Digital Signal Processing for Music

Part 24: Source Coding

Andrew Beck

Intro

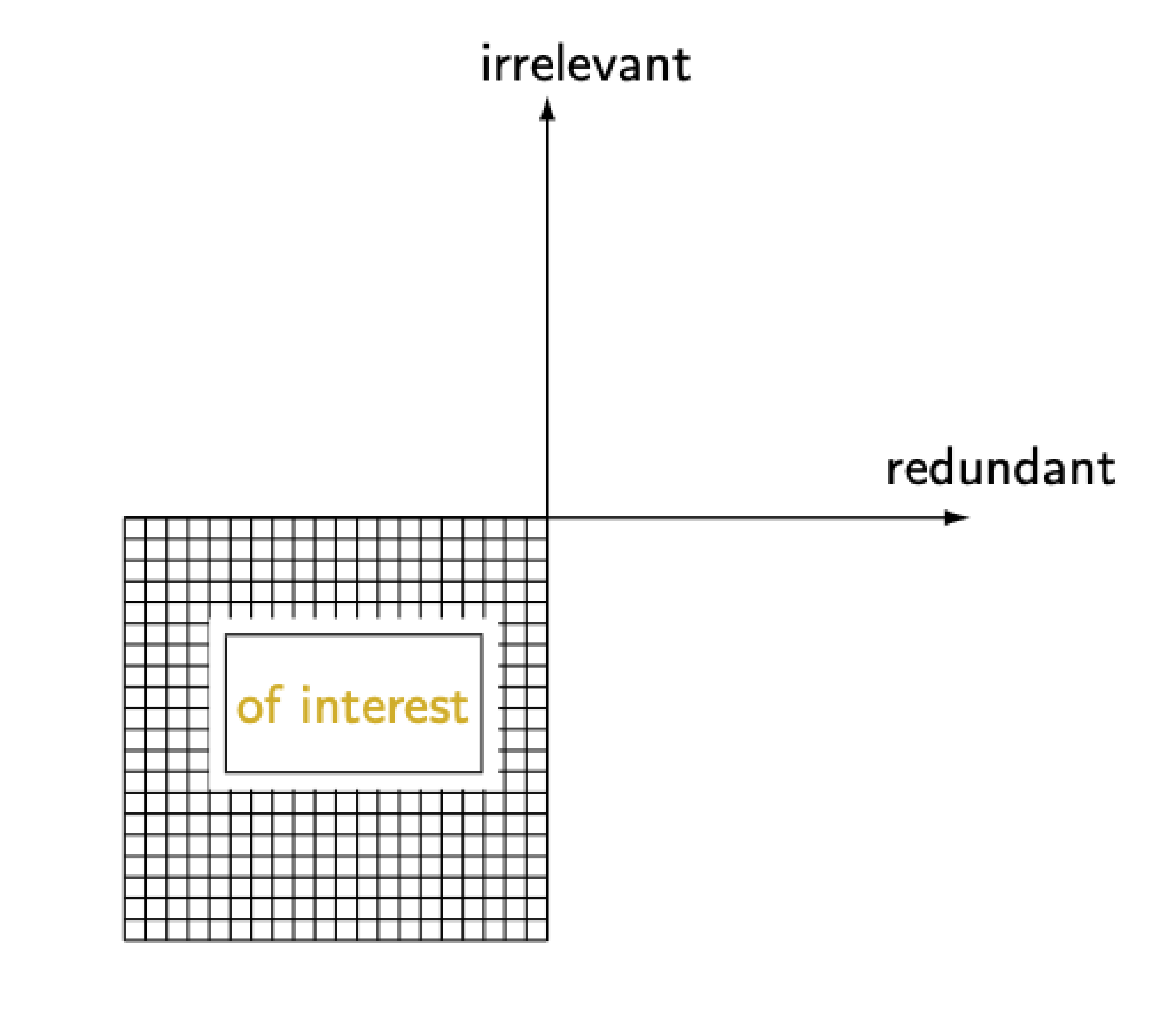
- Typical audio **bit rates**

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

- **Reasons** for bit rate reduction
  - Economical reasons: Cheaper transmission/storage
  - Technical reasons: Restricted storage / transmission bandwidth
- **Applications** for source coding
  - Internet: streaming, distribution, p2p, VoIP, ...
  - Media: DVD-V/A, ...

Reducing Bitrate

- **Lossless**:
  Remove *redundant* information (unnecessary to reconstruct the signal)
  - Entropy coding
  - (Linear predictive coding)
- **Lossy**:
  Remove *irrelevant* information (not "missed" by the recipient)
  - Waveform coding
  - Perceptual coding


irrelevant
redundant
of interest

Information Theory: Definitions

Note: Words to be transmitted referred to as *symbols*

Information Content

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

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

Information Theory: Definitions

Entropy

The *Expected Value* of the information content; the *theoretic minimum of bits* required for transmission

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

Information Content & Entropy Examples

Dice: $p_n = \frac{1}{6}$

$$I_n = \log_2\left(\frac{1}{p_n}\right) = 2.58\text{bit}$$

$$H = 2.58\text{bit}$$

Imperfect dice: $p_0 = \frac{1}{2},\ p_{1...5} = \frac{1}{10}$

$$I_1 = \log_2(2) = 1\text{bit}$$

$$I_{2...6} = \log_2(10) = 3.32\text{bit}$$

$$H = \frac{1}{2} \cdot 1 + \frac{5}{10} \cdot 3.32 = 2.16\text{bit}$$

Entropy Coding

Idea: Use shorter words for frequent symbols

- 3 possible signals

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 → 010110
- Required bits:

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

≡

Entropy Coding

- 3 possible signals

Symbol	Probability	Word
A	$p = 0.7$	0
B	$p = 0.2$	10
C	$p = 0.1$	11

- Entropy

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

- Transmit the following group of symbols: ABCA → 010110
- Required bits:

