to all symbols, e.g. $A = [0, 0.7[$, $B = [0.7, 0.9[$, $C = [0.9, 1[$
    3. select interval based on current symbol
    4. go to Step 2. or terminate

source coding
arithmetic coding

- **Huffman coding is only optimal if $p_n = \frac{1}{2^k}$**
- alternative: **arithmetic coding**
  - allows other probability distributions
  - encodes the whole sequence in one fractional number $0.0 \leq f < 1.0$
  - **principle**:
    1. assume initial interval of $[0, 1[$
    2. assign interval segments to all symbols, e.g. $A = [0, 0.7[, B = [0.7, 0.9[, C = [0.9, 1[$
    3. select interval based on current symbol
    4. go to Step 2. or terminate

source coding
arithmetic coding

- **Huffman coding is only optimal if $p_n = \frac{1}{2^k}$**
- alternative: **arithmetic coding**
  - allows other probability distributions
  - encodes the whole sequence in one fractional number $0.0 \leq f < 1.0$
  - **principle**:
    1. assume initial interval of $[0, 1[$
    2. assign interval segments to all symbols, e.g. $A = [0, 0.7[$, $B = [0.7, 0.9[$, $C = [0.9, 1[$
    3. select interval based on current symbol
    4. go to Step 2. or terminate

source coding
arithmetic coding

- **Huffman coding is only optimal if $p_n = \frac{1}{2^k}$**
- alternative: **arithmetic coding**
    - allows other probability distributions
    - encodes the whole sequence in one fractional number $0.0 \leq f < 1.0$
    - **principle**:
        1. assume initial interval of $[0, 1[$
        2. assign interval segments to all symbols, e.g. $A = [0, 0.7[, B = [0.7, 0.9[, C = [0.9, 1[$
        3. select interval based on current symbol
        4. go to Step 2. or terminate

source coding
arithmetic coding

- **Huffman coding is only optimal if $p_n = \frac{1}{2^k}$**
- alternative: **arithmetic coding**
  - allows other probability distributions
  - encodes the whole sequence in one fractional number $0.0 \leq f < 1.0$
  - **principle**:
    1. assume initial interval of $[0, 1[$
    2. assign interval segments to all symbols, e.g. $A = [0, 0.7[$, $B = [0.7, 0.9[$, $C = [0.9, 1[$
    3. select interval based on current symbol
    4. go to Step 2. or terminate

source coding

arithmetic coding: example 1/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6$, $p_{\mathrm{B}} = 0.2$, $p_{\mathrm{C}} = 0.1$, $p_{\mathrm{T}} = 0.1$,

$A = [0, 0.6[$, $B = [0.6, 0.8[$, $C = [0.8, 0.9[$, $T = [0.9, 1[$

- **decoding** 0.463:
  1. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0, 0.6[$ $\rightarrow$ bounds: $0, 0.36, 0.48, 0.54, 0.6$
  2. $0.463 \in$ segment 2 ($\rightarrow B$),
     - set interval $[0.36, 0.48[$ $\rightarrow$ bounds: $0.36, 0.432, 0.456, 0.468, 0.48$
  3. $0.463 \in$ segment 3 ($\rightarrow C$),
     - set interval $[0.456, 0.468[$ $\rightarrow$ bounds: $0.456, 0.4632, 0.4656, 0.4668, 0.468$
  4. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0.456, 0.4632[$ $\rightarrow$ bounds: $0.456, 0.46032, 0.46176, 0.46248, 0.4632$
  5. $0.463 \in$ segment 4 ($\rightarrow$ terminate)

source coding

arithmetic coding: example 1/2

sequence $ABCA$, $p_\mathrm{A} = 0.6$, $p_\mathrm{B} = 0.2$, $p_\mathrm{C} = 0.1$, $p_\mathrm{T} = 0.1$,
$A = [0, 0.6[$, $B = [0.6, 0.8[$, $C = [0.8, 0.9[$, $T = [0.9, 1[$

