30.08.05.

Last time;

① Kraft's inequality:

② Lemma

a) $p_j > p_k \Rightarrow l_k \leq l_j$

b) Two least probable symbols have same codeword length

c) Two longest codewords $\equiv p_m, p_{m-1}$ & they differ only in the last bit

Huffman codes:

$p_1 \geq p_2 \geq \cdots \geq p_m$

Code C_m which satisfies a) to c)

$\{p_1, p_2, \cdots, p_{m-2}, p_{m-1} + p_m\}$

C_{m-1} : Remove last bit in codewords for $m-1$ & m in C_m & assign the common prefix to the new symbol $\equiv p_{m-1} + p_m$.

$L(C_m) = \sum_i p_i l_i$

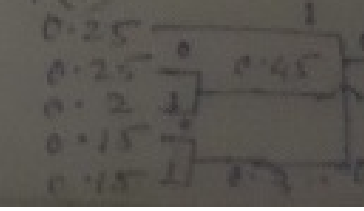
$= L(C_{m-1}) + p_{m-1} + p_m$

→ It does not depend on the code

Reduce till we have to find optimal code for 2 symbols which is {0, 1}

e.g.

	X	P(x)
01	1	0.25
10	2	0.25
11	3	0.2
000	4	0.15
001	5	0.15



ions: a) $O(|x|)$ ↑ — dependent

⇒ high complexity.

b) We need a probabilistic source model.

Arithmetic Coding

code actually depends upon the statistics in Huffman Coding.

* Lempel-Ziv Algorithm:

(Penality is that: it approaches entropy slowly, than the case when we know the statistics (Huffman Coding))

$\mathcal{X} = \{0, 1\}$.

Algo: Given a source string, we parse it into shortest phrases that have not occurred before.

1 0 11 01 01 0 0 0 1 0 (shortest phrases we have to take)

Dictionary: 1, 0, 11, 01, 010, 00, 10.

Dictionary

Index Code	Index	Phrases	Code
000 ←	0	NULL	
001	1	1	000 1
010	2	0	000 0
011	3	11	001 1 → address
100	4	01	010 1
101	5	010	100 0
110	6	00	
111	7	10	

1, 0

[Number of phrases logarithmic]

Decoded similarly decoder will also have a dictionary

No error (channel) or adversary is assumed in communication

$2^{12} = 4096$ bound

$c(n)$ = # phrases

$l(x_1^n) = c(n)\left(\log(c(n)) + 1\right)$

* Defn : A distinct parsing is a parsing of x_1^n in which no two phrases are equal

* Lemma : For a distinct parsing [This lemma does not depend upon statistics]

$$c(n) \leq \frac{n}{(1-\epsilon_n)\log(n)}$$

where $\epsilon_n = \min\left\{1, \frac{\log(\log(n)) + 4}{\log(n)}\right\}$

$\to 0$ as $n \to \infty$.

Proof:

n_k = sum of the lengths of all phrases with length $\leq k$

$$= \sum_{j=1}^{k} j 2^j = (k-1)\,2^{k+1} + 2$$

[long phrase of longer length will ?? decrease number of sequences]

$n = n_k$

$$c(n_k) < 2^{k+1} < \frac{n_k}{(k-1)}$$

~~We are trying to prove~~

$n = n_k + \Delta$

$0 \le \Delta < n_{k+1} - n_k = k\,2^{k+2} - (k-1)\,2^{k+1}$

$= 2^{k+1}\{2k - (k-1)\} = (k+1)\,2^{k+1}$

$$C(n) \le C(n_k) + \frac{\Delta}{k+1} \le \frac{n_k}{k-1} + \frac{\Delta}{k+1}$$

$$\le \frac{n_k}{k-1} + \frac{\Delta}{k-1} \le \frac{n_k + \Delta}{k-1} \le \frac{n}{k-1}$$