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

Huffman Coding

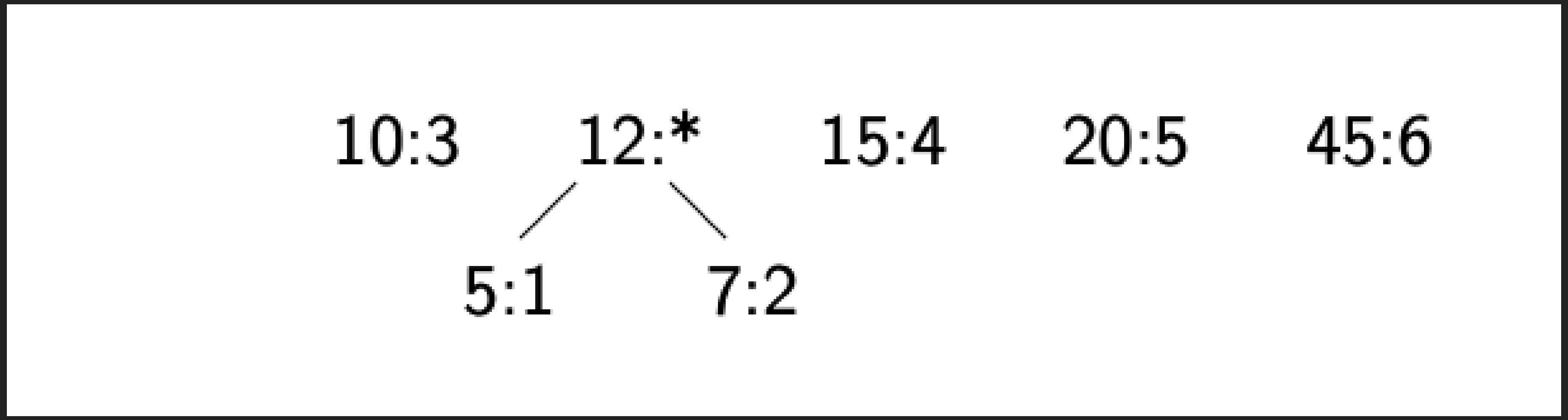
Steps

- Sort symbols according to frequency
- Combine two lowest symbols into new entry (sum)
- Add new entry to list
- Repeat until only one element left