- **decoding** 0.463:
  1. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0, 0.6[$ $\rightarrow$ bounds: $0, 0.36, 0.48, 0.54, 0.6$
  2. $0.463 \in$ segment 2 ($\rightarrow B$),
     - set interval $[0.36, 0.48[$ $\rightarrow$ bounds: $0.36, 0.432, 0.456, 0.468, 0.48$
  3. $0.463 \in$ segment 3 ($\rightarrow C$),
     - set interval $[0.456, 0.468[$ $\rightarrow$ bounds: $0.456, 0.4632, 0.4656, 0.4668, 0.468$
  4. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0.456, 0.4632[$ $\rightarrow$ bounds: $0.456, 0.46032, 0.46176, 0.46248, 0.4632$
  5. $0.463 \in$ segment 4 ($\rightarrow$ terminate)

source coding

arithmetic coding: example 1/2

sequence $ABCA$, $p_\mathrm{A} = 0.6$, $p_\mathrm{B} = 0.2$, $p_\mathrm{C} = 0.1$, $p_\mathrm{T} = 0.1$,
$A = [0, 0.6[$, $B = [0.6, 0.8[$, $C = [0.8, 0.9[$, $T = [0.9, 1[$

- **decoding** 0.463:
  1. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0, 0.6[\rightarrow$ bounds: $0, 0.36, 0.48, 0.54, 0.6$
  2. $0.463 \in$ segment 2 ($\rightarrow B$),
     - set interval $[0.36, 0.48[\rightarrow$ bounds: $0.36, 0.432, 0.456, 0.468, 0.48$
  3. $0.463 \in$ segment 3 ($\rightarrow C$),
     - set interval $[0.456, 0.468[\rightarrow$ bounds: $0.456, 0.4632, 0.4656, 0.4668, 0.468$
  4. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0.456, 0.4632[\rightarrow$ bounds: $0.456, 0.46032, 0.46176, 0.46248, 0.4632$
  5. $0.463 \in$ segment 4 ($\rightarrow$ terminate)

source coding

arithmetic coding: example 1/2

sequence $ABCA$, $p_\mathrm{A} = 0.6$, $p_\mathrm{B} = 0.2$, $p_\mathrm{C} = 0.1$, $p_\mathrm{T} = 0.1$,
$A = [0, 0.6[$, $B = [0.6, 0.8[$, $C = [0.8, 0.9[$, $T = [0.9, 1[$

- **decoding** 0.463:
  1. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0, 0.6[$ $\rightarrow$ bounds: $0, 0.36, 0.48, 0.54, 0.6$
  2. $0.463 \in$ segment 2 ($\rightarrow B$),
     - set interval $[0.36, 0.48[$ $\rightarrow$ bounds: $0.36, 0.432, 0.456, 0.468, 0.48$
  3. $0.463 \in$ segment 3 ($\rightarrow C$),
     - set interval $[0.456, 0.468[$ $\rightarrow$ bounds: $0.456, 0.4632, 0.4656, 0.4668, 0.468$
  4. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0.456, 0.4632[$ $\rightarrow$ bounds: $0.456, 0.46032, 0.46176, 0.46248, 0.4632$
  5. $0.463 \in$ segment 4 ($\rightarrow$ terminate)

source coding

arithmetic coding: example 1/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6, p_{\mathrm{B}} = 0.2, p_{\mathrm{C}} = 0.1, p_{\mathrm{T}} = 0.1,$
$A = [0, 0.6[, B = [0.6, 0.8[, C = [0.8, 0.9[, T = [0.9, 1[$

- **decoding** 0.463:
  1. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0, 0.6[\rightarrow$ bounds: $0, 0.36, 0.48, 0.54, 0.6$
  2. $0.463 \in$ segment 2 ($\rightarrow B$),
     - set interval $[0.36, 0.48[\rightarrow$ bounds: $0.36, 0.432, 0.456, 0.468, 0.48$
  3. $0.463 \in$ segment 3 ($\rightarrow C$),
     - set interval $[0.456, 0.468[\rightarrow$ bounds: $0.456, 0.4632, 0.4656, 0.4668, 0.468$
  4. $0.463 \in$ segment 1 ($\rightarrow A$),
     - set interval $[0.456, 0.4632[\rightarrow$ bounds: $0.456, 0.46032, 0.46176, 0.46248, 0.4632$
  5. $0.463 \in$ segment 4 ($\rightarrow$ terminate)

source coding
arithmetic coding: example 1/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6, p_{\mathrm{B}} = 0.2, p_{\mathrm{C}} = 0.1, p_{\mathrm{T}} = 0.1$,
$A = [0, 0.6[, B = [0.6, 0.8[, C = [0.8, 0.9[, T = [0.9, 1[$

