Chapter 3

Lossless Data Compression

Po-Ning Chen, Professor
Department of Communications Engineering
National Chiao Tung University
Hsin Chu, Taiwan 300, R.O.C.
Principle of Data Compression

• Average codeword length

E.g.

\[
\begin{align*}
P_X(x = \text{outcome}_A) &= 0.5; \\
P_X(x = \text{outcome}_B) &= 0.25; \quad \text{and} \quad \text{code(outcome}_A) = 0; \\
P_X(x = \text{outcome}_C) &= 0.25. \quad \text{code(outcome}_B) = 10; \\
\end{align*}
\]

Then the average codeword length is

\[
\begin{align*}
\text{len}(0) \cdot P_X(A) + \text{len}(10) \cdot P_X(B) + \text{len}(11) \cdot P_X(C) \\
&= 1 \cdot 0.5 + 2 \cdot 0.25 + 2 \cdot 0.25 \\
&= 1.5 \text{ bits.}
\end{align*}
\]

• Categories of codes

– Variable-length codes
– Fixed-length codes (often treated as a subclass of variable-length codes)
  * Segmentation is normally considered an implicit part of the codewords.
Example of segmentation of fixed-length codes.

E.g. To encode the final grades of a class with 100 students. There are three grade levels: $A$, $B$ and $C$.

- Without segmentation

$$\lceil \log_2 3^{100} \rceil = 159 \text{ bits.}$$

- With segmentation length of 10 students

$$10 \times \lceil \log_2 3^{10} \rceil = 160 \text{ bits.}$$
Principle of Data Compression

• Fixed-length codes
  – Block codes
    * Encoding of the next segment is independent of the previous segments
  – Fixed-length tree codes
    * Encoding of the next segment, somehow, retains and uses some knowledge of earlier segments

• Block diagram of a data compression system

Block diagram of a data compression system.
Key Difference in Data Compress Schemes

- Block codes for *asymptotic* lossless data compression
  - Asymptotic in blocklength $n$
- Variable-length codes for *completely* lossless data compression

**Block Codes for DMS**

**Definition 3.1 (discrete memoryless source)** A discrete memoryless source (DMS) consists of a sequence of independent and identically distributed (i.i.d.) random variables, $X_1, X_2, X_3, \ldots$, etc. In particular, if $P_X(\cdot)$ is the common distribution of $X_i$'s, then

$$P_{X^n}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P_X(x_i).$$
Block Codes for DMS

**Definition 3.2** An \((n, M)\) block code for lossless data compression of blocklength \(n\) and size \(M\) is a set \(\{c_1, c_2, \ldots, c_M\}\) consisting of \(M\) codewords, each codeword represents a group of source symbols of length \(n\).

- One can binary-index the codewords in \(\{c_1, c_2, \ldots, c_M\}\) by \(r^\triangleq \lceil \log_2 M \rceil\) bits.
- Since the behavior of block codes is investigated as \(n\) and \(M\) large (or more precisely, tend to infinity), it is legitimate to replace \(\lceil \log_2 M \rceil\) by \(\log_2 M\).
- With this convention, the *data compression rate* or *code rate for data compression* is
  \[
  \text{bits required per source symbol} = \frac{r}{n} \approx \frac{1}{n} \log_2 M.
  \]
- For analytical convenience, *nats* (natural logarithm) is often used instead of *bits*; and hence, the code rate becomes:
  \[
  \text{nats required per source symbol} = \frac{1}{n} \log M.
  \]
Block Codes for DMS

• Encoding of a block code

\[ \cdots (x_{3n}, \ldots, x_{31})(x_{2n}, \ldots, x_{21})(x_{1n}, \ldots, x_{11}) \rightarrow \cdots |c_{m_3}|c_{m_2}|c_{m_1} \]

• AEP or entropy stability property

**Theorem 3.3 (asymptotic equipartition property or AEP)** If \( X_1, X_2, \ldots, X_n, \ldots \) are i.i.d., then

\[
\frac{-1}{n} \log P_{X^n}(X_1, \ldots, X_n) \rightarrow H(X) \quad \text{in probability.}
\]

**Proof:** This theorem follows by the observation that for i.i.d. sequence,

\[
\frac{-1}{n} \log P_{X^n}(X_1, \ldots, X_n) = -\frac{1}{n} \sum_{i=1}^{n} \log P_X(X_i),
\]

and the weak law of large numbers. \( \square \)
Block Codes for DMS

- (Weakly) $\delta$-typical set

$$\mathcal{F}_n(\delta) \triangleq \left\{ x^n \in \mathcal{X}^n : \left| -\frac{1}{n} \sum_{i=1}^{n} \log P_X(x_i) - H(X) \right| < \delta \right\}.$$ 

**E.g.** $n = 2$ and $\delta = 0.3$ and $\mathcal{X} = \{A, B, C, D\}$.

The source distribution is

$$\begin{align*}
P_X(A) &= 0.4 \\
P_X(B) &= 0.3 \\
P_X(C) &= 0.2 \\
P_X(D) &= 0.1
\end{align*}$$

The entropy equals:

$$0.4 \log \frac{1}{0.4} + 0.3 \log \frac{1}{0.3} + 0.2 \log \frac{1}{0.2} + 0.1 \log \frac{1}{0.1} = 1.27985 \text{ nats}$$

Then for $x_1^n = (A, A)$,

$$\left| -\frac{1}{2} \sum_{i=1}^{2} \log P_X(x_i) - H(X) \right| = \left| -\frac{1}{2} (\log P_X(A) + \log P_X(A)) - 1.27985 \right|$$

$$= \left| -\frac{1}{2} (\log 0.4 + \log 0.4) - 1.27985 \right| = 0.364$$
## Block Codes for DMS

\[ I: 3-8 \]

### Table

<table>
<thead>
<tr>
<th>Source</th>
<th>[-\frac{1}{2} \sum_{i=1}^{2} \log P_X(x_i) - H(X)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AA)</td>
<td>0.364 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(AB)</td>
<td>0.220 nats (\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(AC)</td>
<td>0.017 nats (\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(AD)</td>
<td>0.330 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(BA)</td>
<td>0.220 nats (\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(BB)</td>
<td>0.076 nats (\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(BC)</td>
<td>0.127 nats (\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(BD)</td>
<td>0.473 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(CA)</td>
<td>0.017 nats (\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(CB)</td>
<td>0.127 nats (\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(CC)</td>
<td>0.330 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(CD)</td>
<td>0.676 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(DA)</td>
<td>0.330 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(DB)</td>
<td>0.473 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(DC)</td>
<td>0.676 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
<tr>
<td>(DD)</td>
<td>1.023 nats (\not\in \mathcal{F}_2(0.3))</td>
</tr>
</tbody>
</table>

\[ \Rightarrow \mathcal{F}_2(0.3) = \{ AB, AC, BA, BB, BC, CA, CB \} \]

## Block Codes for DMS

\[
-\frac{1}{2} \sum_{i=1}^{2} \log P_X(x_i) - H(X) \]

<table>
<thead>
<tr>
<th>Source</th>
<th>$-\frac{1}{2} \sum_{i=1}^{2} \log P_X(x_i) - H(X)$</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>0.364 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>AB</td>
<td>0.220 nats $\in \mathcal{F}_2(0.3)$</td>
<td>001</td>
</tr>
<tr>
<td>AC</td>
<td>0.017 nats $\in \mathcal{F}_2(0.3)$</td>
<td>010</td>
</tr>
<tr>
<td>AD</td>
<td>0.330 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>BA</td>
<td>0.220 nats $\in \mathcal{F}_2(0.3)$</td>
<td>011</td>
</tr>
<tr>
<td>BB</td>
<td>0.076 nats $\in \mathcal{F}_2(0.3)$</td>
<td>100</td>
</tr>
<tr>
<td>BC</td>
<td>0.127 nats $\in \mathcal{F}_2(0.3)$</td>
<td>101</td>
</tr>
<tr>
<td>BD</td>
<td>0.473 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>CA</td>
<td>0.017 nats $\in \mathcal{F}_2(0.3)$</td>
<td>110</td>
</tr>
<tr>
<td>CB</td>
<td>0.127 nats $\in \mathcal{F}_2(0.3)$</td>
<td>111</td>
</tr>
<tr>
<td>CC</td>
<td>0.330 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>CD</td>
<td>0.676 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>DA</td>
<td>0.330 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>DB</td>
<td>0.473 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>DC</td>
<td>0.676 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
<tr>
<td>DD</td>
<td>1.023 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
</tr>
</tbody>
</table>

We can therefore encode the **seven** outcomes in $\mathcal{F}_2(0.3)$ by **seven** distinct codewords, and encode all the remaining **nine** outcomes outside $\mathcal{F}_2(0.3)$ by a **single** codeword.
Almost all the source sequences in $\mathcal{F}_n(\delta)$ are nearly *equiprobable* or *equally surprising* (cf. Property 3 of Theorem 3.4); Hence, Theorem 3.3 is named AEP.