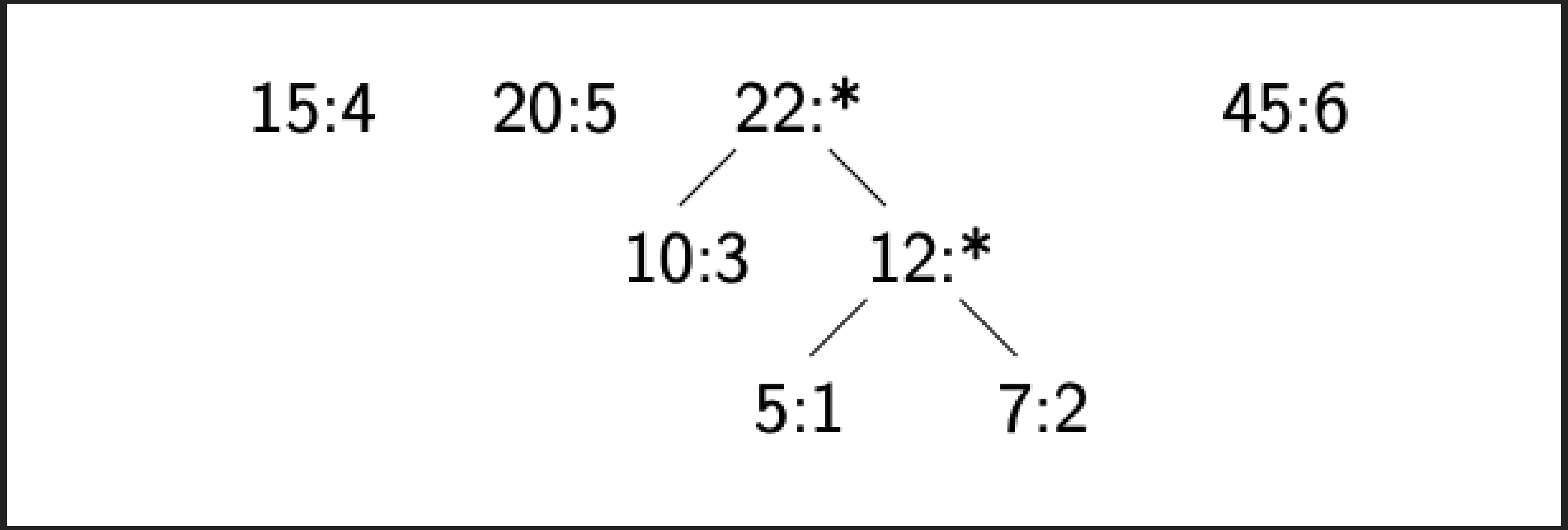
Huffman Coding: Example

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

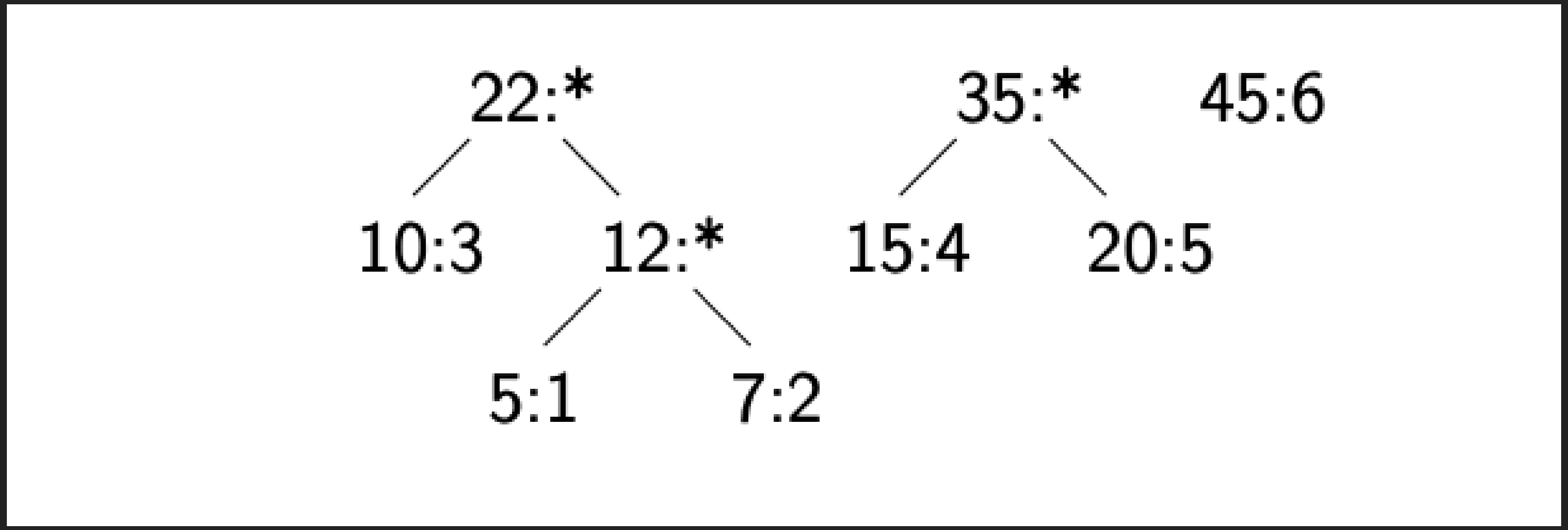
Huffman Coding: Example



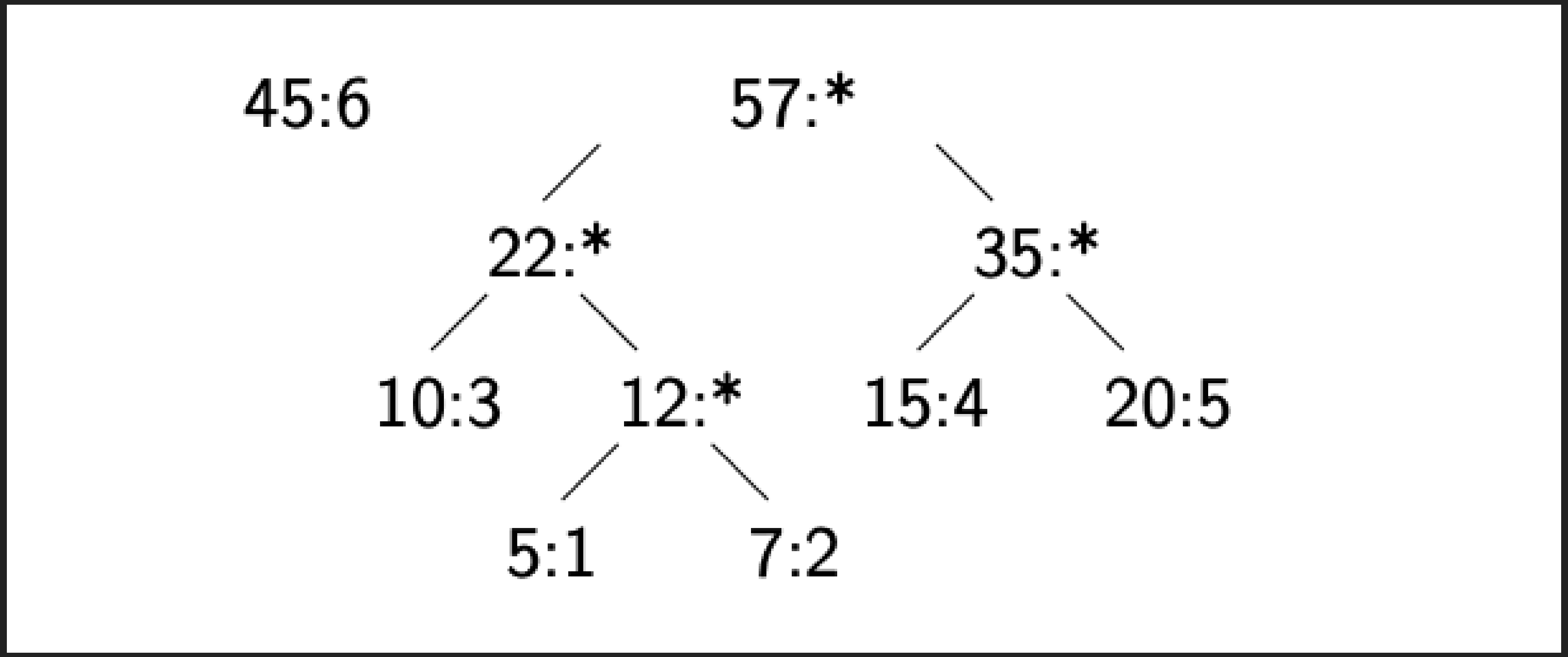
Huffman Coding: Example



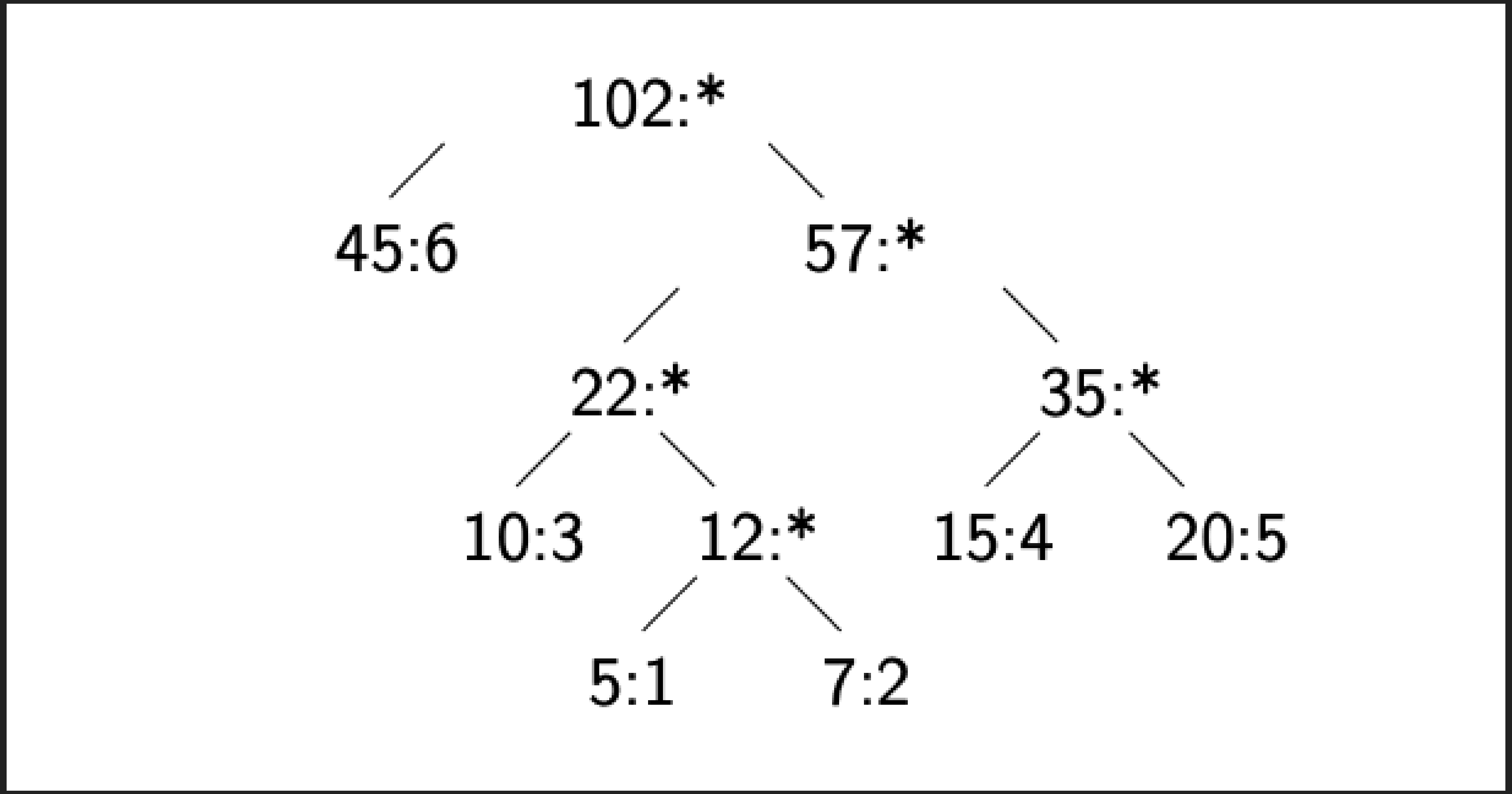
Huffman Coding: Example



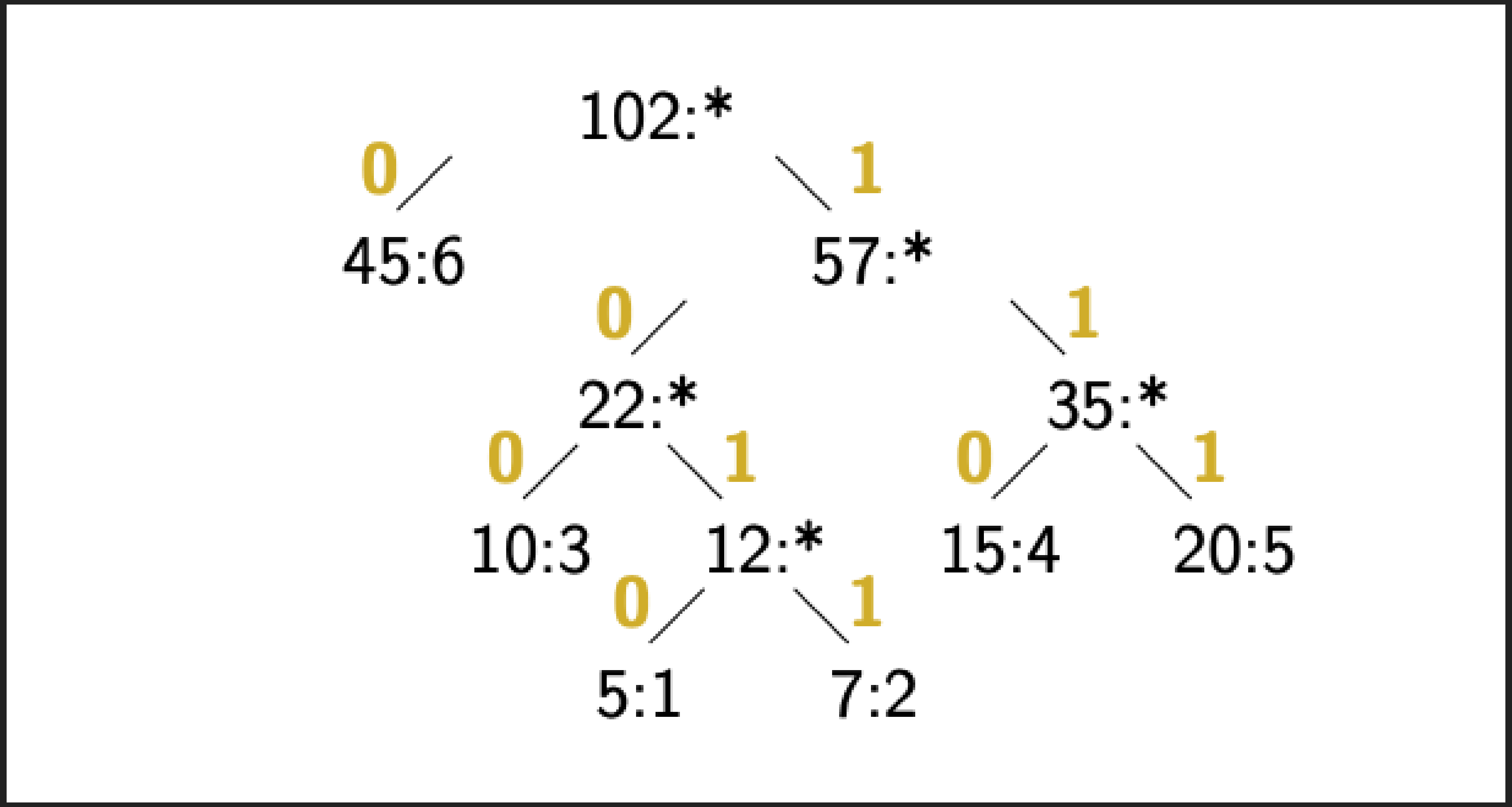
Huffman Coding: Example



Huffman Coding: Example

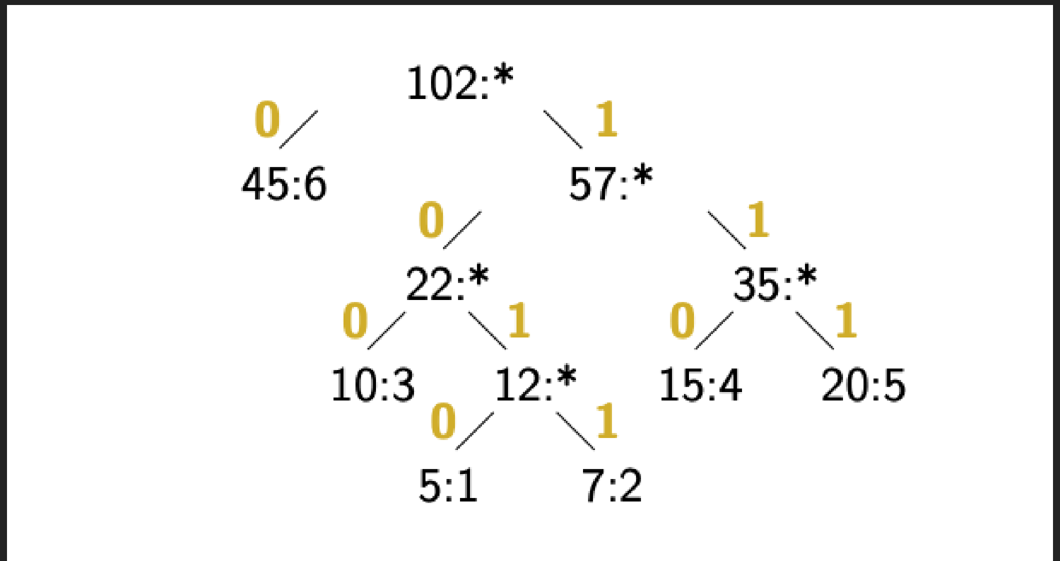


Huffman Coding: Example



Huffman Coding: Example

Frequency	Symbol	Code
5	1	1010
7	2	1011
10	3	100
15	4	110
20	5	1111
45	6	0



Huffman Coding for audio signals

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

Arithmetic Coding

- Huffman coding is only optimal if $p_n = 1/2^k$

Alternative: **Arithmetic Coding**

