2/5/14 Class 7

{ ~~what is~~ how much can a signal be compress
what is the ultimate transmission rate

9.2 Uncertainty, Information and Entropy.

Finite alphabet $\varphi = \{S_0, S_1 \cdots S_{K-1}\}$

with probability $P(S = S_k) = P_k$, $\sum_{k=0}^{K-1} P_k = 1$

logarithmic function $I(S_k) = \log\left(\frac{1}{P_k}\right) = -\log_2 P_k$ (bit)

① $I(S_k) = 0$, for $P_k = 1$

② $I(S_k) \geq 0$ for $0 \leq P_k \leq 1$

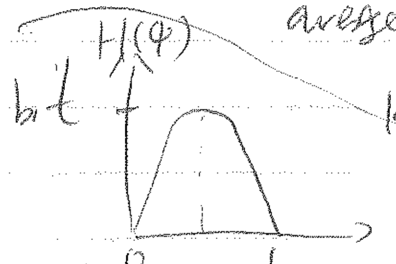
③ $I(S_k) > I(S_i)$ $P_k < P_i$

④ $I(S_k S_l) = I(S_k) + I(S_l)$ if S_k, S_l are statistically independent

Entropy: $H(\varphi) = E[I(S_k)] = \sum_{k=0}^{K-1} P_k I(S_k)$

$= \sum_{k=0}^{K-1} P_k \log_2\left(\frac{1}{P_k}\right)$

average information content per source symbol



ex: 9.1 Entropy of binary memoryless source

$K = 2$

properties of Entropy

$0 \leq H(\varphi) \leq \log_2 K$

$H(P_0) = -P_0 \log P_0 - (1-P_0)\log_2(1-P_0)$

Extension of a discrete memoryless source

n source symbol: $H(\varphi^n) = n H(\varphi)$

EX 9.2 Entropy of Extended source

$H(\varphi^2) \sum_{i=0}^{8} P(\sigma_i) \log_2 \frac{1}{P(\sigma_i)}$

$= \frac{1}{16}\log_2(16) + \frac{1}{16}\log_2(16) \cdots = 3$ bit

Table 9.1

9.3 Source coding Theorem

requirements { 1 code words are binary form
2 uniquely decodable original ~~can be~~ source sequence can be reconstructed perfectly from encoded binary sequency.

example Morse code.
~~eg~~ keyboard

average code length $\bar{L} = \sum_{k=0}^{K-1} p_k l_k$

theorem: given ~~the~~ a ~~minimal~~ discrete memoryless source of entropy $H(\phi)$ the average code-word length $\bar{L}$ for any distortionless source encoding scheme is bounded by

$$\bar{L} \geq H(\phi)$$

9.4 Data compaction: Question 1

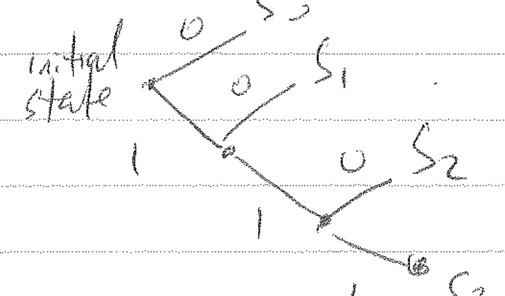
how zip works

prefix coding, Huffman coding, Lempel-Ziv coding

— prefix coding. table 9.2

Symbol	probability	code I	code II	code III
s_0	0.5	0	0	0
s_1	0.25	1	10	01
s_2	0.125	00	110	011
s_3	0.125	11	111	0111

prefix code



ex $\underbrace{10}_{s_1}\underbrace{111}_{s_3}\underbrace{110}_{s_2}\underset{s_0}{0}\underset{s_0}{0}$

Kraft-McMillian Inequality

$$\sum_{k=0}^{K-1} 2^{-l_k} \leq 1$$

$\bar{L} = 0.5 \cdot 1 + 0.25 \cdot 2 + 0.125 \cdot 3 + 0.125 \cdot 3$

given discrete memoryless source with entropy $H(\varphi)$
a prefix code can be constructed with length $\bar{L}$

$$H(\varphi) \le \bar{L} < H(\varphi)+1$$

for extended code

$$nH(\varphi) = H(\varphi^n) \le \bar{L}_n < H(\varphi^n)+1 = nH(\varphi)+1$$

$$\lim_{n\to\infty} \frac{1}{n}\bar{L}_n = H(\varphi)$$ Asymptotically achieving entropy.

- Huffman coding : ~~known~~ a ~~special~~ important class of prefix codes

Huffman encoding algorithm

1. source symbols are listed in order of decreasing probability. Two source symbols of lowest probability are assigned a 0 and 1. splitting stage
2. These two source symbols are regarded as being combined into new source symbol with probability equal to the sum of the two original probabilities (the list of source ~~ability of the new~~ symbols, and therefore source statistics, is thereby reduced in size by one) The probability of the new symbol is placed in the list in accordance with its value
3. The procedure is repeated until we are left with a final list of source statistics (symbols) of only two for which a 0 and 1 are assigned

example 9.3 Huffman Tree

symbol	stage I	stage II	stage III	stage IV
s_0	0.4	0.4	0.4	0.6 0
s_1	0.2	0.2 0	0.4 0	0.4 1
s_2	0.2	0.2	0.2 1	
s_3	0.1 0	0.2 1		
s_4	0.1 1			

symbol	probability	code word
s_0	0.4	00
s_1	0.2	10
s_2	0.2	11
s_3	0.1	010
s_4	0.1	011

$\bar{L} = 0.4\cdot 2 + 0.2\cdot 2 + 0.2\cdot 2 + 0.1\cdot 3 + 0.1\cdot 3 = 2.2$

$H(\phi) = 0.4\log_2\left(\frac{1}{0.4}\right) + 0.2\log_2\left(\frac{1}{0.2}\right) + 0.2\log_2\left(\frac{1}{0.2}\right) + 0.1\log_2\left(\frac{1}{0.1}\right) + 0.1\log_2\left(\frac{1}{0.1}\right) = 2.12193$

drawback: need probability

example: morse code

- Lempel-Ziv coding   adaptive and simpler

parsing the source data stream into segments that are the shortest subsequences not encountered previously

initial: subsequence stored 0, 1

Data to be parsed 000101110010100101

shortested subsequence not stored

step1 subsequence stored 0, 1, 00

Data to be parsed 0101110010100101

step2 subsequence stored 0, 1, 00, 01

Data to be parsed 01110010100101

~~word~~ numerical positions	1	2	3	4	5	6	7	8	9
subsequences	0	1	00	01	011	10	010	100	101
~~(Numerical representation~~	~~0~~	~~02~~	~~11~~	~~12~~	~~42~~	~~21~~	~~41~~	~~61~~	~~62~~
encoding → binary block	0000,	0001,	0010,	0011,	1001,	0100,	1000,	1100,	1101

decoding 1101, position 9 last bit 1, remaining 110 → sequence 6 (10) so decoding is 101

- file compression zip
  fixed length codes to represent a variable number of source symbols

$C(n)$ is the number of phrases in L-Z parsing

$\log C(n)$ is the number of bits

the total length of the compressed sequence is

$$C(n)(\log C(n)+1) \to nH(\varphi)$$

$$C(n) \le \frac{n}{(1-\varepsilon_n)\lg n} \qquad \varepsilon_n \to 0,\ n \to \infty$$