**E.g.** The probabilities of the elements in

$$\mathcal{F}_2(0.3) = \{AB, AC, BA, BB, BC, CA, CB\}$$

are respectively 0.12, 0.08, 0.12, 0.09, 0.06, 0.08 and 0.06.

The sum of these seven probability masses are 0.61.
## Block Codes for DMS

I: 3-11

<table>
<thead>
<tr>
<th>Source</th>
<th>$-\frac{1}{2} \sum_{i=1}^{2} \log P_{X}(x_i) - H(X)$</th>
<th>codeword</th>
<th>reconstructed sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$AA$</td>
<td>0.364 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$AB$</td>
<td>0.220 nats $\in \mathcal{F}_2(0.3)$</td>
<td>001</td>
<td>AB</td>
</tr>
<tr>
<td>$AC$</td>
<td>0.017 nats $\in \mathcal{F}_2(0.3)$</td>
<td>010</td>
<td>AC</td>
</tr>
<tr>
<td>$AD$</td>
<td>0.330 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$BA$</td>
<td>0.220 nats $\in \mathcal{F}_2(0.3)$</td>
<td>011</td>
<td>BA</td>
</tr>
<tr>
<td>$BB$</td>
<td>0.076 nats $\in \mathcal{F}_2(0.3)$</td>
<td>100</td>
<td>BB</td>
</tr>
<tr>
<td>$BC$</td>
<td>0.127 nats $\in \mathcal{F}_2(0.3)$</td>
<td>101</td>
<td>BC</td>
</tr>
<tr>
<td>$BD$</td>
<td>0.473 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$CA$</td>
<td>0.017 nats $\in \mathcal{F}_2(0.3)$</td>
<td>110</td>
<td>CA</td>
</tr>
<tr>
<td>$CB$</td>
<td>0.127 nats $\in \mathcal{F}_2(0.3)$</td>
<td>111</td>
<td>CB</td>
</tr>
<tr>
<td>$CC$</td>
<td>0.330 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$CD$</td>
<td>0.676 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$DA$</td>
<td>0.330 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$DB$</td>
<td>0.473 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$DC$</td>
<td>0.676 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
<tr>
<td>$DD$</td>
<td>1.023 nats $\notin \mathcal{F}_2(0.3)$</td>
<td>000</td>
<td>ambiguous</td>
</tr>
</tbody>
</table>
Block Codes for DMS

**Theorem 3.4 (Shannon-McMillan theorem)** Given a DMS and any $\delta > 0$, $\mathcal{F}_n(\delta)$ satisfies

1. $P_{X^n}(\mathcal{F}_n^c(\delta)) < \delta$ for sufficiently large $n$.
2. $|\mathcal{F}_n(\delta)| > (1 - \delta)e^{n(H(X) - \delta)}$ for sufficiently large $n$, and $|\mathcal{F}_n(\delta)| < e^{n(H(X) + \delta)}$ for every $n$.
3. If $x^n \in \mathcal{F}_n(\delta)$, then
   
   $$e^{-n(H(X) + \delta)} < P_{X^n}(x^n) < e^{-n(H(X) - \delta)}.$$

**Proof:** Property 3 is an immediate consequence of the definition of $\mathcal{F}_n(\delta)$. I.e.,

$$\mathcal{F}_n(\delta) \triangleq \left\{ x^n \in \mathcal{X}^n : \left| -\frac{1}{n} \sum_{i=1}^{n} \log P_X(x_i) - H(X) \right| < \delta \right\}.$$

Thus,

$$\left| -\frac{1}{n} \sum_{i=1}^{n} \log P_X(x_i) - H(X) \right| < \delta \iff \left| -\frac{1}{n} \log P_{X^n}(x^n) - H(X) \right| < \delta$$

$$\iff H(X) - \delta < -\frac{1}{n} \log P_{X^n}(x^n) < H(X) + \delta.$$
Block Codes for DMS

For Property 1, we observe that by Chebyshev’s inequality, 

\[ P_{X^n}(\mathcal{F}_n^c(\delta)) = P_{X^n} \left\{ x^n \in \mathcal{X}^n : \left| \frac{1}{n} \log P_{X^n}(x^n) - H(X) \right| \geq \delta \right\} \leq \frac{\sigma^2}{n\delta^2} < \delta, \]

for \( n > \sigma^2/\delta^3 \), where

\[ \sigma^2 = \sum_{x \in \mathcal{X}} P_X(x) (\log P_X(x))^2 - H(X)^2 \]

is a constant independent of \( n \).

To prove Property 2, we have

\[ 1 \geq \sum_{x^n \in \mathcal{F}_n(\delta)} P_{X^n}(x^n) > \sum_{x^n \in \mathcal{F}_n(\delta)} e^{-n(H(X)+\delta)} = |\mathcal{F}_n(\delta)|e^{-n(H(X)+\delta)}, \]

and, using Property 1,

\[ 1 - \delta \leq 1 - \frac{\sigma^2}{n\delta^2} \leq \sum_{x^n \in \mathcal{F}_n(\delta)} P_{X^n}(x^n) < \sum_{x^n \in \mathcal{F}_n(\delta)} e^{-n(H(X)-\delta)} = |\mathcal{F}_n(\delta)|e^{-n(H(X)-\delta)}, \]

for \( n \geq \sigma^2/\delta^3 \).
In the proof, we assume that $\sigma^2 = \text{Var}[-\log P_X(X)] < \infty$. This is true for finite alphabet:

$$\text{Var}[-\log P_X(X)] \leq E[(\log P_X(X))^2] = \sum_{x \in \mathcal{X}} P_X(x)(\log P_X(x))^2$$

$$\leq \sum_{x \in \mathcal{X}} 0.5414 = 0.5414 \times |\mathcal{X}| < \infty.$$
Theorem 3.5 (Shannon’s source coding theorem) Fix a DMS 

\[ X = \{X^n = (X_1, X_2, \ldots, X_n)\}_{n=1}^\infty \]

with marginal entropy \( H(X_i) = H(X) \), and \( \varepsilon > 0 \) arbitrarily small. There exists \( \delta \) with \( 0 < \delta < \varepsilon \), and a sequence of block codes \( \{c_n = (n, M_n)\}_{n=1}^\infty \) with

\[
\frac{1}{n} \log M_n < H(X) + \delta
\]

such that

\[ P_e(c_n) < \varepsilon \quad \text{for all sufficiently large } n, \]

where \( P_e(c_n) \) denotes the probability of decoding error for block code \( c_n \).

**Keys of the proof**

- Only need to prove the existence of such block code.
- The code chosen is indeed the weakly \( \delta \)-typical set.
Shannon’s Source Coding Theorem

**Proof:** Fix \( \delta \) satisfying \( 0 < \delta < \varepsilon \). Binary-index the source symbols in \( \mathcal{F}_n(\delta/2) \) starting from one.

