Symbol Codes (§5)

Last week: Shannon's source coding theorem: $H(X)$ is "optimal" lossy compr. + "optimal" average lossless compression rate $\frac{H(X)}{2}$ large block size $\frac{H(X)}{2}$ complicated

Today's goal: Lessloss compression one symbol at a time with $H(X) \leq L \leq H(X) + 1$, where $L =$ average length of codeword per symbol.

Notation:

$S^+ = \bigcup_{i=1}^{N} S_i$ = nonempty strings over $S$

$l(w) =$ length of string $w \in S^+$

Symbol code: $C_t: A \rightarrow \{0, 1\}^+$ for alphabet $A$

Extended code:

$C_t^+: A^+ \rightarrow \{0, 1\}^+$, $C_t^+(x_1 \ldots x_n) := C_t(x_1) \ldots C_t(x_n)$

$C_t$ is called uniquely decodable (UD) if

$C_t^+(w) = C_t^+(w') \Rightarrow w = w'$ for all $w, w' \in S^+$

$C_t$ is called a prefix code if no codeword $C_t(x)$ is prefix of any other.

Any prefix code is UD!

Examples:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$P(x)$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>$\infty$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>$\frac{1}{4}$</td>
<td>10</td>
<td>01</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>C</td>
<td>$\frac{1}{8}$</td>
<td>110</td>
<td>10</td>
<td>$\infty$</td>
<td>011</td>
</tr>
<tr>
<td>D</td>
<td>$\frac{1}{8}$</td>
<td>111</td>
<td>11</td>
<td>11</td>
<td>111</td>
</tr>
</tbody>
</table>

Prefix code? $\checkmark$ $\checkmark$ $\times$ $\times$ $\checkmark$

UD? $\checkmark$ $\checkmark$ $\times$ $\times$ $\checkmark$

Average length: 1.75 2 1.25 1.75

Entropy: $H(P) = 1.75$

Defined as $L(C_t(P)) = L(C_t(x)) = \sum_{x \in \mathcal{X}} P(x) l(C_t(x)) = E[l(C_t(x))]$ usually want to minimize
Prefix codes = binary trees:

* leaves labeled by \( x \in \mathcal{X} \)
* path to leaf = codeword \( C(x) \)

What constraint is imposed by UD/prefixness?

Kraft-McMillan inequality: If \( C \) is UD then

\[
\sum_{x \in \mathcal{X}} 2^{-l(C(x))} \leq 1 \quad \text{optimal codes should saturate this "complete" code)
\]

Pf: Let \( S := \sum_{x} 2^{-l(C(x))} \) and \( l_{\text{max}} = \max_{x} C \left( l(x) \right) \). Then:

\[
S^N = \sum_{x \in \mathcal{X}_1 \cdot \cdot \cdot \cdot x_N} 2^{-l(C^{+}(x_1 \cdot \cdot \cdot \cdot x_N))} \leq N \cdot l_{\text{max}} \leq \sum_{l=0}^{\infty} 2^{-l} \cdot \# \text{codewords of length } l^\prime \leq 2^L \text{ by UD}
\]

Exponential growth \( \leq N \cdot l_{\text{max}} + 1 \quad \Rightarrow \quad S \leq 1 \).

Kraft's converse: If \( \sum_{x} 2^{-l(x)} \leq 1 \) then \( \exists \) prefix code with these lengths.

Pf: Construct as follows:

1. Order the lengths:

\[
l(x_1) \leq l(x_2) \leq \ldots \quad \text{were} \quad c_{l'} = l(x_1 x_2 \cdot \cdot \cdot x_l) \]

2. For \( k = 1, 2, \ldots \) choose \( C_k(x_k) \in \{0, 1\}^{l(x_k)} \) s.t. LHS of the \( C(x_1) \cdot \cdot \cdot C(x_k) \) is prefix. This is possible since

\[
\# \text{bitstrings of length } l(x_k) \text{ that have such prefix } \leq \sum_{i=1}^{k-1} 2^{l(x_k)-l(x_i)} < \sum_{i=1}^{k} 2^{l(x_k)-l(x_i)} = 2^{l(x_k)} \cdot \sum_{i=1}^{k} 2^{-l(x_i)} \leq 2^{l(x_k)} \sum_{x} 2^{-l(x)} \leq 2^{l(x_k)}
\]
How short can UD codes be? Need one more tool: —

**Gibbs inequality**: Let $P_i$ be prob. distributions. Then:

$$\sum_x P(x) \log \frac{1}{Q(x)} \geq H(P), \quad \text{iff } P = Q$$

**Proof**: LHS - RHS = $\sum_x P(x) \log \frac{P(x)}{Q(x)} = -\sum_x P \log P/Q$ & use Jensen. □

**Lower bound**: $L(C_1, P) \geq H(P)$ for every UD code. (And equality holds iff $l(C_i(x)) = \log \frac{1}{P(x)}$)

**Proof**: Define

$$Q(x) = \frac{2^{-l(x)}}{S}, \quad \text{where } S = \sum_x 2^{-l(C_i(x))} \quad \text{Kraft} \leq 1.$$  

**Gibbs**

$$H(P) \leq \sum_x P(x) \log \frac{1}{Q(x)} = L(C_1, P) + \log S \leq L(C_1, P) \quad \square$$

We can easily achieve this!

**Existence of good codes**: There exist codes with $L(C_1, X) \leq H(X) + 1$

**Proof**: Define $l(x) = \lceil \log \frac{1}{P(x)} \rceil$ — ie round equality condition from above!

* $\sum_x 2^{-l(x)} \leq \sum_x P(x) = 1 \implies$ prefix code exists by Kraft's converse

* $\sum_x P(x) l(x) \leq \sum_x P(x) \left( \log \frac{1}{P(x)} + 1 \right) = H(X) + 1$. □

**NB**: This code is in general **not** optimal. E.g.:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$P(x)$</th>
<th>$l(x)$</th>
<th>$C_i(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\frac{1}{3}$</td>
<td>2</td>
<td>00</td>
</tr>
<tr>
<td>B</td>
<td>$\frac{1}{3}$</td>
<td>2</td>
<td>01</td>
</tr>
<tr>
<td>C</td>
<td>$\frac{1}{3}$</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

$\log_{2}(2) = 1.585...$

Optimal prefix (and therefore UD) codes can be achieved as follows:
Huffman's coding algorithm:

Input: probability dist. P on $\mathcal{X}$

Output: binary tree corresponding to prefix code $C$ with minimal $L(C; P)$

alg: ① Start with forest of isolated leaves
      ② While more than one tree: merge two trees with smallest probabilities

Example:

<table>
<thead>
<tr>
<th>$X$</th>
<th>$P_X(x)$</th>
<th>$C(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.25</td>
<td>00</td>
</tr>
<tr>
<td>B</td>
<td>0.25</td>
<td>01</td>
</tr>
<tr>
<td>C</td>
<td>0.2</td>
<td>01</td>
</tr>
<tr>
<td>D</td>
<td>0.15</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>0.15</td>
<td>11</td>
</tr>
</tbody>
</table>

$H(P) = 2.28$...

$L(C, P) = 2.3$

Summary:

Source Coding Theorem for Prefix Codes: Let $C$ be the optimal UD/prefix code for $X \sim P$ (e.g., Huffman's). Then: $H(X) \leq L(C; X) \leq H(X) + 1$

$H(X) + 1$...ok if $\Delta$ large, but terrible when compressing bits

Remedy: Look at blocks and use AEP!

Changing symbols + local correlations very likely