Recitation 2

1) Information

\[ I = \log_2 \left( \frac{1}{p} \right) \text{ bits} \]

- Measure of uncertainty

- More info revealed by less probable event outcome!

2) Information conveyed or revealed by announcing independent events \( A \) and \( B \):

\[ I_{AB} = \log_2 \left( \frac{1}{P_A P_B} \right) = \log_2 \left( \frac{1}{P_A} \right) + \log_2 \left( \frac{1}{P_B} \right) = I_A + I_B \text{ ~ information additive! over independent events} \]

3) Entropy

- Expected (or average) information over all possible outcomes.

\[ H(X) = E[I(X)] = \sum_{x} p(x) \log_2 \left( \frac{1}{p(x)} \right) \text{ bits.} \]

Case 1: Certain event!

\[ \text{Log probability } 0 \text{ bits to entropy!} \sim (1) \log_2(1) = 0 \]

Case 2: Likewise, uncertain (or impossible) event

\[ \text{Log probability } \log_2(\frac{1}{2}) \rightarrow \text{ Need 1 bit(s)}! \]

\[ \therefore = 0 \text{ bits} \]
Another Example: 2 (from lecture)
Ex. 2

X is an unknown 4-bit binary number picked at random.
Y is another 4-bit number given that Hamming dist. \( d \) between X and Y is 2.

Q: How many bits of info have you been given?

Sol.: There are 2

\( \binom{4}{2} \) possibilities where 2 positions differ = \( \frac{4!}{2!2!} = 6 \)

Thus \( p = \frac{6}{16} \)

and \( I = \log_2 \left( \frac{1}{6/16} \right) = \log_2 \left( 2^{16}/6 \right) = 1.42 \text{ bits} \).

3. Compression

Use fewest numbers to transmit messages!

(\( W, P \), other considerations).

Trick Idea: Use 'smaller' code for frequently occurring symbols.

'Tongue' \( \rightarrow \) 'Rarely' \( \rightarrow \)

Most frequently occurring alphabets \( \sim E, e, a, \ldots \)

Rarest \( \sim Z, Q, \ldots \)
Huffman codes:

Property 1: Non-uniqueness

* 0/1 labels on any pair of branches in a code tree can be reversed

Ex.

A: 1/3
B: 1/3
C: 1/6
D: 1/6

- ABCD
  - 0
  - 1
  - 1/3
  - 1/3

- BCD
  - 1
  - 1/3
  - 1/3

- C
  - 1
  - 1/3
  - 1/3

- D
  - 1
  - 1/3
  - 1/3

* Huffman coding can produce different non-isomorphic code trees

Ex.

A: 1/3
B: 1/3
C: 1/6
D: 1/6

- ABCD
  - 0
  - 1
  - 1/3
  - 1/3

- CD
  - 1
  - 1/3
  - 1/3

- B
  - 0
  - 1/3
  - 1/3

- A
  - 1
  - 1/3
  - 1/3

- C
  - 1
  - 1/3
  - 1/3

- D
  - 1
  - 1/3
  - 1/3
Property 2: Huffman coding over a set of symbols with known probabilities produces a code tree whose expected length is optimal.

Huffman coding with grouped symbols:

* To group symbols into large "metasymbols" and encode those metasymbols instead
* Get some gain in compression but at a cost of encoding and decoding complexity

Example: send messages containing 1000 grades of A, B, C, D

<table>
<thead>
<tr>
<th>Size of grouping</th>
<th>Expected length for 1000 grades</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.67</td>
</tr>
<tr>
<td>2</td>
<td>16.46</td>
</tr>
<tr>
<td>3</td>
<td>16.37</td>
</tr>
<tr>
<td>4</td>
<td>16.33</td>
</tr>
</tbody>
</table>
Another Example:
The following table shows the student enrollments for the school of engineering:

<table>
<thead>
<tr>
<th>Department</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Civil &amp; Env</td>
<td>7%</td>
</tr>
<tr>
<td>Mech. Eng.</td>
<td>23%</td>
</tr>
<tr>
<td>Mat Sci</td>
<td>7%</td>
</tr>
<tr>
<td>EECS</td>
<td>38%</td>
</tr>
<tr>
<td>Chem. Eng.</td>
<td>13%</td>
</tr>
<tr>
<td>Aero &amp; Astro</td>
<td>12%</td>
</tr>
</tbody>
</table>

Q1: design a Huffman code
Q2: If you code is used to send messages containing only the encoding of the departments for each student in a group of 100 randomly chosen students. What is the average length of such messages?

A1: 

```
0 38%
    /\ 62%
   / \ 1
 0 37% 23%
    /\ 1
   / \ 25% 1
 0 14% 25%
    /\ 1
   / \ 1
 0 14% 12% 13%
    /\ 1/ 0
   / \ 0 \ 0
 0 7% 7% 12%
    /\ 1/ 0
   / \ 0 0
 0 11 0 01 010
 0 0 0 0 01
 0 0 0 1
 0 0 1
```

A2: average message length = \(100 \times \sum P_i \times (\text{length of code})\)

\[
= 100 \times (7\% \times 4 + 23\% \times 3 + 7\% \times 4 + 38\% \times 1 + 13\% \times 3 + 12\% \times 3)
\]

\[
= 230
\]