For \( n \leq 2 \log(2)/\delta \), pick any \( \mathcal{C}_n = (n, M_n) \) block code satisfying

\[
\frac{1}{n} \log M_n < H(X) + \delta.
\]

For \( n > 2 \log(2)/\delta \), choose \( \mathcal{C}_n = (n, M_n) \) block encoder as

\[
\begin{cases}
  x^n \rightarrow \text{binary index of } x^n, & \text{if } x^n \in \mathcal{F}_n(\delta/2); \\
  x^n \rightarrow \text{all-zero codeword}, & \text{if } x^n \not\in \mathcal{F}_n(\delta/2).
\end{cases}
\]

Then by Shannon-McMillan theorem, we obtain

\[
M_n = |\mathcal{F}_n(\delta/2)| + 1 < e^{n(H(x)+\delta/2)} + 1 < 2e^{n(H(X)+\delta/2)} < e^{n(H(X)+\delta)},
\]

for \( n > 2 \log(2)/\delta \). Hence, a sequence of \( (n, M_n) \) block code satisfying

\[
\frac{1}{n} \log M_n < H(X) + \delta.
\]

is established. It remains to show that the error probability for this sequence of \( (n, M_n) \) block code can be made smaller than \( \varepsilon \) for all sufficiently large \( n \).
Shannon’s Source Coding Theorem

By Shannon-McMillan theorem,

\[ P_{X^n}(\mathcal{F}^c_n(\delta/2)) < \frac{\delta}{2} \]

for all sufficiently large \( n \).

Consequently, for those \( n \) satisfying the above inequality, and being bigger than \( 2 \log(2)/\delta \),

\[ P_e(\mathcal{E}_n) \leq P_{X^n}(\mathcal{F}^c_n(\delta/2)) < \delta \leq \varepsilon. \]

\( \square \)
Code Rates for Data Compression

- Ultimate data compression rate

\[ R \equiv \limsup_{n \to \infty} \frac{1}{n} \log M_n \text{ nats per source symbol.} \]

- Shannon’s source coding theorem

  - Arbitrary good performance can be achieved by extending the block-length.

  \[(\forall \varepsilon > 0 \text{ and } 0 < \delta < \varepsilon)(\exists \mathcal{C}_n) \text{ such that } \frac{1}{n} \log M_n < H(X) + \delta \text{ and } P_e(\mathcal{C}_n) < \varepsilon.\]

So \( R = \limsup_{n \to \infty} \frac{1}{n} \log M_n \) can be made smaller than \( H(X) + \delta \) for arbitrarily small \( \delta \).

In other words, at rate \( R < H(X) + \delta \) for arbitrarily small \( \delta > 0 \), the error probability can be made arbitrarily zero \( (< \varepsilon) \).

- How about further making \( R < H(X) \)? Answer:

\[
\left( (\forall \{\mathcal{C}_n\}_{n \geq 1} \text{ with } \limsup_{n \to \infty} \frac{1}{n} \log |\mathcal{C}_n| < H(X)) \right) \quad P_e(\mathcal{C}_n) \to 1.
\]
Strong Converse Theorem

**Theorem 3.6 (strong converse theorem)** Fix a DMS

\[ X = \{X^n = (X_1, X_2, \ldots, X_n)\}_{n=1}^{\infty} \]

with marginal entropy \( H(X_i) = H(X) \), and \( \varepsilon > 0 \) arbitrarily small. For any block code sequence of rate \( R < H(X) \) and sufficiently large blocklength \( n \),

\[ P_e > 1 - \varepsilon. \]

**Proof:** Fix any sequence of block codes \( \{\mathcal{C}_n\}_{n=1}^{\infty} \) with

\[ R = \limsup_{n \to \infty} \frac{1}{n} \log |\mathcal{C}_n| < H(X). \]

Let \( \mathcal{S}_n = \mathcal{S}_n(\mathcal{C}_n) \) be the set of source symbols that can be correctly decoded through \( \mathcal{C}_n \)-coding system (cf. slide I: 3-21). Then \( |\mathcal{S}_n| = |\mathcal{C}_n| \).

By choosing \( \delta \) small enough with \( \varepsilon/2 > \delta > 0 \), and also by definition of limsup operation, we have

\[ (\exists N_0)(\forall n > N_0) \quad \frac{1}{n} \log |\mathcal{S}_n| = \frac{1}{n} \log |\mathcal{C}_n| < H(X) - 2\delta, \]

which implies

\[ |\mathcal{S}_n| < e^{n(H(X)-2\delta)}. \]

Furthermore, from Property 1 of Shannon-McMillan Theorem, we obtain

\[ (\exists N_1)(\forall n > N_1) \quad P_{X^n} (\mathcal{F}_n^c(\delta)) < \delta. \]
Consequently, for $n > N \triangleq \max\{N_0, N_1, \log(2/\varepsilon)/\delta\}$, the probability of correctly block decoding satisfies

$$1 - P_e(\mathcal{C}_n) = \sum_{x^n \in \mathcal{S}_n} P_{X^n}(x^n)$$

$$= \sum_{x^n \in \mathcal{S}_n \cap \mathcal{F}_n^c(\delta)} P_{X^n}(x^n) + \sum_{x^n \in \mathcal{S}_n \cap \mathcal{F}_n(\delta)} P_{X^n}(x^n)$$

$$\leq P_{X^n}(\mathcal{F}_n^c(\delta)) + |\mathcal{S}_n \cap \mathcal{F}_n(\delta)| \cdot \max_{x^n \in \mathcal{F}_n(\delta)} P_{X^n}(x^n)$$

$$< \delta + |\mathcal{S}_n| \cdot \max_{x^n \in \mathcal{F}_n(\delta)} P_{X^n}(x^n)$$

$$< \frac{\varepsilon}{2} + e^{-n(H(X)-2\delta)} \cdot e^{-n(H(X)-\delta)}$$

$$< \frac{\varepsilon}{2} + e^{-n\delta}$$

$$< \varepsilon,$$

which is equivalent to $P_e(\mathcal{C}_n) > 1 - \varepsilon$ for $n > N$. \qed
Strong Converse Theorem

- Possible codebook \( \mathcal{C}_n \) and its corresponding \( \mathcal{S}_n \). The solid box indicates the decoding mapping from \( \mathcal{C}_n \) back to \( \mathcal{S}_n \).

**Example of \( \mathcal{S}_n \)**

Source Symbols

Codeword Set \( \mathcal{C}_n \)
Summary of Shannon’s Source Coding Theorem

- Behavior of error probability as blocklength \( n \rightarrow \infty \) for a DMS

\[
\begin{align*}
    P_e \xrightarrow{n \rightarrow \infty} 1 & \quad \text{for all block codes} \\
    P_e \xrightarrow{n \rightarrow \infty} 0 & \quad \text{for the best data compression block code}
\end{align*}
\]

- Key to the achievability proof

\[
\text{Existence of a typical-like set } \mathcal{A}_n = \{x_1^n, x_2^n, \ldots, x_M^n\} \text{ with } M \approx e^{nH(X)} \text{ and } P_{X^n}(\mathcal{A}_n^c) \rightarrow 0 \ (\text{or } P_{X^n}(\mathcal{A}_n) \rightarrow 1.)
\]

In words,

- This is the basic idea for the generalization of Shannon’s source coding theorem to a more general (than i.i.d.) source.
Summary of Shannon’s Source Coding Theorem

• Notes to the strong converse theorem

  – It is named the strong converse theorem because the result is very strong.
    * All code sequences with $R < H(X)$ have error probability approaching 1!
  – The strong converse theorem applies to all stationary-ergodic sources.

• A weak converse statement (than the strong converse) is:

  – For general sources, such as non-stationary non-ergodic sources, we can find some code sequence with $R < H(X)$ whose error probability is bounded away from zero, and does not approach 1 at all.
    * Of course, you can always design a lousy code with error probability approaching 1. Here, what the theorem truly claims is that all designs are lousy.
Recall that the merit of the stationary ergodic assumption is on its validity of law of large numbers.

However, in order to extend the Shannon’s source coding theorem to stationary ergodic sources, we need to generalize the information measure for such sources.