- **decoding** 0.463:
  1. $0.463 \in$ segment 1 $(\rightarrow A)$,
     - set interval $[0, 0.6[\rightarrow$ bounds: $0, 0.36, 0.48, 0.54, 0.6$
  2. $0.463 \in$ segment 2 $(\rightarrow B)$,
     - set interval $[0.36, 0.48[\rightarrow$ bounds: $0.36, 0.432, 0.456, 0.468, 0.48$
  3. $0.463 \in$ segment 3 $(\rightarrow C)$,
     - set interval $[0.456, 0.468[\rightarrow$ bounds: $0.456, 0.4632, 0.4656, 0.4668, 0.468$
  4. $0.463 \in$ segment 1 $(\rightarrow A)$,
     - set interval $[0.456, 0.4632[\rightarrow$ bounds: $0.456, 0.46032, 0.46176, 0.46248, 0.4632$
  5. $0.463 \in$ segment 4 $(\rightarrow$ terminate$)$

source coding
arithmetic coding: example 2/2

sequence $ABCA$, $p_\mathrm{A} = 0.6$, $p_\mathrm{B} = 0.2$, $p_\mathrm{C} = 0.1$, $p_\mathrm{T} = 0.1$,
$A = [0, 0.6[$, $B = [0.6, 0.8[$, $C = [0.8, 0.9[$, $T = [0.9, 1[$

- **encoding**
  1. select segment 1, set interval to $[0, 0.6[$
  2. select segment 2, set interval to $[0.36, 0.48[$
  3. select segment 3, set interval to $[0.456, 0.468[$
  4. select segment 1, set interval to $[0.456, 0.4632[$
  5. select segment 4, set interval to $[0.46248, 0.4632[$
  6. choose value from last segment (e.g., 0.463) and transmit

source coding
arithmetic coding: example 2/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6$, $p_{\mathrm{B}} = 0.2$, $p_{\mathrm{C}} = 0.1$, $p_{\mathrm{T}} = 0.1$,
$A = [0, 0.6[$, $B = [0.6, 0.8[$, $C = [0.8, 0.9[$, $T = [0.9, 1[$

- **encoding**

1. select segment 1, set interval to $[0, 0.6[$
2. select segment 2, set interval to $[0.36, 0.48[$
3. select segment 3, set interval to $[0.456, 0.468[$
4. select segment 1, set interval to $[0.456, 0.4632[$
5. select segment 4, set interval to $[0.46248, 0.4632[$
6. choose value from last segment (e.g., 0.463) and transmit

source coding

arithmetic coding: example 2/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6, p_{\mathrm{B}} = 0.2, p_{\mathrm{C}} = 0.1, p_{\mathrm{T}} = 0.1$,
$A = [0, 0.6[, B = [0.6, 0.8[, C = [0.8, 0.9[, T = [0.9, 1[$

- **encoding**
  1. select segment 1, set interval to $[0, 0.6[$
  2. select segment 2, set interval to $[0.36, 0.48[$
  3. select segment 3, set interval to $[0.456, 0.468[$
  4. select segment 1, set interval to $[0.456, 0.4632[$
  5. select segment 4, set interval to $[0.46248, 0.4632[$
  6. choose value from last segment (e.g., 0.463) and transmit

source coding

arithmetic coding: example 2/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6$, $p_{\mathrm{B}} = 0.2$, $p_{\mathrm{C}} = 0.1$, $p_{\mathrm{T}} = 0.1$,
$A = [0, 0.6[$, $B = [0.6, 0.8[$, $C = [0.8, 0.9[$, $T = [0.9, 1[$

- **encoding**
  1. select segment 1, set interval to $[0, 0.6[$
  2. select segment 2, set interval to $[0.36, 0.48[$
  3. select segment 3, set interval to $[0.456, 0.468[$
  4. select segment 1, set interval to $[0.456, 0.4632[$
  5. select segment 4, set interval to $[0.46248, 0.4632[$
  6. choose value from last segment (e.g., 0.463) and transmit

source coding
arithmetic coding: example 2/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6, p_{\mathrm{B}} = 0.2, p_{\mathrm{C}} = 0.1, p_{\mathrm{T}} = 0.1,$
$A = [0, 0.6[, B = [0.6, 0.8[, C = [0.8, 0.9[, T = [0.9, 1[$