$$\boxed{C(n) \le \frac{n}{(k-1)}, \quad n_k \le n < n_{k+1}}$$

~~$k-1$ $n < n_{k+1}$~~

$n \ge n_k$

$n_k = (k-1)\,2^{k+1} + 2 \ge 2^k$

$k \le \log(n)$

$n < n_{k+1}$

$n_{k+1} = k\,2^{k+2} + 2 \le (k+2)\,2^{k+2}$

$$2^{k+2} \ge \frac{n}{(k+2)} \ge \frac{n}{(\log(n)+2)}$$

$$k+2 \ge \log(n) - \log(\log(n)+2)$$

$$\Rightarrow k-1 \ge \log(n)\left[1 - \frac{\log(\log(n)+2)+3}{\log(n)}\right], \quad n \ge 4$$

$$n \ge 4 \Rightarrow \quad \ge \log(n)\left[1 - \frac{\log(\log(n))+4}{\log(n)}\right] \quad \{\because \log(n)+2 \le 2\log(n)\}$$

$$= (1 - \epsilon_n)\log(n), \quad n \ge 4$$

$$\therefore C(n) \leq \frac{n}{(k-1)} \leq \frac{n}{(1-\epsilon_n)\log(n)} \quad \text{--- (*)}$$ [Combinatorial lemma

* Markov Process:

A process $\{X_n\}_{n=-\infty}^{+\infty}$ is said to be a stationary Markov Process of order k if,

a) It is stationary

b) $P(X_1^n = x_1^n \mid X_{-\infty}^{n-1} = x_{-\infty}^{n-1})$ (Markov Property)

$= P(X_1^n = x_1^n \mid X_{n-k}^{n-1} = x_{n-k}^{n-1})$ [Only last k samples will suffice]

$$H(\{X_n\}) = \lim_{n\to\infty} H(X_n \mid X_1^{n-1}) \longrightarrow H(X_0 \mid X_{-k}^{-1})$$

$= H(X_0 \mid X_{-k}^{-1})$ [Using the fact that the process is stationary]

[k-order markov process]

* $P_k(x_1^n \mid x_{-k}^{-1}) = P(X_1^n = x_1^n \mid X_{-k}^{-1} = x_{-k}^{-1})$

$= P(X_n = x_n \mid X_{n-k}^{n-1} = x_{n-k}^{n-1})$

↳ [Since Markov Process, it depends on the last k]

$P(X_1^n = x_1^{n-1} \mid x_{-k}^{-1})$

$= \prod_{j=1}^{n} P(X_j = x_j \mid X_{j-k}^{j-1} = x_{j-k}^{j-1})$

It's possible to extend a markov process of order k to ergodic process

$$= \prod_{j=1}^{n} p\left(x_j \mid x_{j-k}^{j-1}\right)$$

$$\log\left(\hat{p}_k\left(x_1^n \mid x_{-k}^{-1}\right)\right) = -\frac{1}{n}\sum_{j=1}^{n} \log\left(p\left(x_j \mid x_{j-k}^{j-1}\right)\right)$$

$$\xrightarrow{\text{Ergodicity}} -E\left[\log\left(p\left(x_0 \mid x_{-k}^{-1}\right)\right)\right]$$

$$= H\left(x_0 \mid x_{-k}^{-1}\right) \quad \text{a.s.}$$

This is A.E.P. [Not all population is important]

Suppose x_1^n is parsed by LZ into $y_1, y_2, \ldots, y_{c(n)}$

r_i = time index at which phrase y_i starts

$= x_{r_i}^{r_{i+1}-1}$

$s_i = x_{r_i-k}^{r_i-1}$,

$s_1 = x_{-(k-1)}^{0}$,

$s_i \in \mathcal{X}^k$

$x_1\ x_2\ x_3\ x_4$ (phrase)

$y_1 = x_1$, $r_1 = 1$

$r_2 = 2$

$r_3 = 4$

C_{ls} = # phrases y_i with length l and state s

$x_1\ x_2\ x_3\ x_4$ (y_2) $\quad C_{2x_1} = 1$

$l = 1, 2, \ldots$

$s \in \mathcal{X}^k$

$$c(n) = \sum_{l=1}^{\infty} \sum_{s \in \mathcal{X}^k} C_{l,s} \qquad \sum_{l,s} C$$

$$n = \sum_{l,s} l\, C_{l,s}$$

Homework due Tuesday