**Definition 3.7 (entropy rate)** The entropy rate for a source $X$ is defined by

$$\lim_{n \to \infty} \frac{1}{n} H(X^n),$$

provided the limit exists.

*Comment:* The limit of $\lim_{n \to \infty} \frac{1}{n} H(X^n)$ exists for all stationary sources.
Lemma 3.8 For a stationary source, the conditional entropy

\[ H(X_n|X_{n-1}, \ldots, X_1) \]

is non-increasing in \( n \) and also bounded from below by zero. Hence by Lemma A.25 (i.e., convergence of monotone sequence), the limit

\[ \lim_{n \to \infty} H(X_n|X_{n-1}, \ldots, X_1) \]

exists.

Proof:

\[ H(X_n|X_{n-1}, \ldots, X_1) \leq H(X_n|X_{n-1}, \ldots, X_2) \]  \hspace{1cm} (3.2.2)

\[ = H(X_{n-1}|X_{n-2}, \ldots, X_1), \]  \hspace{1cm} (3.2.3)

where (3.2.2) follows since conditioning never increases entropy, and (3.2.3) holds because of the stationarity assumption. \( \square \)
Lemma 3.9 (Cesáro-mean theorem) If \( a_n \to a \) and \( b_n = \frac{1}{n} \sum_{i=1}^{n} a_i \), then \( b_n \to a \) as \( n \to \infty \).

Proof: \( a_n \to a \) implies that for any \( \varepsilon > 0 \), there exists \( N \) such that for all \( n > N \), \( |a_n - a| < \varepsilon \). Then

\[
|b_n - a| = \left| \frac{1}{n} \sum_{i=1}^{n} (a_i - a) \right|
\leq \frac{1}{n} \sum_{i=1}^{n} |a_i - a|
\leq \frac{1}{n} \sum_{i=1}^{N} |a_i - a| + \frac{1}{n} \sum_{i=N+1}^{n} |a_i - a|
\leq \frac{1}{n} \sum_{i=1}^{N} |a_i - a| + \frac{n - N}{n} \varepsilon.
\]

Hence, \( \lim_{n \to \infty} |b_n - a| \leq \varepsilon \). Since \( \varepsilon \) can be made arbitrarily small, the lemma holds. \( \square \)
Block Codes for Stationary Ergodic Sources

**Theorem 3.10** For a stationary source, its entropy rate always exists and is equal to

\[
\lim_{n \to \infty} \frac{1}{n} H(X^n) = \lim_{n \to \infty} H(X_n|X_{n-1}, \ldots, X_1).
\]

**Proof:** This theorem can be proved by writing

\[
\frac{1}{n} H(X^n) = \frac{1}{n} \sum_{i=1}^{n} H(X_i|X_{i-1}, \ldots, X_1)
\]

(chaír-rule for entropy)

and applying Cesaro-Mean theorem.

\[\square\]

**Exercise 3.11** \((1/n) H(X^n)\) is non-increasing in \(n\) for a stationary source.
Practices of Finding the Entropy Rate

• I.i.d. source

\[ \lim_{n \to \infty} \frac{1}{n} H(X^n) = H(X) \]

since \( H(X^n) = n \times H(X) \) for every \( n \).

• First-order stationary Markov source

\[ \lim_{n \to \infty} \frac{1}{n} H(X^n) = \lim_{n \to \infty} H(X_n|X_{n-1}, \ldots, X_1) = H(X_2|X_1), \]

where

\[ H(X_2|X_1) \triangleq -\sum_{x_1 \in \mathcal{X}} \sum_{x_2 \in \mathcal{X}} \pi(x_1) P_{X_2|X_1}(x_2|x_1) \cdot \log P_{X_2|X_1}(x_2|x_1), \]

and \( \pi(\cdot) \) is the stationary distribution for the Markov source.

− In addition, if the Markov source is also binary,

\[ \lim_{n \to \infty} \frac{1}{n} H(X^n) = \frac{\beta}{\alpha + \beta} H_b(\alpha) + \frac{\alpha}{\alpha + \beta} H_b(\beta), \]

where \( H_b(\alpha) \triangleq -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha) \) is the binary entropy function, and \( P_{X_2|X_1}(0|1) = \alpha \) and \( P_{X_2|X_1}(1|0) = \beta \)
Shannon’s Source Coding Theorem Revisited

**Theorem 3.12 (generalized AEP or Shannon-McMillan-Breiman Theorem)** If $X_1, X_2, \ldots, X_n, \ldots$ are stationary-ergodic, then

$$-rac{1}{n} \log P_{X^n}(X_1, \ldots, X_n) \xrightarrow{a.s.} \lim_{n \to \infty} \frac{1}{n} H(X^n).$$

**Theorem 3.13 (Shannon’s source coding theorem for stationary-ergodic sources)** Fix a stationary-ergodic source

$$X = \{X^n = (X_1, X_2, \ldots, X_n)\}_{n=1}^\infty$$

with entropy rate

$$H \triangleq \lim_{n \to \infty} \frac{1}{n} H(X^n),$$

and $\varepsilon > 0$ arbitrarily small. There exists $\delta$ with $0 < \delta < \varepsilon$ and a sequence of block codes $\{\mathcal{C}_n = (n, M_n)\}_{n=1}^\infty$ with

$$\frac{1}{n} \log M_n < H + \delta,$$

such that

$$P_e(\mathcal{C}_n) < \varepsilon \quad \text{for all sufficiently large } n,$$

where $P_e(\mathcal{C}_n)$ denotes the probability of decoding error for block code $\mathcal{C}_n$. 
Shannon’s Source Coding Theorem Revisited

**Theorem 3.14 (strong converse theorem)** Fix a stationary-ergodic source

\[ X = \{X^n = (X_1, X_2, \ldots, X_n)\}_{n=1}^\infty \]

with entropy rate \( H \), and \( \varepsilon > 0 \) arbitrarily small. For any block code of rate \( R < H(X) \) and sufficiently large blocklength \( n \), the probability of block decoding failure \( P_e \) satisfies

\[ P_e > 1 - \varepsilon. \]
Problems of Ergodicity Assumption

- In general, it is hard to check whether a process is ergodic or not.
- A specific case that ergodicity can be verified is that of the Markov sources.

Observation 3.15

1. An irreducible finite-state Markov source is ergodic.
   - Note that irreducibility can be verified in terms of the transition probability matrix. For example, all the entries in transition probability matrix are non-zero.

2. The generalized AEP theorem holds for irreducible stationary Markov sources. For example, if the Markov source is of the first-order, then

\[
-\frac{1}{n} \log P_{X^n}(X^n) \xrightarrow{a.s.} \lim_{n \to \infty} \frac{1}{n} H(X^n) = H(X_2|X_1).
\]
A source can be compressed only when it has redundancy.

- A very important concept is that the output of a perfect lossless data compressor should be i.i.d. with completely uniform marginal distribution. Because if it were not so, there would be redundancy in the output and hence the compressor cannot be claimed perfect.

- This arises the need to define the redundancy of a source.

- Categories of redundancy
  - intra-sourceword redundancy
    * due to non-uniform marginal distribution
  - inter-sourceword redundancy
    * due to the source memory
Redundancy for Lossless Data Compression

**Definition 3.16 (redundancy)**

1. The *redundancy* of a stationary source due to non-uniform marginals is

   \[ \rho_D \triangleq \log |\mathcal{X}| - H(X_1). \]

2. The *redundancy* of a stationary source due to source memory is

   \[ \rho_M \triangleq H(X_1) - \lim_{n \to \infty} \frac{1}{n} H(X^n). \]

3. The *total redundancy* of a stationary source is

   \[ \rho_T \triangleq \rho_D + \rho_M = \log |\mathcal{X}| - \lim_{n \to \infty} \frac{1}{n} H(X^n). \]
Redundancy for Lossless Data Compression

E.g.