- Allows other probability distributions
- Encodes the whole sequence in one fractional number $0.0 \leq f < 1.0$

Steps

Arithmetic Coding: Decoding Example

Sequence ABCA, $p_A = \mathbf{0.6}, p_B = \mathbf{0.2}, p_C = \mathbf{0.1}, p_T = \mathbf{0.1}$

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

Decoding 0.463

Arithmetic Coding: Encoding Example

Sequence ABCA, $p_A = 0.6, p_B = 0.2, p_C = 0.1, p_T = 0.1$

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

Encoding 0.463

- Select segment 1, set interval to [0, 0.6)
- Select segment 2, set interval to [0.36, 0.48)
- Select segment 3, set interval to [0.456, 0.468)
- Select segment 1, set interval to [0.456, 0.4632)
- Select segment 4, set interval to [0.46248, 0.4632)
- Choose value from last segment (e.g., 0.463) and transmit

Fundamentals: Linear Prediction

Idea: Use preceding samples to estimate/predict future samples

Fundamentals: Linear Prediction - First Order Prediction

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

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

Fundamentals: Linear Prediction - First Order Prediction

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

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

- **Minimum prediction error power:**

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

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

Linear Prediction - Coefficients

- Prediction gain depends on
  - Predictor coefficients b_j
  - Signal
- Optimal coefficients can be derived by finding minimum of prediction error

