2018-02-06

Class Information

- Professor: Chris Terman(cjt@mit.edu)
- Class on implementation of digital systems
- Code to 6.004 lab = 775533

Introduction to Information

- Technology is organized using abstraction along a stack of concepts
  - 6.004 goes from digital circuits to cloud computing
- Digital systems are characterized as the flow of "information" through a circuit
  - But how do we define it?
    - **Information** = data communicated or received that reduces the ambiguity in a solution space
      - Data differs from information in that information has to provide specificity towards some kind of end result

Mathematical formulation

Given a discrete random variable X with possible states $x_i \in \{x_1, x_2, x_3, ...\}$ with each state having probibility $p_i \in \{p_1, p_2, p_3, ...\}$

The amount of information passed by specifying that the x_nth state is the actual state is

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

If you go from N equally possible states to M possible states, the probability that the actual state would be within the N elements is $p = \frac{N}{M}$.

So, for that situation,

$$I(n) = log_2\left(\frac{M}{N}\right)$$

Entropy

- **Entropy** = the average of each information value for each state in the state sapce, weighted by its its probibility

    - Represents the “average” number of bits you need to send the actual data

Encoding

- **Encoding** = a mapping between bit strings and values in the state space
    - **Fixed-length encoding** = a mapping where each bitstring that maps to a state is the same length
        * *e.g.* ASCII
    - **Variable-length encoding** = a mapping where bitstrings that map to certain states can vary by length
        * *e.g.* UTF-8
        * *How do you choose bit tokens so a bitstream can be unambiguously represented?*

Binary tree representation of encodings

- Unambiguous codes can be written as a binary tree
    - Fixed-length encodings have all end states the same distance down the tree
    - Variable-length encodings have end states that vary in difference from the root of the tree

A note on negative integer representation

- In this class, we use two’s complement encoding

Huffman Codes

- **Huffman codes** = scheme for creating encodings where the bit tokens for low-information(high probability) states are short and the bit token for high-information(low-probability) states are long
- Algorithm:
    1. Create tree with two elements with states with smallest p_i
    2. Merge that subtree with the element with the next smallest p_i
    3. Terminate once all states are assigned a place

Error Detection

- **Hamming distance** = the number of digits for which two bit strings of equal length are different
    - If a single bit flips, the hamming distance increases by one

- **Even-parity single bit check** = add a bit to the end of the