<table>
<thead>
<tr>
<th>Source</th>
<th>$\rho_D$</th>
<th>$\rho_M$</th>
<th>$\rho_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>i.i.d. uniform</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>i.i.d. non-uniform</td>
<td>$\log</td>
<td>\mathcal{X}</td>
<td>- H(X_1)$</td>
</tr>
<tr>
<td>stationary first-order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>symmetric Markov</td>
<td>0</td>
<td>$H(X_1) - H(X_2</td>
<td>X_1)$</td>
</tr>
<tr>
<td>stationary first-order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>non-symmetric Markov</td>
<td>$\log</td>
<td>\mathcal{X}</td>
<td>- H(X_1)$</td>
</tr>
</tbody>
</table>

- Note that a first-order Markov process is symmetric if for any $x_1$ and $\hat{x}_1$

  $$\{a : a = P_{X_2|X_1}(y|x_1) \text{ for some } y\} = \{a : a = P_{X_2|X_1}(y|\hat{x}_1) \text{ for some } y\}.$$
Variable-Length Code for Lossless Data Compression

- Non-singular codes
  - To encode all sourcewords with distinct variable-length codewords

- Uniquely decodable codes
  - Concatenation of codewords (without punctuation mechanism) can be uniquely decodable.
  
  **E.g.**, a non-singular but non-uniquely decodable code

  code of $A = 0$,
  code of $B = 1$,
  code of $C = 00$,
  code of $D = 01$,
  code of $E = 10$,
  code of $F = 11$.

  The code is not uniquely decodable because the codeword sequence, 01, can be reconstructed as either $AB$ or $D$. 
Variable-Length Code for Lossless Data Compression

**Theorem 3.17 (Kraft inequality)** A uniquely decodable code $C$ with binary code alphabet $\{0, 1\}$ and with $M$ codewords having lengths $\ell_0, \ell_1, \ell_2, \ldots, \ell_{M-1}$ must satisfy the following inequality

$$
\sum_{m=0}^{M-1} 2^{-\ell_m} \leq 1.
$$

**Proof:** Suppose we use the codebook $C$ to encode $N$ source symbols (arriving in sequence); this yields a concatenated codeword sequence

$$
c_1c_2c_3\ldots c_N.
$$

Let the lengths of the codewords be respectively denoted by

$$
\ell(c_1), \ell(c_2), \ldots, \ell(c_N).
$$

Consider the identity:

$$
\left( \sum_{c_1 \in C} \sum_{c_2 \in C} \ldots \sum_{c_N \in C} 2^{-(\ell(c_1) + \ell(c_2) + \ldots + \ell(c_N))} \right).
$$

It is obvious that the above identity is equal to

$$
\left( \sum_{c \in C} 2^{-\ell(c)} \right)^N = \left( \sum_{m=0}^{M-1} 2^{-\ell_m} \right)^N.
$$
Variable-Length Code for Lossless Data Compression

(Note that $|C| = M$.) On the other hand, all the code sequences with length

$$i = \ell(c_1) + \ell(c_2) + \cdots + \ell(c_N)$$

contribute equally to the sum of the identity, which is $2^{-i}$. Let $A_i$ denote the number of code sequences that have length $i$. Then the above identity can be re-written as

$$\left(\sum_{m=0}^{M-1} 2^{-\ell_m}\right)^N = \sum_{i=1}^{LN} A_i 2^{-i},$$

where

$$L \triangleq \max_{c \in C} \ell(c).$$

(Here, we implicitly and reasonably assume that the smallest length of the code sequence is 1.)

Since $C$ is by assumption a uniquely decodable code, the codeword sequence must be unambiguously decodable. Observe that a code sequence with length $i$ has at most $2^i$ unambiguous combinations. Therefore, $A_i \leq 2^i$, and

$$\left(\sum_{m=0}^{M-1} 2^{-\ell_m}\right)^N = \sum_{i=1}^{LN} A_i 2^{-i} \leq \sum_{i=1}^{LN} 2^i 2^{-i} = LN,$$
which implies that
\[ \sum_{m=0}^{M-1} 2^{-\ell_m} \leq (LN)^{1/N}. \]

The proof is completed by noting that the above inequality holds for every \( N \), and the upper bound \((LN)^{1/N}\) goes to 1 as \( N \) goes to infinity.
Theorem 3.18 The average binary codeword length of every uniquely decodable code of a source is lower bounded by the source entropy (measured in bits.)

Proof: Let the source be modelled as a random variable $X$, and denote the associated source symbol by $x$. The codeword for source symbol $x$ and its length are respectively denoted as $c_x$ and $\ell(c_x)$. Hence,

$$\sum_{x \in \mathcal{X}} P_X(x) \ell(c_x) - H(X) = \sum_{x \in \mathcal{X}} P_X(x) \ell(c_x) - \sum_{x \in \mathcal{X}} (-P_X(x) \log_2 P_X(x))$$

$$= \frac{1}{\log(2)} \sum_{x \in \mathcal{X}} P_X(x) \log \frac{P_X(x)}{2^{-\ell(c_x)}}$$

$$\geq \frac{1}{\log(2)} \left[ \sum_{x \in \mathcal{X}} P_X(x) \right] \log \left[ \frac{\sum_{x \in \mathcal{X}} P_X(x)}{\sum_{x \in \mathcal{X}} 2^{-\ell(c_x)}} \right]$$

$log$-sum inequality

$$= -\frac{1}{\log(2)} \log \left[ \sum_{x \in \mathcal{X}} 2^{-\ell(c_x)} \right]$$

$$\geq 0.$$
Summaries for Uniquely Decodability

1. Uniquely decodability $\Rightarrow$ the Kraft inequality.

2. Uniquely decodability $\Rightarrow$
   
   average codeword length of variable-length codes $\geq H(X)$.

Exercise 3.19

1. Find a non-singular and also non-uniquely decodable code that violates the Kraft inequality. (Hint: Slide I: 3-35.)

2. Find a non-singular and also non-uniquely decodable code that beats the entropy lower bound. (Hint: Same as the previous one.)
A Special Case of Uniquely Decodable Codes

• Prefix codes or instantaneous codes
  – Note that a uniquely decodable code may not necessarily be decoded instantaneously.

**Definition 3.20** A code is called a *prefix code* or an *instantaneous code* if no codeword is a prefix of any other codeword.
The codewords are those residing on the leaves, which in this case are 00, 01, 10, 110, 1110 and 1111.
Classification of Variable-Length Codes

- Non-singular codes
  - Uniquely decodable codes
    - Prefix codes
Observation 3.21 (prefix code to Kraft inequality) There exists a binary prefix code with $M$ codewords of length $\ell_m$ for $m = 0, \ldots, M - 1$ if, and only if, the Kraft inequality holds.

Proof:
1. [The forward part] Prefix codes satisfy the Kraft inequality.

The codewords of a prefix code can always be put on a tree. Pick up a length

$$\ell_{\text{max}} \triangleq \max_{0 \leq m \leq M - 1} \ell_m.$$ 

- A tree has originally $2^{\ell_{\text{max}}}$ nodes on level $\ell_{\text{max}}$.
- Each codeword of length $\ell_m$ obstructs $2^{\ell_{\text{max}} - \ell_m}$ nodes on level $\ell_{\text{max}}$.
  
  - In other words, when any node is chosen as a codeword, all its children will be excluded from being codewords.
  
  - There are exactly $2^{\ell_{\text{max}} - \ell_m}$ excluded nodes on level $\ell_{\text{max}}$ of the tree. We therefore say that each codeword of length $\ell_m$ obstructs $2^{\ell_{\text{max}} - \ell_m}$ nodes on level $\ell_{\text{max}}$.

- Note that no two codewords obstruct the same node on level $\ell_{\text{max}}$. Hence the number of totally obstructed codewords on level $\ell_{\text{max}}$ should be less than $2^{\ell_{\text{max}}}$,
Prefix Code to Kraft Inequality

i.e.,

\[ \sum_{m=0}^{M-1} 2^{\ell_{\text{max}} - \ell_m} \leq 2^{\ell_{\text{max}}}, \]

which immediately implies the Kraft inequality:

\[ \sum_{m=0}^{M-1} 2^{-\ell_m} \leq 1. \]

This part can also be proven by stating the fact that a prefix code is a uniquely decodable code. The objective of adding this proof is to illustrate the characteristics of a tree-like prefix code.
Prefix Code to Kraft Inequality

2. [The converse part] Kraft inequality implies the existence of a prefix code.