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

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

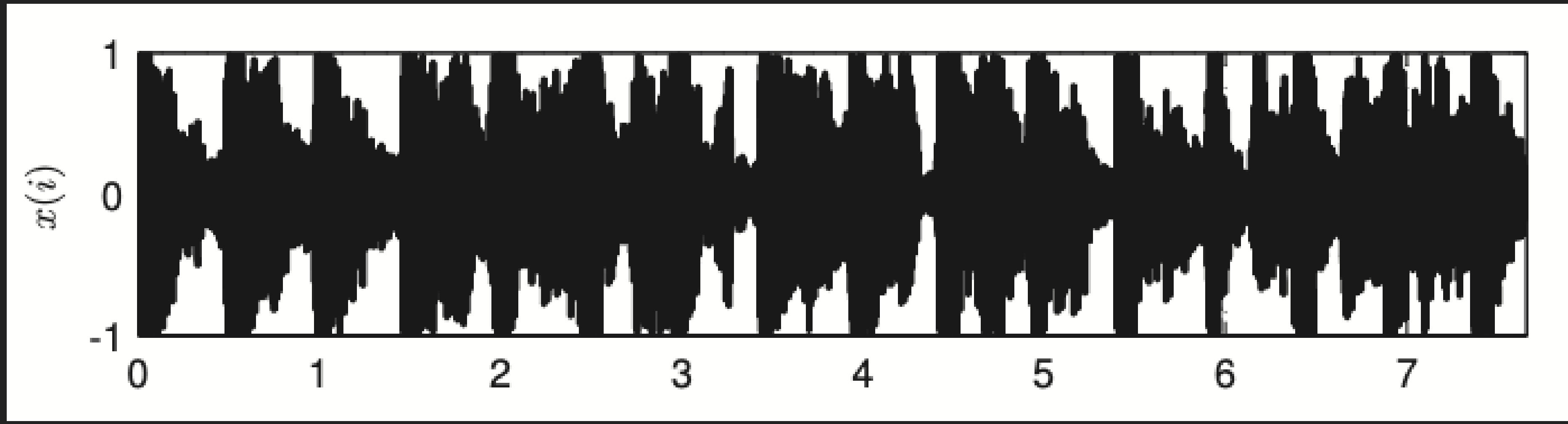
$$\vec{r}_{xx} = R_{xx} \cdot \vec{b}_{\mathrm{opt}}$$

$$\vec{b}_{\mathrm{opt}} = R_{xx}^{-1} \cdot \vec{r}_{xx}$$

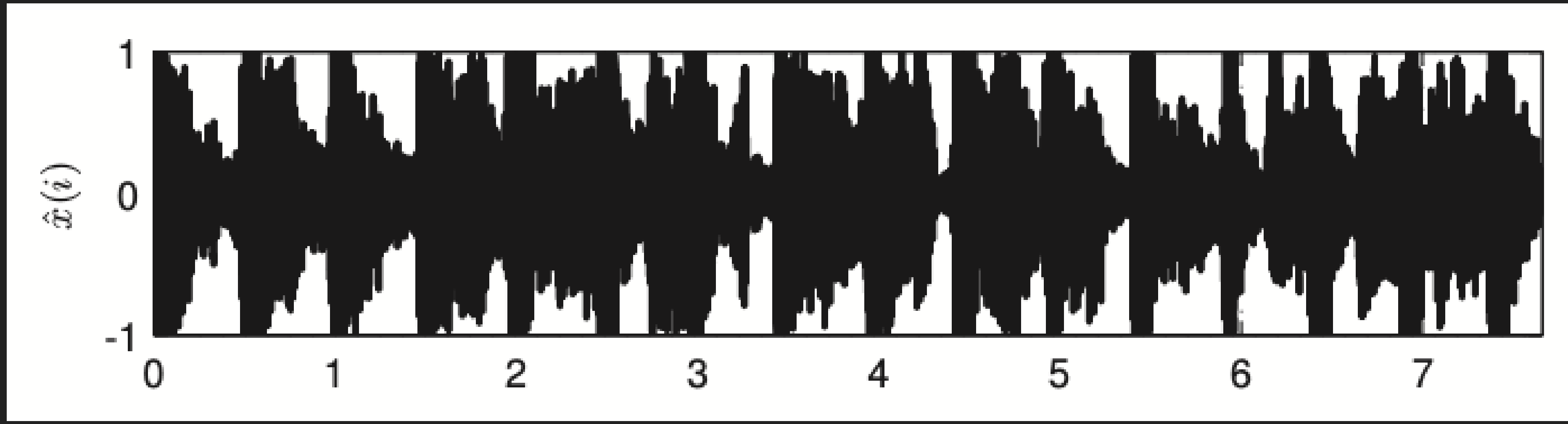
Linear Prediction - Summary

- **Predictor Length**
  - Rule of thumb: the longer the predictor, the better the prediction
  - Can range from 10 coefficients to hunders
- **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

