LECTURE 7

Note Title 2/25/2006

TODAY: DATA COMPRESSION UPPER BOUNDS

- TIGHTNESS OF KRAFT
- ENTROPY BASED ENCODING (SHANNON)
- HUFFMAN CODING (OPTIMAL)
- OTHER SCHEMES

Review of last lecture

Definitions

- Lossless or non-singular compression
- Uniquely Decodable
- Prefix code

Limitations

Kraft

↓

(1) C is prefix-free of lengths $l_1 \dots l_N \Rightarrow \sum D^{-l_i} \leq 1$

uniquely-decodable

↑

McMillan

(2) **Entropy lower bound**:

if C is prefix-free

then $\underset{x}{E}\left[|C(x)|\right] \geq \frac{H(x)}{\log D}$

(3) if C is any code with expected length L, then $\exists$ prefix code C' with expected length $L + O(\sqrt{L})$.

(so non-prefix-codes are not much better anyway)

Today: first see Kraft is tight.

Theorem: if $l_1 \dots l_N$ satisfy $\sum D^{-l_i} \leq 1$
then $\exists$ prefix free code C with $|C(i)| = l_i$.

Proof: Back to infinite D-ary tree.

Assume $l_1 \leq l_2 \leq \dots l_N$.

Assign codewords as follows.

For $C(1)$ - assign left most node at level l_1 & delete subtree below it.

$C(2)$ - assign left most node in remaining tree at level l_2 & delete subtree below it.

⋮

To prove we don't get stuck need to prove that if

$$\sum_{i=1}^{k} D^{-l_i} \leq 1 - D^{-l_{i+1}}$$

then $\exists$ free node at level l_{i+1}.

(Simple exercise: left to class).

Entropy upper bound:

Recall proof of Entropy lower bound.

set $D^{-l_i} = q_i$ & q_i look like probabilities. Resulting length is

$$\frac{H(x)}{\log D} + D(p \| q).$$

To minimize compression length should make $q_i \sim p_i$

$$\Rightarrow D^{-l_i} \sim p_i$$

$$\Rightarrow l_i = -\frac{\log p_i}{\log D}$$

But l_i needs to be an integer?

so set $l_i = \left\lceil -\frac{\log p_i}{\log D} \right\rceil$

$$\leq -\frac{\log p_i}{\log D} + 1$$

By tightness of Kraft, there exists (prefix) code C with $|C(i)| = l_i$

Expected length of C

$$= \sum_i p_i l_i$$

$$\leq -\sum \frac{p_i \log p_i}{\log D} + \sum p_i$$

$$= \frac{H(x)}{\log D} + 1$$

Conclude

for prefix code C

$$\frac{H(x)}{\log D} \leq E[|C(x)|] \leq \frac{H(x)}{\log D} + 1$$

[Code known as Shannon code]

Notes about Shannon Code

- Not "optimal" for every X.
  (Example in Cover & Thomas)
- But nearly optimal if $H(X)$ large
- So can apply to $\overline{X} = (X_1 \dots X_n)$ & then inefficiency is very small $\left(\leq \frac{1}{n}\right)$.
- But this introduces other complications.

  Need to maintain encoding & decoding table. Their lengths $\sim 2^{H(\overline{X})} \doteq 2^{H(X)\cdot n}$.
- So if we can avoid appealing to n, it is good....
- Can do optimal "effectively".

Huffman Coding

$C : \Omega \rightarrow \{0,1\}^*$

$\Omega = \{1 \dots N\}$

with $P_1 \dots P_N$

Coding algorithm Huff $(P_1 \dots P_N)$;

if $N = 2$ then $C(1) = 0$; $C(2) = 1$;

else if $N \geq 3$

[Sort so that $P_1 \geq \dots \geq P_N$;

$C' \leftarrow$ Huff $(P_1 \dots P_{N-2}, P_{N-1} + P_N)$

Let $C(i) = C'(i)$ if $i \leq N-2$

& $C(N-1) = C'(N-1) \cdot 0$

$C(N) = C'(N) \cdot 1$

Return C ;]

Optimality of Huffman Code

Other coding schemes

Shannon-Elias-Fano

Arithmetic

} Described well in Covers Thomas.