Suppose that $\ell_0, \ell_1, \ldots, \ell_{M-1}$ satisfy the Kraft inequality. We will show that there exists a binary tree with $M$ selected nodes, where the $i^{th}$ node resides on level $\ell_i$.

- Let $n_i$ be the number of nodes (among the $M$ nodes) residing on level $i$ (i.e., $n_i$ is the number of codewords with length $i$ or $n_i = |\{m : \ell_m = i\}|$), and let

  $$\ell_{\text{max}} \triangleq \max_{0 \leq m \leq M-1} \ell_m.$$  

- Then from the Kraft inequality, we have

  $$n_1 2^{-1} + n_2 2^{-2} + \cdots + n_{\ell_{\text{max}}} 2^{-\ell_{\text{max}}} \leq 1.$$
• The above inequality can be re-written in a form that is more suitable for this proof as:

\[
\begin{align*}
n_1 2^{-1} &\leq 1 \\
n_1 2^{-1} + n_2 2^{-2} &\leq 1 \\
\cdots \\
n_1 2^{-1} + n_2 2^{-2} + \cdots + n_{\ell_{\text{max}}} 2^{-\ell_{\text{max}}} &\leq 1.
\end{align*}
\]

Hence,

\[
\begin{align*}
n_1 &\leq 2 \\
n_2 &\leq 2^2 - n_1 2^1 \\
\cdots \\
n_{\ell_{\text{max}}} &\leq 2^{\ell_{\text{max}}} - n_1 2^{\ell_{\text{max}}-1} - \cdots - n_{\ell_{\text{max}}-1} 2^1,
\end{align*}
\]

which can be interpreted in terms of a tree model as:
Prefix Code to Kraft Inequality

– the 1st inequality says that the number of codewords of length 1 is less than the available number of nodes on the 1st level, which is 2.
– The 2nd inequality says that the number of codewords of length 2 is less than the total number of nodes on the 2nd level, which is $2^2$, minus the number of nodes obstructed by the 1st level nodes already occupied by codewords.
– The succeeding inequalities demonstrate the availability of a sufficient number of nodes at each level after the nodes blocked by shorter length codewords have been removed.
– Because this is true at every codeword length up to the maximum codeword length, the assertion of the theorem is proved.
Source Coding Theorem for Variable-Length Codes

\textbf{Theorem 3.22}

1. For any prefix code, the average codeword length is no less than entropy.

2. There must exist a prefix code whose average codeword length is no greater than (entropy +1) bits, namely,

\[ \bar{\ell} \triangleq \sum_{x \in X} P_X(x) \ell(c_x) \leq H(X) + 1, \quad (3.3.1) \]

where \( c_x \) is the codeword for source symbol \( x \), and \( \ell(c_x) \) is the length of codeword \( c_x \).

\textbf{Proof:} A prefix code is uniquely decodable, and hence its average codeword length is no less than entropy (measured in bits.)

To prove the second part, we can design a prefix code satisfying both \( \bar{\ell} \leq H(X) + 1 \) and the Kraft inequality, which immediately implies the existence of the desired code by Observation 3.21 (the observation that is just proved).
Choose the codeword length for source symbol $x$ as

$$\ell(c_x) = \lceil -\log_2 P_X(x) \rceil + 1. \quad (3.3.2)$$

Then

$$2^{-\ell(c_x)} \leq P_X(x).$$

Summing both sides over all source symbols, we obtain

$$\sum_{x \in \mathcal{X}} 2^{-\ell(c_x)} \leq 1,$$

which is exactly the Kraft inequality.

On the other hand, (3.3.2) implies

$$\ell(c_x) \leq -\log_2 P_X(x) + 1,$$

which in turn implies

$$\sum_{x \in \mathcal{X}} P_X(x)\ell(c_x) \leq \sum_{x \in \mathcal{X}} \left[-P_X(x) \log_2 P_X(x)\right] + \sum_{x \in \mathcal{X}} P_X(x) = H(X) + 1.$$
Prefix Codes for Block Sourcewords

**E.g.** A source with source alphabet \(\{A, B, C\}\) and probability

\[
P_X(A) = 0.8, \quad P_X(B) = P_X(C) = 0.1
\]

has entropy

\[
-0.8 \cdot \log_2 0.8 - 0.1 \cdot \log_2 0.1 - 0.1 \cdot \log_2 0.1 = 0.92 \text{ bits}.
\]

- One of the best prefix codes for single-letter encoding of the above source is

\[
c(A) = 0, \quad c(B) = 10, \quad c(C) = 11.
\]

Then the resultant average codeword length is

\[
0.8 \times 1 + 0.2 \times 2 = 1.2 \text{ bits} \geq 0.92 \text{ bits}.
\]

The optimal variable-length code for a specific source \(X\) has average codeword length strictly larger than the source entropy.

- Now if we consider to prefix-encode two consecutive source symbols at a time, the new source alphabet becomes

\[
\{AA, AB, AC, BA, BB, BC, CA, CB, CC\},
\]

and the resultant probability is calculated by

\[
(P \forall x_1, x_2 \in \{A, B, C\}), \quad P_{X^2}(x_1, x_2) = P_X(x_1)P_X(x_2)
\]

under the assumption that the source is memoryless.
Prefix Codes for Block Sourcewords

Then one of the best prefix codes for the new source symbol pair is

\[ c(AA) = 0 \]
\[ c(AB) = 100 \]
\[ c(AC') = 101 \]
\[ c(BA) = 110 \]
\[ c(BB) = 111100 \]
\[ c(BC') = 111101 \]
\[ c(CA) = 1110 \]
\[ c(CB) = 111110 \]
\[ c(CC') = 111111. \]

The average codeword length per source symbol now becomes

\[
\frac{0.64(1 \times 1) + 0.08(3 \times 3 + 4 \times 1) + 0.01(6 \times 4)}{2} = 0.96 \text{ bits}
\]

which is closer to the per-source-symbol entropy 0.92 bits.
Prefix Codes for Block Sourcewords

**Corollary 3.23** Fix $\varepsilon > 0$, and a memoryless source $X$ with marginal distribution $P_X$. A prefix code can always be found with

$$\bar{\ell} \leq H(X) + \varepsilon,$$

where $\bar{\ell}$ is the average per-source-symbol codeword length, and $H(X)$ is the per-source-symbol entropy.

**Proof:** Choose $n$ large enough such that $1/n < \varepsilon$. Find a prefix code for $n$ concatenated source symbols $X^n$. Then there exists a prefix code satisfying

$$\sum_{x^n \in X^n} P_X^n(x^n)\ell_{x^n} \leq H(X^n) + 1,$$

where $\ell_{x^n}$ denotes the resultant codeword length of the concatenated source symbol $x^n$. By dividing both sides by $n$, and observing that $H(X^n) = nH(X)$ for a memoryless source, we obtain

$$\bar{\ell} \leq H(X) + \frac{1}{n} \leq H(X) + \varepsilon.$$
Final Note on Prefix Codes

**Corollary 3.24** A uniquely decodable code can always be replaced by a prefix code with the same average codeword length.
Huffman Codes

Observation 3.25 Give a source with source alphabet \( \{1, \ldots, K\} \) and probability \( \{p_1, \ldots, p_K\} \). Let \( \ell_i \) be the binary codeword length of symbol \( i \). Then there exists an optimal uniquely-decodable variable-length code satisfying:

1. \( p_i > p_j \) implies \( \ell_i \leq \ell_j \).
2. The two longest codewords have the same length.
3. The two longest codewords differ only in the last bit, and correspond to the two least-frequent symbols.

Proof:

- First, we note that any optimal code that is uniquely decodable must satisfy the Kraft inequality.