- **encoding**
  1. select segment 1, set interval to $[0, 0.6[$
  2. select segment 2, set interval to $[0.36, 0.48[$
  3. select segment 3, set interval to $[0.456, 0.468[$
  4. select segment 1, set interval to $[0.456, 0.4632[$
  5. select segment 4, set interval to $[0.46248, 0.4632[$
  6. choose value from last segment (e.g., 0.463) and transmit

source coding

arithmetic coding: example 2/2

sequence $ABCA$, $p_{\mathrm{A}} = 0.6, p_{\mathrm{B}} = 0.2, p_{\mathrm{C}} = 0.1, p_{\mathrm{T}} = 0.1,$
$A = [0, 0.6[, B = [0.6, 0.8[, C = [0.8, 0.9[, T = [0.9, 1[$

- **encoding**
  1. select segment 1, set interval to $[0, 0.6[$
  2. select segment 2, set interval to $[0.36, 0.48[$
  3. select segment 3, set interval to $[0.456, 0.468[$
  4. select segment 1, set interval to $[0.456, 0.4632[$
  5. select segment 4, set interval to $[0.46248, 0.4632[$
  6. choose value from last segment (e.g., 0.463) and transmit

source coding

arithmetic coding: example 2/2

sequence $ABCA$, $p_\mathrm{A} = 0.6, p_\mathrm{B} = 0.2, p_\mathrm{C} = 0.1, p_\mathrm{T} = 0.1$,
$A = [0, 0.6[, B = [0.6, 0.8[, C = [0.8, 0.9[, T = [0.9, 1[$

- **encoding**
  1. select segment 1, set interval to $[0, 0.6[$
  2. select segment 2, set interval to $[0.36, 0.48[$
  3. select segment 3, set interval to $[0.456, 0.468[$
  4. select segment 1, set interval to $[0.456, 0.4632[$
  5. select segment 4, set interval to $[0.46248, 0.4632[$
  6. choose value from last segment (e.g., 0.463) and transmit

source coding
fundamentals: linear prediction

idea: use preceding samples to estimate/predict future samples.

- **estimate the signal** x

$$\hat{x}(i) = \sum_{j=1}^{\mathcal{O}} b_j \cdot x(i-j)$$

- prediction quality is measured by **power of prediction error**

$$\begin{aligned} e_{\mathrm{P}}(i) &= x(i) - \hat{x}(i) \\ &= x(i) - \sum_{j=1}^{\mathcal{O}} b_j \cdot x(i-j) \end{aligned}$$

source coding

fundamentals: linear prediction

idea: use preceding samples to estimate/predict future samples.

- **estimate the signal** x

$$\hat{x}(i) = \sum_{j=1}^{\mathcal{O}} b_j \cdot x(i-j)$$

- prediction quality is measured by **power of prediction error**

$$\begin{aligned} e_{\mathrm{P}}(i) &= x(i) - \hat{x}(i) \\ &= x(i) - \sum_{j=1}^{\mathcal{O}} b_j \cdot x(i-j) \end{aligned}$$

source coding

fundamentals: linear prediction — first order prediction 1/2

Georgia Tech | Center for Music Technology College of Design

- **prediction** $\hat{x}(i) = b_1 \cdot x(i-1)$
- **prediction error**

$$\begin{aligned}\sigma_e^2 &= \mathcal{E}\left\{(x(i) - b_1 x(i-1))^2\right\} \\ &= \sigma_x^2 + b_1^2\sigma_x^2 - 2b_1 r_{xx}(1) \\ &= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2\end{aligned}$$

- **optimum coefficient**: $\frac{\partial\sigma_e^2}{\partial b_1} = 0$

$$\begin{aligned}2b_1\sigma_x^2 - 2\rho_{xx}(1)\sigma_x^2 &= 0 \\ b_1 &= \rho_{xx}(1)\end{aligned}$$

- **minimum prediction error power**

$$\begin{aligned}\sigma_e^2 &= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2 \\ &= \left(1 + \rho_{xx}(1)^2 - 2\rho_{xx}(1)\rho_{xx}(1)\right)\sigma_x^2 \\ &= (1 - \rho_{xx}(1))\sigma_x^2\end{aligned}$$

source coding

fundamentals: linear prediction — first order prediction 1/2

- **prediction** $\hat{x}(i) = b_1 \cdot x(i-1)$
- **prediction error**

$$\begin{aligned}\sigma_e^2 &= \mathcal{E}\left\{(x(i) - b_1 x(i-1))^2\right\} \\ &= \sigma_x^2 + b_1^2\sigma_x^2 - 2b_1 r_{xx}(1) \\ &= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2\end{aligned}$$