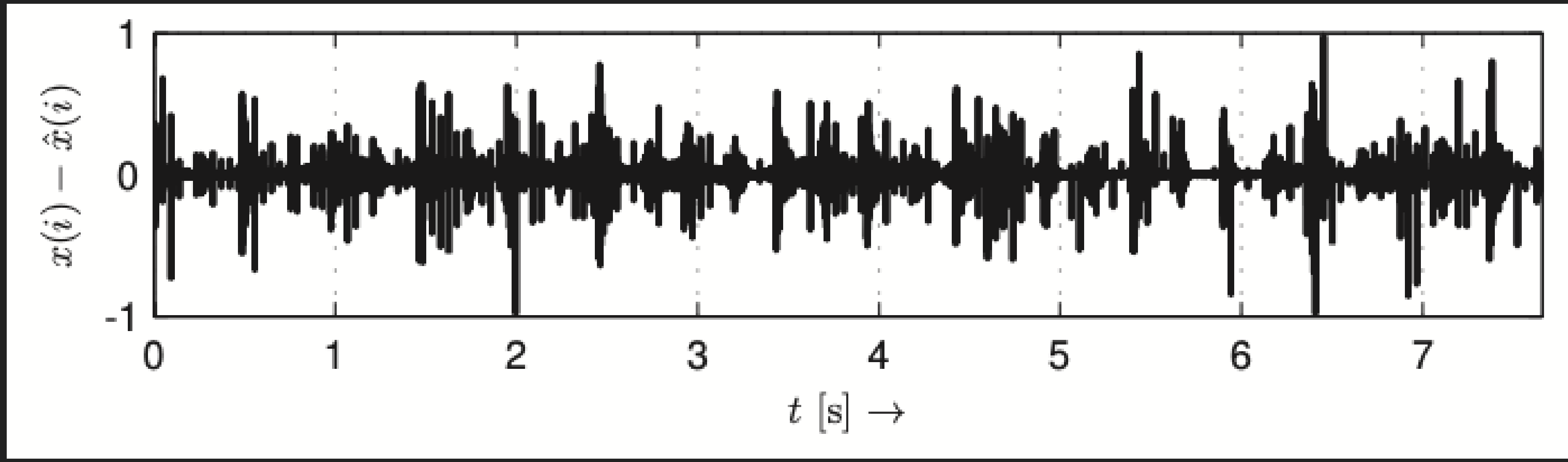
Linear Prediction Examples



0:00 / 0:07

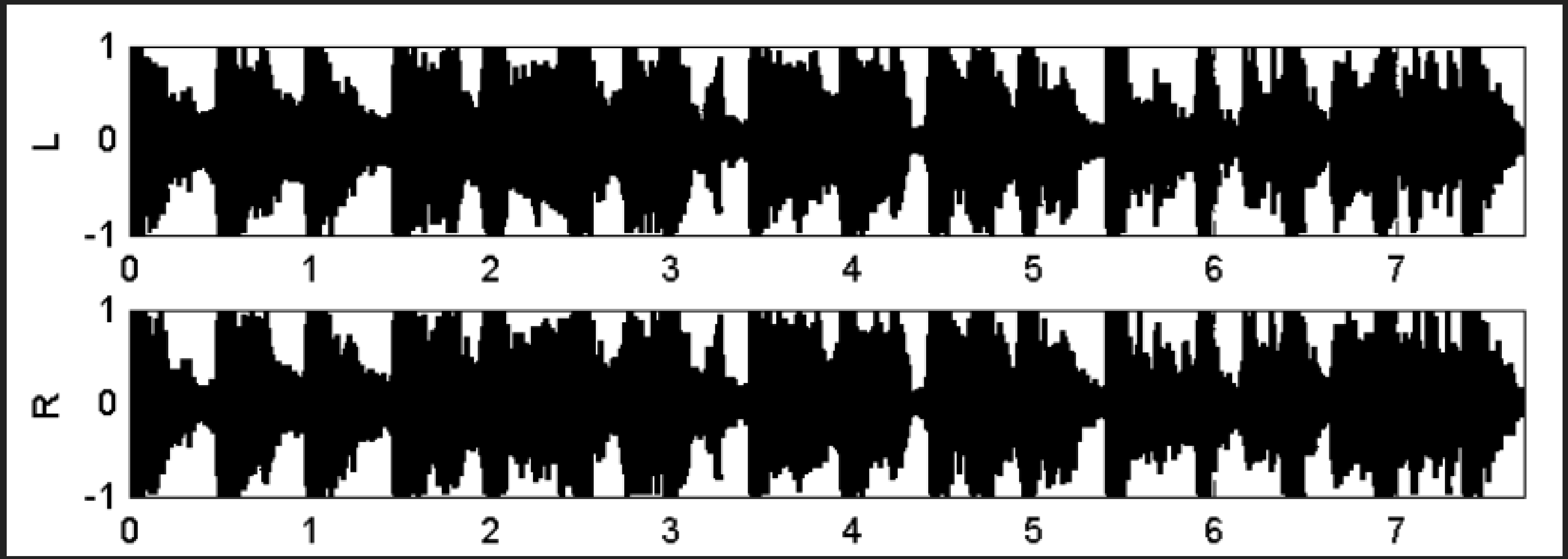


0:00 / 0:07

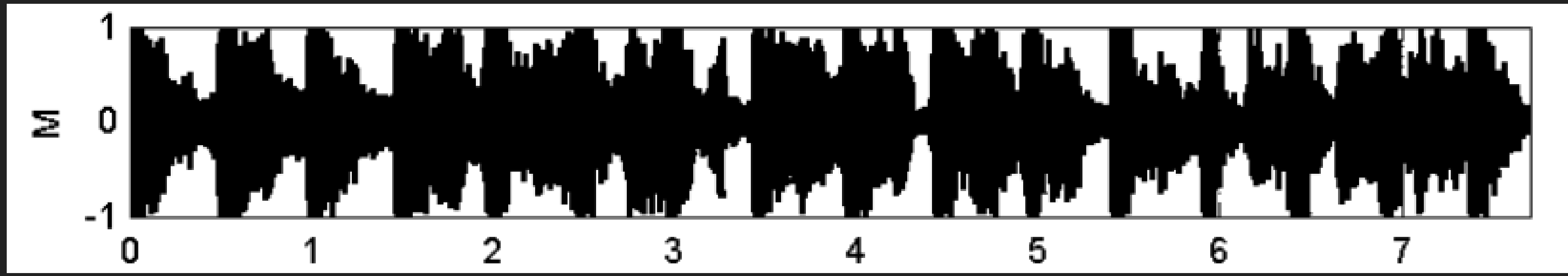


0:00 / 0:07

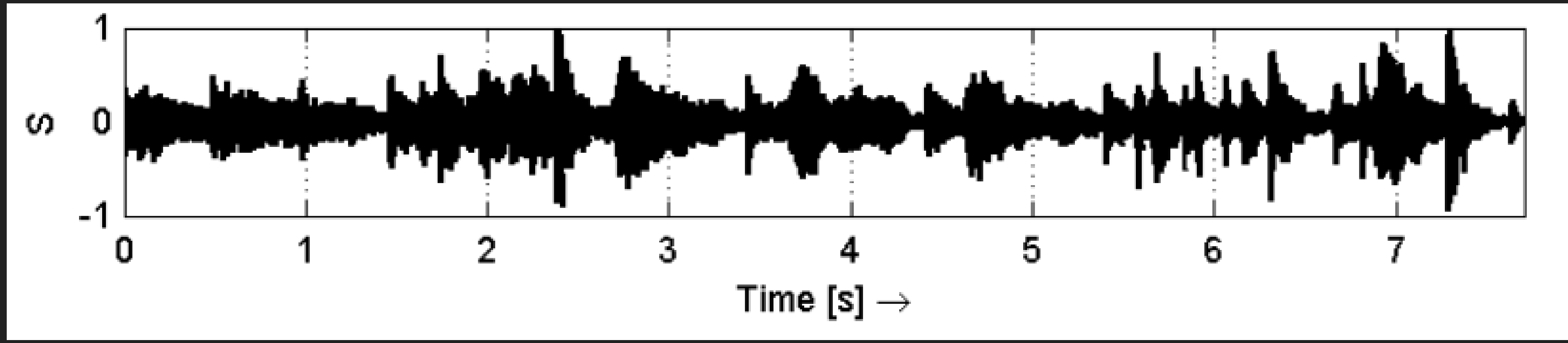
Linear Prediction Examples



0:00 / 0:07



0:00 / 0:07

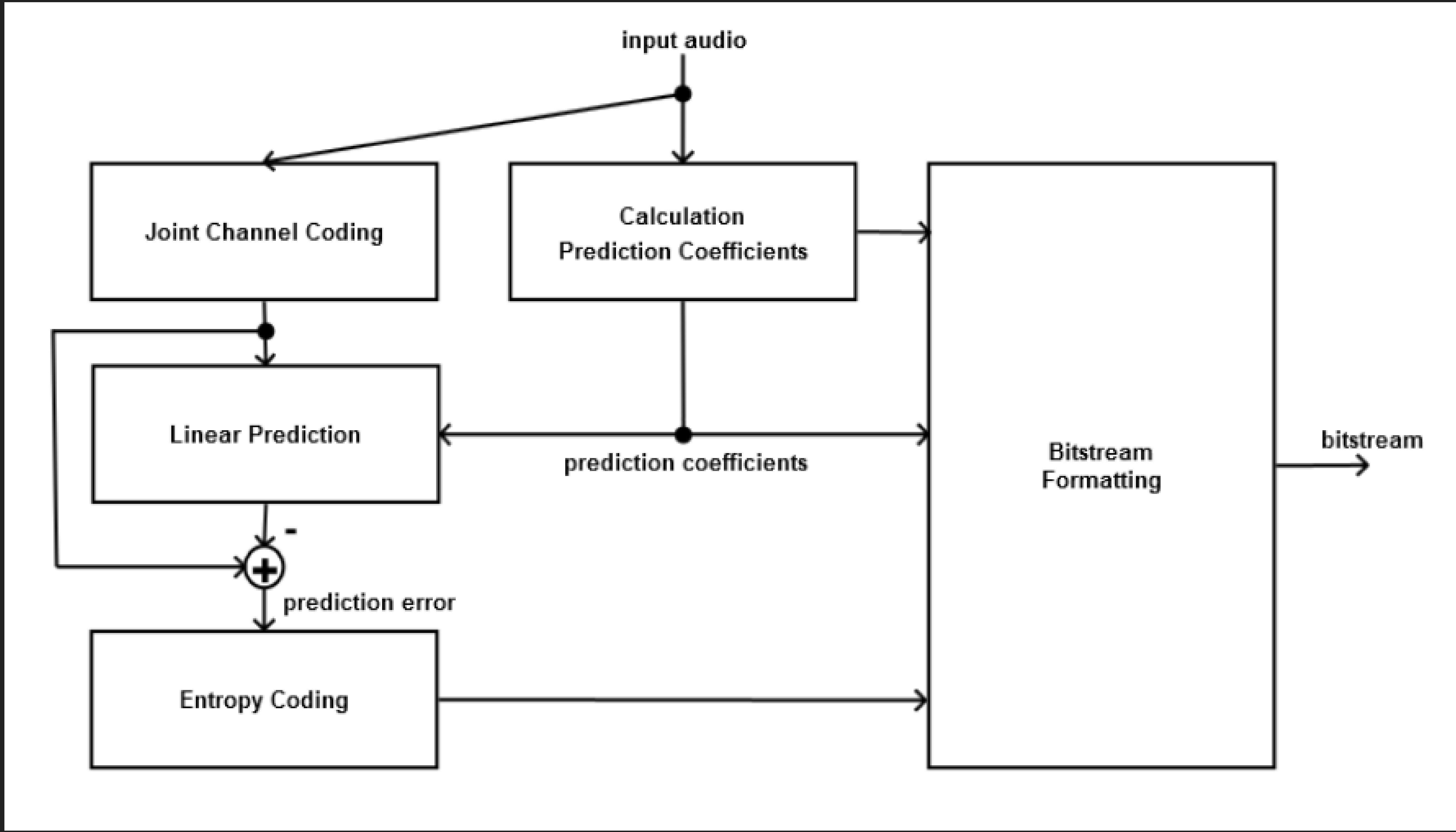


0:00 / 0:07

Summary

- Bitrate can be reduced by 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 / lower sample rate
  - More techniques discussed in future classes

Redundancy Coding



- **Properties**
  - Perfect signal reconstruction
  - Bitrate reduction depends on input signal
    Typical gain (stereo, 48k): Factor 2
  - No constant bitrate → Streaming only with large buffers

Name	Sampling Rates	Channels	Word Length
Shorten	All	2	8 / 16
FLAC	1-1048k	8	4-32
MLP	44.1k - 192k	63	1 - 24
ALS	All	65536	1 - 32 (int), 32 (float)