- In addition, for any set of codeword lengths that satisfy the Kraft inequality, there exists a prefix code who takes the same set as its set of codeword lengths.

- Therefore, it suffices to show that there exists an optimal prefix code satisfying the above three properties.
Huffman Codes

1. Suppose there is an optimal prefix code violating the observation. Then we can interchange the codeword for symbol $i$ with that for symbol $j$, and yield a better code.

2. Without loss of generality, let the probability of the source symbols satisfy

$$p_1 \leq p_2 \leq p_3 \leq \cdots \leq p_K.$$ 

Then by the first property, there exists an optimal prefix code with codeword lengths

$$\ell_1 \geq \ell_2 \geq \ell_3 \geq \cdots \geq \ell_K.$$ 

Suppose the codeword length for the two least-frequent source symbols satisfy $\ell_1 > \ell_2$. Then we can discard $\ell_1 - \ell_2$ code bits from the first codewords, and yield a better code. (From the definition of prefix codes, it is obviously that the new code is still a prefix code.)

3. Since all the codewords of a prefix code reside in the leaves (if we figure the code as a binary tree), we can interchange the siblings of two branches without changing the average codeword length. Property 2 implies that the two least-frequent codewords has the same codeword length. Hence, by repeatedly interchanging the siblings of a tree, we can result in a prefix code which meets the requirement.

\[ \square \]
Huffman Codes

• With this observation, the optimality of the Huffman coding in average codeword length is confirmed.

_Huffman encoding algorithm:_

1. Combine the two least probable source symbols into a new single symbol, whose probability is equal to the sum of the probabilities of the original two.
   • Thus we have to encode a new source alphabet of one less symbol.

   Repeat this step until we get down to the problem of encoding just two symbols in a source alphabet, which can be encoded merely using 0 and 1.

2. Go backward by splitting one of the two (combined) symbols into two original symbols, and the codewords of the two split symbols is formed by appending 0 for one of them and 1 for the other from the codeword of their combined symbol.

   Repeat this step until all the original symbols have been recovered and obtained a codeword.
Huffman Codes

Example 3.26 Consider a source with alphabet \{1, 2, 3, 4, 5, 6\} with probability 0.25, 0.25, 0.25, 0.1, 0.1, 0.05, respectively.

- Step 1:
Huffman Codes

- Step 2:

By following the Huffman encoding procedure as shown in above figure, we obtain the Huffman code as

00, 01, 10, 110, 1110, 1111.
Shannon-Fano-Elias Codes

Assume $\mathcal{X} = \{0, 1, \ldots, L - 1\}$, and $P_X(x) > 0$ for all $x \in \mathcal{X}$. Define

$$F(x) \triangleq \sum_{a \leq x} P_X(a),$$

and

$$\bar{F}(x) \triangleq \sum_{a < x} P_X(a) + \frac{1}{2} P_X(x).$$

**Encoder:** For any $x \in \mathcal{X}$, express $\bar{F}(x)$ in binary decimal, say

$$\bar{F}(x) = .c_1 c_2 \ldots c_k \ldots,$$

and take the first $k$ bits as the codeword of source symbol $x$, i.e.,

$$(c_1, c_2, \ldots, c_k),$$

where $k \triangleq \lceil \log_2(1/P_X(x)) \rceil + 1$. 
Shannon-Fano-Elias Codes

**Decoder:** Given codeword \((c_1, \ldots, c_k)\), compute the cumulative sum of \(F(\cdot)\) starting from the smallest element in \(\{0, 1, \ldots, L - 1\}\) until the first \(x\) satisfying

\[
F(x) \geq c_1 \ldots c_k.
\]

Then this \(x\) should be the original source symbol.
Shannon-Fano-Elias Codes

**Proof of decodability:** For any number $a \in [0, 1]$, let $\lfloor a \rfloor_k$ denote the operation that chops the binary representation of $a$ after $k$ bits (i.e., remove $(k + 1)^{th}$ bit, $(k + 2)^{th}$ bit, etc). Then

$$\bar{F}(x) - \lfloor \bar{F}(x) \rfloor_k < \frac{1}{2^k}. $$

Since $k = \lceil \log_2(1/P_X(x)) \rceil + 1 \geq \log_2(1/P_X(x)) + 1 \ (\Rightarrow 2^{k-1} \geq 1/P_X(x))$,

$$\frac{1}{2^k} \leq \frac{1}{2} P_X(x) = \left[ \sum_{a < x} P_X(a) + \frac{P_X(x)}{2} \right] - \sum_{a \leq x-1} P_X(a) = \bar{F}(x) - F(x - 1).$$

Hence,

$$F(x - 1) = \left[ F(x - 1) + \frac{1}{2^k} \right] - \frac{1}{2^k} \leq \bar{F}(x) - \frac{1}{2^k} < \lfloor \bar{F}(x) \rfloor_k.$$ 

In addition,

$$F(x) > \bar{F}(x) \geq \lfloor \bar{F}(x) \rfloor_k.$$ 

Consequently, $x$ is the first element satisfying 

$$ \bar{F}(x) \geq .c_1c_2 \ldots c_k.$$
Shannon-Fano-Elias Codes

Average codeword length:

\[
\bar{\ell} = \sum_{x \in \mathcal{X}} P_X(x) \left\lceil \log_2 \frac{1}{P_X(x)} \right\rceil + 1 < \sum_{x \in \mathcal{X}} P_X(x) \log_2 \frac{1}{P_X(x)} + 2 = (H(X) + 2) \text{ bits.}
\]

- We can again apply the same “concatenation” procedure to obtain:

\[
\bar{\ell}_2 < (H(X^2) + 2) \text{ bits, where } \bar{\ell}_2 \text{ is the average codeword length for paired source.}
\]

\[
\bar{\ell}_3 < (H(X^3) + 2) \text{ bits, where } \bar{\ell}_3 \text{ is the average codeword length for three-symbol source.}
\]

As a result, the average per-letter average codeword length can be made arbitrarily close to source entropy for i.i.d. source.

\[
\bar{\ell} = \frac{1}{n} \bar{\ell}_n < \frac{1}{n} H(X^n) + \frac{2}{n} = H(X) + \frac{2}{n}.
\]
Universal Lossless Variable-Length Codes

- The Huffman codes and Shannon-Fano-Elias codes can be constructed only when the source statistics is known.

- If the source statistics is unknown, is it possible to find a code whose average codeword length is arbitrary close to entropy? Yes, if “asymptotic achievability” is allowed.
Adaptive Huffman Codes

- Let the source alphabet be $\mathcal{X} \triangleq \{a_1, \ldots, a_J\}$.
- Define $N(a_i|x^n) \triangleq$ number of $a_i$ appearances in $x_1, x_2, \ldots, x_n$.
- Then the (current) relative frequency of $a_i$ is
  \[ \frac{N(a_i|x^n)}{n} \]
- Let $c_n(a_i)$ denote the Huffman codeword of source symbol $a_i$ w.r.t. distribution
  \[ \left\{ \frac{N(a_1|x^n)}{n}, \frac{N(a_2|x^n)}{n}, \ldots, \frac{N(a_J|x^n)}{n} \right\} \].
Adaptive Huffman Codes

Now suppose that $x_{n+1} = a_j$.

1. The codeword $c_n(a_j)$ is outputted.

2. Update the relative frequency for each source outcome according to:

$$\frac{N(a_j|x^{n+1})}{n+1} = \frac{n \times [N(a_j|x^n)/n] + 1}{n+1}$$

and

$$\frac{N(a_i|x^{n+1})}{n+1} = \frac{n \times [N(a_i|x^n)/n]}{n+1} \quad \text{for } i \neq j.$$
Adaptive Huffman Codes

Definition 3.27 (sibling property) A prefix code is said to have the sibling property if its codetree satisfies:

1. every node in the code tree (except for the root node) lies a sibling (i.e., tree is saturated), and

2. the node can be listed in non-decreasing order of probabilities with each node being adjacent to its sibling.

