Lecture 7: Source Coding and Kraft Inequality

- Codes
- Kraft inequality and consequences
Horse Racing

Which horse won?
<table>
<thead>
<tr>
<th>$p_i$</th>
<th>Code 1</th>
<th>Code 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>1/4</td>
<td>001</td>
<td>10</td>
</tr>
<tr>
<td>1/8</td>
<td>010</td>
<td>110</td>
</tr>
<tr>
<td>1/16</td>
<td>011</td>
<td>1110</td>
</tr>
<tr>
<td>1/64</td>
<td>100</td>
<td>111100</td>
</tr>
<tr>
<td>1/64</td>
<td>101</td>
<td>111101</td>
</tr>
<tr>
<td>1/64</td>
<td>110</td>
<td>111110</td>
</tr>
<tr>
<td>1/64</td>
<td>111</td>
<td>111111</td>
</tr>
</tbody>
</table>

$El_i$ 3 2

\[
H(X) = - \sum p_i \log p_i = 2 \text{bits}
\]

How to find the best code?
Codes

- Source code $C$ for a random variable $X$ is

$$C(x) : \mathcal{X} \rightarrow \mathcal{D}^*$$

$\mathcal{D}^*$: set of finite-length strings of symbol from $D$-ary alphabet $\mathcal{D}$

- Code length: $l(x)$

- Example: $C(\text{red}) = 00$, $C(\text{blue}) = 11$, $\mathcal{X} = \{\text{red, blue}\}$, $\mathcal{D} = \{0, 1\}$
Morse’s code (1836)

- A code for English alphabet of four symbols
- Developed for electric telegraph system
- \( D = \{\text{dot, dash, letter space, word space}\} \)
- Short sequences represent frequent letters
- Long sequences represent infrequent letters
International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.
Source coding applications

- Magnetic recording: cassette, hardrive, USB...
- Speech compression
- Compact disk (CD)
- Image compression: JPEG

Still an active area of research:
- Solid state hard drive
- Sensor network: distributed source coding
What defines a good code

- Non-singular:
  \[ x \neq x' \Rightarrow C(x) \neq C(x') \]

- non-singular enough to describe a single RV \( X \)

- When we send sequences of value of \( X \), without “comma” can we still uniquely decode

- Uniquely decodable if extension of the code is nonsingular

  \[ C(x_1)C(x_2) \cdots C(x_n) \]
<table>
<thead>
<tr>
<th>$X$</th>
<th>Singular</th>
<th>Nonsingular</th>
<th>Uniquely Decodable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>not</td>
<td>not</td>
<td></td>
</tr>
<tr>
<td></td>
<td>uniquely</td>
<td>uniquely</td>
<td></td>
</tr>
<tr>
<td></td>
<td>decodable</td>
<td>decodable</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>010</td>
<td>00</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>10</td>
<td>110</td>
</tr>
</tbody>
</table>

- Uniquely decodable if only one possible source string producing it

- However, we have to look at entire string to determine

- Prefix code (instantaneous code): no codeword is a prefix of any other code
All codes

Nonsingular codes

Uniquely decodable codes

Instantaneous codes
Expected code length

- Expected length $L(C')$ of a source code $C(x)$ for $X$ with pdf $p(x)$

$$L(C') = \sum_{x \in \mathcal{X}} p(x)l(x)$$

- We wish to construct instantaneous codes of minimum expected length
Kraft inequality

- By Kraft in 1949
- Coded over alphabet size $D$
- $m$ codes with length $l_1, \ldots, l_m$
- The code length of all instantaneous code must satisfy Kraft inequality

$$\sum_{i=1}^{m} D^{-l_i} \leq 1$$

- Given $l_1, \ldots, l_m$ satisfy Kraft, can construct instantaneous code
- Can be extended to uniquely decodable code (McMillan inequality)
Proof of Kraft inequality

- Consider $D$-ary tree
- Each codeword is represented by a leaf node
- Path from the root traces out the symbol
- Prefix code: no codeword is an ancestor of any other codeword on the tree
- Each code eliminates its descendants as codewords
• \( l_{\text{max}} \) be the length of longest codeword

• A codeword at level \( l_i \) has \( D^{l_{\text{max}} - l_i} \) descendants

• Descendant sets must be disjoint:

\[
\sum D^{l_{\text{max}} - l_i} \leq D^{l_{\text{max}}}
\]

\[
\Rightarrow \sum D^{-l_i} \leq 1
\]

• Converse: if \( l_1, \ldots, l_{\text{max}} \) satisfy Kraft inequality, can label first node at depth \( l_1 \), remove its descendants...

• Can extend to infinite prefix code \( l_{\text{max}} \to \infty \)
Optimal expected code length

• One application of Kraft inequality

• Expected code length of $D$-ary is lower bounded by entropy:

$$L \geq H_D(X)$$

Proof:

$$L - H_D(X) = \sum p_i l_i - \sum p_i \log_D \frac{1}{p_i}$$

$$= D(p||r) + \log_D \frac{1}{c} \geq 0$$

$$r_i = D^{-l_i}/\sum_j D^{-l_j}, \quad c = \sum D^{-l_i} \leq 1$$
• Achieve minimum code length if
  – \( c = 1 \): Kraft inequality is equality
  – \( r_i = p_i \): approximated pdf using \( D \)-ary alphabet is exact

• How to construct such an optimal code?

• Finding the \( D \)-adic distribution that is closest to distribution of \( X \)

• Construct the code by converse of Kraft inequality
Construction of optimal codes

• Finding the $D$-adic distribution that is closest to distribution of $X$ is impractical because finding the closest $D$-adic distribution is not obvious.

• Good suboptimal procedure
  – Shannon-Fano coding
  – Arithmetic coding

• Optimal procedure: Huffman coding
First step: finding optimal code length

- Solving optimization problem

\[
\min_{l_i} \sum_{i=1}^{m} p_i l_i \\
\text{subject to } \sum_{i=1}^{m} D^{-l_i} \leq 1.
\]

- Solve using Lagrangian multiplier

\[
J = \sum_{i=1}^{m} p_i l_i + \lambda \left( \sum_{i=1}^{m} D^{-l_i} - 1 \right)
\]
• Solution:
  \[ l^*_i = - \log_D p_i. \]

• Achieves the lower bound:
  \[ L^* = \sum p_i l^*_i = - \sum p_i \log_D p_i = H_D(X). \]

• Problem: \(- \log_D p_i\) may not be an integer!

• Rounding up
  \[ l_i = \lceil - \log_D p_i \rceil. \]
  may not be optimal.

• Usable code constructions?
Summary

- Nonsingular $>$ Uniquely decodable $>$ Instantaneous codes
- Kraft inequality for Instantaneous code
- Entropy is lower bound on expected code length