- **optimum coefficient**: $\frac{\partial \sigma_e^2}{\partial b_1} = 0$

$$\begin{aligned}2b_1\sigma_x^2 - 2\rho_{xx}(1)\sigma_x^2 &= 0 \\ b_1 &= \rho_{xx}(1)\end{aligned}$$

- **minimum prediction error power**

$$\begin{aligned}\sigma_e^2 &= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2 \\ &= \left(1 + \rho_{xx}(1)^2 - 2\rho_{xx}(1)\rho_{xx}(1)\right)\sigma_x^2 \\ &= (1 - \rho_{xx}(1))\sigma_x^2\end{aligned}$$

source coding

fundamentals: linear prediction — first order prediction 1/2

- **prediction** $\hat{x}(i) = b_1 \cdot x(i-1)$
- **prediction error**

$$\begin{aligned}
\sigma_e^2 &= \mathcal{E}\left\{(x(i) - b_1 x(i-1))^2\right\} \\
&= \sigma_x^2 + b_1^2\sigma_x^2 - 2b_1 r_{xx}(1) \\
&= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2
\end{aligned}$$

- **optimum coefficient**: $\frac{\partial\sigma_e^2}{\partial b_1} = 0$

$$\begin{aligned}
2b_1\sigma_x^2 - 2\rho_{xx}(1)\sigma_x^2 &= 0 \\
b_1 &= \rho_{xx}(1)
\end{aligned}$$

- **minimum prediction error power**

$$\begin{aligned}
\sigma_e^2 &= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2 \\
&= \left(1 + \rho_{xx}(1)^2 - 2\rho_{xx}(1)\rho_{xx}(1)\right)\sigma_x^2 \\
&= (1 - \rho_{xx}(1))\sigma_x^2
\end{aligned}$$

source coding

fundamentals: linear prediction — first order prediction 1/2

- **prediction** $\hat{x}(i) = b_1 \cdot x(i-1)$
- **prediction error**

$$\begin{aligned}
\sigma_e^2 &= \mathcal{E}\left\{(x(i) - b_1 x(i-1))^2\right\} \\
&= \sigma_x^2 + b_1^2\sigma_x^2 - 2b_1 r_{xx}(1) \\
&= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2
\end{aligned}$$

- **optimum coefficient**: $\frac{\partial\sigma_e^2}{\partial b_1} = 0$

$$\begin{aligned}
2b_1\sigma_x^2 - 2\rho_{xx}(1)\sigma_x^2 &= 0 \\
b_1 &= \rho_{xx}(1)
\end{aligned}$$

- **minimum prediction error power**

$$\begin{aligned}
\sigma_e^2 &= \left(1 + b_1^2 - 2b_1\rho_{xx}(1)\right)\sigma_x^2 \\
&= \left(1 + \rho_{xx}(1)^2 - 2\rho_{xx}(1)\rho_{xx}(1)\right)\sigma_x^2 \\
&= (1 - \rho_{xx}(1))\sigma_x^2
\end{aligned}$$

source coding

fundamentals: linear prediction — first order prediction 2/2

$$\sigma_e^2 = (1 - \rho_{xx}(1))\sigma_x^2$$

- **observations**:
  - power of prediction error always smaller or equal the power of the signal
  - question: when is it equal to the signal?

- **special case**: $b_1 = 1$

$$\begin{aligned}\hat{x}(i) &= x(i-1)\\ e_{\mathrm{P}} &= x(i) - x(i-1)\\ \sigma_e^2 &= (1 + b_1^2 - 2b_1\rho_{xx}(1))\sigma_x^2\\ &= 2(1 - \rho_{xx}(1))\sigma_x^2\end{aligned}$$

source coding

fundamentals: linear prediction — first order prediction 2/2

$$\sigma_e^2 = (1 - \rho_{xx}(1))\sigma_x^2$$

- **observations**:
  - power of prediction error always smaller or equal the power of the signal
  - question: when is it equal to the signal?
- **special case**: $b_1 = 1$

$$\begin{aligned} \hat{x}(i) &= x(i-1) \\ e_\mathrm{P} &= x(i) - x(i-1) \\ \sigma_e^2 &= (1 + b_1^2 - 2b_1\rho_{xx}(1))\sigma_x^2 \\ &= 2(1 - \rho_{xx}(1))\sigma_x^2 \end{aligned}$$

source coding

fundamentals: linear prediction — prediction coefficients

- prediction gain depends on
  - predictor coefficients b_j
  - signal

- optimal coefficients can be derived by finding minimum of prediction error

$$\frac{\partial \sigma_e^2}{\partial b_j} = 0$$

$\Rightarrow$ (without derivation)