Observation 3.28 A prefix code is a Huffman code if, and only if, it satisfies the sibling property.
Adaptive Huffman Codes

E.g.

\[
\begin{align*}
  a_1(00, 3/8) & \quad \quad \quad b_0(5/8) \\
  a_2(01, 1/4) & \quad \quad \quad b_1(3/8) \\
  a_3(100, 1/8) & \quad \quad \quad b_{10}(1/4) \\
  a_4(101, 1/8) & \quad \quad \quad b_{11}(1/8) \\
  a_5(110, 1/16) & \quad \quad \quad 8/8 \\
  a_6(111, 1/16) &
\end{align*}
\]

\[
\begin{align*}
  b_0 \left( \frac{5}{8} \right) & \geq b_1 \left( \frac{3}{8} \right) \geq a_1 \left( \frac{3}{8} \right) \geq a_2 \left( \frac{1}{4} \right) \\
  & \quad \quad \quad \quad \text{sibling pair} \quad \quad \quad \quad \text{sibling pair} \\
  \geq b_{10} \left( \frac{1}{4} \right) & \geq b_{11} \left( \frac{1}{8} \right) \geq a_3 \left( \frac{1}{8} \right) \geq a_4 \left( \frac{1}{8} \right) \geq a_5 \left( \frac{1}{16} \right) \geq a_6 \left( \frac{1}{16} \right) \\
  & \quad \quad \quad \quad \text{sibling pair} \quad \quad \quad \quad \text{sibling pair} \quad \quad \quad \quad \text{sibling pair}
\end{align*}
\]
Adaptive Huffman Codes

E.g. (cont.)

- If the next observation \((n = 17)\) is \(a_3\), then its codeword 100 is outputted.
- The estimated distribution is updated as

\[
\begin{align*}
P^{(17)}_{\hat{X}}(a_1) &= \frac{16 \times (3/8)}{17} = \frac{6}{17}, \\
P^{(17)}_{\hat{X}}(a_2) &= \frac{16 \times (1/4)}{17} = \frac{4}{17} \\
P^{(17)}_{\hat{X}}(a_3) &= \frac{16 \times (1/8) + 1}{17} = \frac{3}{17}, \\
P^{(17)}_{\hat{X}}(a_4) &= \frac{16 \times (1/8)}{17} = \frac{2}{17} \\
P^{(17)}_{\hat{X}}(a_5) &= \frac{16 \times [1/(16)]}{17} = \frac{1}{17}, \\
P^{(17)}_{\hat{X}}(a_6) &= \frac{16 \times [1/(16)]}{17} = \frac{1}{17}.
\end{align*}
\]

The sibling property is no longer true; hence, the Huffman codetree needs to be updated.
Adaptive Huffman Codes

\[
\begin{align*}
  a_1(00, 6/17) & \geq b_0(10/17) \\
  a_2(01, 4/17) & \geq b_{10}(5/17) \\
  a_3(100, 3/17) & \geq b_{10}(5/17) \\
  a_4(101, 2/17) & \geq b_1(7/17) \\
  a_5(110, 1/17) & \geq b_{11}(2/17) \\
  a_6(111, 1/17) & \geq b_{11}(2/17)
\end{align*}
\]

\[
\begin{align*}
  b_0 \left( \frac{10}{17} \right) & \geq b_1 \left( \frac{7}{17} \right) \geq a_1 \left( \frac{6}{17} \right) \geq b_{10} \left( \frac{5}{17} \right) \\
  \text{sibling pair} & \\
  \geq a_2 \left( \frac{4}{17} \right) \geq a_3 \left( \frac{3}{17} \right) \geq a_4 \left( \frac{2}{17} \right) \geq b_{11} \left( \frac{2}{17} \right) \geq a_5 \left( \frac{1}{17} \right) \geq a_6 \left( \frac{1}{17} \right) \text{ sibling pair}
\end{align*}
\]

\[a_1 \text{ is not adjacent to its sibling } a_2.\]
Adaptive Huffman Codes

E.g. (cont.) The updated Huffman codetree.

\[ a_1(10, 6/17) \]
\[ a_2(00, 4/17) \]
\[ a_3(01, 3/17) \]
\[ a_4(110, 2/17) \]
\[ a_5(1110, 1/17) \]
\[ a_6(1111, 1/17) \]

\[ b_0(7/17) \]
\[ b_1(10/17) \]
\[ 17/17 \]

\[ \begin{align*}
  b_1 \left( \frac{10}{17} \right) & \geq b_0 \left( \frac{7}{17} \right) \\
  & \geq a_1 \left( \frac{6}{17} \right) \\
  & \geq b_{11} \left( \frac{4}{17} \right) \\
  a_2 \left( \frac{4}{17} \right) & \geq a_3 \left( \frac{3}{17} \right) \\
  & \geq a_4 \left( \frac{2}{17} \right) \\
  & \geq b_{111} \left( \frac{2}{17} \right) \\
  a_5 \left( \frac{1}{17} \right) & \geq a_6 \left( \frac{1}{17} \right)
\end{align*} \]
Lempel-Ziv Codes

Encoder:

1. Parse the input sequence into strings that have never appeared before.

2. Let $L$ be the number of distinct strings of the parsed source. Then we need $\log_2 L$ bits to index these strings (starting from one). The codeword of each string is the index of its prefix concatenated with the last bit in its source string.

E.g.

- The input sequence is 1011010100010;

- Step 1:
  - The algorithm first eats the first letter 1 and finds that it never appears before. So 1 is the first string.
  - Then the algorithm eats the second letter 0 and finds that it never appears before, and hence, put it to be the next string.
  - The algorithm eats the next letter 1, and finds that this string has appeared. Hence, it eats another letter 1 and yields a new string 11.
  - By repeating these procedures, the source sequence is parsed into strings as
    
    1, 0, 11, 01, 010, 00, 10.
Lempel-Ziv Codes

- Step 2:
  - $L = 8$. So the indices will be:

    \[
    \text{parsed source} : \quad 1 \quad 0 \quad 11 \quad 01 \quad 010 \quad 00 \quad 10
    \]

    \[
    \text{index} : \quad 001 \quad 010 \quad 011 \quad 100 \quad 101 \quad 110 \quad 111
    \]

  - E.g., the codeword of source string 010 will be the index of 01, i.e. 100, concatenated with the last bit of the source string, i.e. 0.

- The resultant codeword string is:

  \[
  (000, 1)(000, 0)(001, 1)(010, 1)(100, 0)(010, 0)(001, 0)
  \]

  or equivalently,

  \[
  0001000000110101100001000010.
  \]

**Theorem 3.29** The Lempel-Ziv algorithm asymptotically achieves the entropy rate of any (unknown) stationary source.

- *The “compress” Unix command is a variation of the Lempel-Ziv algorithm.*
Notes on Lempel-Ziv Codes

• The conventional Lempel-Ziv encoder requires two passes: the first pass to decide $L$, and the second pass to generate real codewords.

• The algorithm can be modified so that it requires only one pass over the source string.

• Also note that the above algorithm uses an equal number of bits—$\log_2 L$—to all the location index, which can also be relaxed by proper modifications.
Key Notes

• Average per-source-symbol codeword length versus per-source-symbol entropy
  – Average per-source-symbol codeword length is exactly the code rate for fixed-length codes.

• Categories of codes
  – Fixed-length codes (relation with segmentation or blocking)
    * Block codes
    * Fixed-length tree codes
  – Variable-length codes
    * Non-singular codes
    * Uniquely decodable codes
    * Prefix codes

• AEP theorem

• Weakly $\delta$-typical set and Shannon-McMillan theorem

• Shannon’s source coding theorem and its converse theorem for DMS
Key Notes

- Entropy rate and the proof of its existence for stationary sources
- Generalized AEP
- Shannon’s source coding theorem and its converse theorem for stationary-ergodic sources
- Redundancy of sources
- Kraft inequality and its relation to uniquely decodable codes, as well as prefix codes
- Source coding theorem for variable-length codes
- Huffman codes and adaptive Huffman codes
- Shannon-Fano-Elias codes
- Lempel-Ziv codes