LECTURE 6

Last time:

- Kraft inequality
- optimal codes.

Lecture outline

- Huffman codes

Reading: Scts. 5.5-5.7.

Kraft inequality

Any instantaneous code C with code lengths $l_1, l_2, \ldots, l_m$ must satisfy

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

Conversely, given lengths $l_1, l_2, \ldots, l_m$ that satisfy the above inequality, there exists and instantaneous code with these codeword lengths

How do we achieve such a code in a practical fashion?

Make frequent elements short and infrequent one longer.

Huffman codes

Definition: let $\mathcal{X}$ be a set of m source symbols, let $\mathcal{D}$ be a D-ary alphabet. A Huffman code

$C_{Huff} : \mathcal{X} \mapsto \mathcal{D}^*$ is an optimum instantaneous code in which the $2+((m-2) mod (D-1))$ least likely source symbols have the same length and differ only in the last digit

Proposition: for any set of source symbols $\mathcal{X}$ with m symbols, it is possible define a Huffman code for those source symbols

Consider a binary code:

reorder the x_i in terms of decreasing probability

the two least likely symbols are x_{m-1}, x_m

Huffman codes for binary D

For the code C to be optimal, $l(x_i) \geq l(x_j)$ for $i \geq j$

for every maximal length codeword $C(x_i)$ there must a codeword $C(x_j)$ that differs only in the last bit -otherwise erase one bit while still satisfying prefix condition

to satisfy that $C(x_m)$ and $C(x_{m-1})$ differ only in the last bit: find x_i such that $C(x_m)$ and $C(x_i)$ differ only in the last bit and if $x_i \neq x_m$, swap them

repeat with code for symbols $x_1, \ldots, x_{m-2}$

How do we construct them?

Find the $q = 2 + ((m-2) mod (D-1))$ least likely source symbols $x_m, \ldots, x_{m-q+1}$

Delete these symbols from the set of source symbols and replace them with a single symbol y_{m-q}

Assign $p(y_{m-q}) = \sum_{i=m-q}^{m} p(x_i)$

Now we have new set of symbols $\mathcal{X}'$

Construct a code $C_{Huff,m-q} : \mathcal{X}' \mapsto \mathcal{D}^*$

Note: could be using arbitrary weight function instead of probability

Why does this work?

Illustrate for binary

Why does this work?

Amalgamation is not always least likely event in $\mathcal{X}'$

Why does this work?

Two questions arise:

Why is it enough to now find a Huffman code $C_{Huff,m-q}$?

Where does the the $q = 2+((m-2)mod(D-1))$ come from?

Add one more letter for the q symbols $x_m, \ldots, x_{m-q}$ with respect to $C_{Huff,m-q}$

Average length of code is average length of $C_{Huff,m-q}$, plus $p(y_{m-q}) = \sum_{i=m-q}^{m} p(x_i)$

Could we have done better by taking some unused node in $C_{Huff,m-q}$ to represent some of the $x_m, \ldots, x_{m-q}$? We'll see that this is not possible and it is related to the first question

Complete trees

Definition: a complete code tree is a finite code tree in which each intermediate node has D nodes of the next higher order stemming from it

In a complete tree the Kraft inequality is satisfied with equality

Complete trees

The number of terminal nodes in a complete code tree with alphabet size D must be of the form $D + n(D - 1)$

Smallest complete tree has D terminal nodes

When we replace a terminal node by an intermediate node, we lose one terminal node and gain D more, for a net gain of $D - 1$

Optimal codes and complete trees

Optimal code can be seen as a complete tree with some number B of unused terminal nodes

By contradiction, if there are incomplete intermediate nodes, nodes of higher order could complete intermediate nodes without adverse effect on length

$B \leq D-2$, otherwise we could swap unused terminal nodes to group $D-1$ of them, in which case we can altogether eliminate those terminal nodes

Optimal codes and complete trees

How large is B? $B + m = n(D-1) + D$ so $D-2-B$ is the remainder of dividing $m-2$ by $D-1$, or $(m-2)mod(D-1)$

$B = D - 2 - ((m-2)mod(D-1))$

That is why we first group the $q = 2+((m-2)mod(D-1))$ least likely source symbols

After we have grouped those symbols, a complete tree is needed for the remaining $m-q$ symbols plus the symbol created by the amalgamation of the least likely q symbols

Use the fact that $B + q = D$

$$\begin{aligned} m - q + 1 &= n(D-1) + D - B - q + 1 \\ &= n(D-1) + 1 \\ &= (n-1)(D-1) + D \end{aligned}$$

What happens if the unlikely events change probability?

Major change may be necessary in the code

Cannot do a good job of coding until all events have been catalogued

6.441 Information Theory
Spring 2010