$$r_{xx}(\eta) = \sum_{j=1}^{\mathcal{O}} b_{j,\mathrm{opt}} \cdot r_{xx}(\eta - j), \qquad 1 \leq \eta \leq \mathcal{O}$$

$$\begin{aligned} \boldsymbol{r}_{xx} &= \boldsymbol{R}_{xx} \cdot \boldsymbol{b}_{\mathrm{opt}} \\ \boldsymbol{b}_{\mathrm{opt}} &= \boldsymbol{R}_{xx}^{-1} \cdot \boldsymbol{r}_{xx} \end{aligned}$$

source coding

fundamentals: linear prediction — prediction coefficients

- prediction gain depends on
  - predictor coefficients b_j
  - signal

- optimal coefficients can be derived by finding minimum of prediction error

$$\frac{\partial \sigma_e^2}{\partial b_j} = 0$$

$\Rightarrow$ (without derivation)

$$r_{xx}(\eta) = \sum_{j=1}^{\mathcal{O}} b_{j,\mathrm{opt}} \cdot r_{xx}(\eta - j), \qquad 1 \leq \eta \leq \mathcal{O}$$

$$\begin{aligned} \boldsymbol{r}_{xx} &= \boldsymbol{R}_{xx} \cdot \boldsymbol{b}_{\mathrm{opt}} \\ \boldsymbol{b}_{\mathrm{opt}} &= \boldsymbol{R}_{xx}^{-1} \cdot \boldsymbol{r}_{xx} \end{aligned}$$

source coding

fundamentals: linear prediction — summary

- **predictor length**
  - rule of thumb: the longer the predictor, the better the prediction
  - can range from 10 coefficients to hundreds
- **predictor coefficient updates**
  - better signal adaptation if coefficients are updated block-by-block
- **input signals**
  - white noise/random processes cannot be predicted
  - periodic signals may theoretically be perfectly predicted

source coding

fundamentals: linear prediction — summary

- **predictor length**
  - rule of thumb: the longer the predictor, the better the prediction
  - can range from 10 coefficients to hundreds
- **predictor coefficient updates**
  - better signal adaptation if coefficients are updated block-by-block
- **input signals**
  - white noise/random processes cannot be predicted
  - periodic signals may theoretically be perfectly predicted

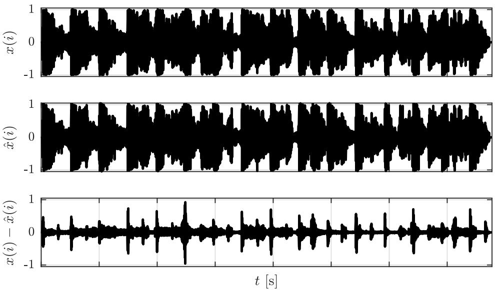
source coding
fundamentals: linear prediction — summary

- **predictor length**
  - rule of thumb: the longer the predictor, the better the prediction
  - can range from 10 coefficients to hundreds
- **predictor coefficient updates**
  - better signal adaptation if coefficients are updated block-by-block
- **input signals**
  - white noise/random processes cannot be predicted
  - periodic signals may theoretically be perfectly predicted

source coding

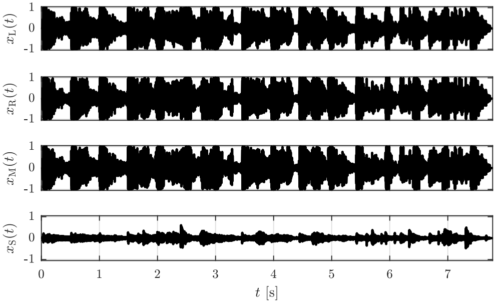
fundamentals: linear prediction — audio example

order: 20



matlab source: plotLinPred.m

source coding
fundamentals: joint channels



$$M = \frac{L+R}{2}$$

$$S = \frac{R-L}{2}$$

$$\begin{aligned} L &= M + S \\ R &= M - S \end{aligned}$$

matlab source: plotMS.m

source coding
summary

- bitrate can be reduced by removing removing redundancy and/or irrelevance
- removing redundancy:
  - entropy coding: transmit frequent symbols with shorter codes
  - linear prediction: transmit diff signal plus predictor coefficients
- removing irrelevance:
  - reduce quantization wordlength
  - see slides below