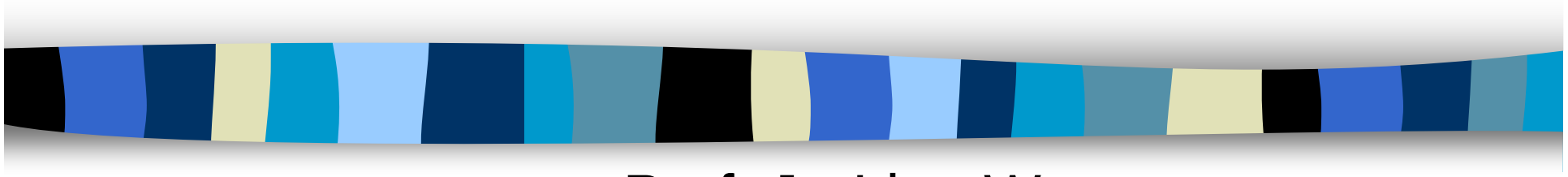
Lempel-Ziv Coding



Prof. Ja-Ling Wu

Department of Computer Science
and Information Engineering
National Taiwan University

J.Ziv and A.Lempel, “Compression of individual sequences by variable rate coding, “IEEE Trans. Information Theory, 1978.

Algorithm :

- The source sequence is sequentially parsed into strings that have not appeared so far.

  For example, 101101010010…, will be parsed as

  1,0,11,0 1,010,00,10,…

- After every comma, we look along the input sequence until we come to the shortest string that has not been marked off before. Since this is the shortest such string, all its prefixes must have occurred easier. In particular, the string consisting of all but the last bit of this string must have occurred earlier.

- We code this phrase by giving the location of the prefix and the value of the last bit.

  Let C(n) be the number of phrases in the parsing of the input n-sequence. We need log C(n) bits to describe the location of the prefix to the phrase and 1 bit to describe the last bit.

  In the above example, the code sequence is :

  (000,1), (000,0), (001,1), (010,1), (100,0), (010,0), (001,0)

  where the first number of each pair gives the index of the prefix and the 2nd number gives the last bit of the phrase.

- Decoding the coded sequence is straight forward and we can recover the source sequence without error!!

  Two passes-algorithm :

  pass-1 : parse the string and calculate C(n) Dictionary construction

  pass-2 : calculate the points and produce the code string

  Pattern Matching

- The two-passes algorithm allots an equal number of bits to all the pointers. This is not necessary, since the range of the pointers is smaller at the initial portion of the string.
  - T-A. Welch, A technique for high-performance data Compression, Computer, 1984.
  - Bell, Cleary, Witlen, Text Compression, Prentice-Hall, 1990.

Definition :

A parsing S of a binary string $x_1, x_2, \ldots, x_n$ is a division of the string into phrases, separated by commas. A distinct parsing is a parsing such that no two phrases are identical.

The L-Z algorithm described above gives a distinct parsing of the source sequence. Let C(n) denote the number of phrases in the L-Z parsing of a sequence of length n.

The compressed sequence (after applying the L-Z algorithm) consists of a list of c(n) pairs of numbers, each pair consisting of a pointer to the previous

$\log c(n)$

occurrence of the prefix of the phrase and the last bit

1

of the phrase.

the total length of the compressed sequence is : C(n)(log c(n)+1)

we will show that

$$\frac{C(n)\left(\log C\left(n\right)+1\right)}{n} \to H\left(\chi\right)$$

for a stationary ergodic sequence $X_1, X_2, \ldots, X_n$.

Lemma 1 (Lempel and Ziv) :

The number of phrase c(n) in a distinct parsing of a binary sequence X1,X2,…,Xn satisfies

$$C(n) \leq \frac{n}{(1-\varepsilon_n)\log n}$$

where $\varepsilon_n \rightarrow 0$ as $n \rightarrow \infty$.

Let $n_k = \sum_{j=1}^{k} j \cdot 2^j = (k-1)2^{k+1} + 2$

: the sum of the lengths of all distinct strings of length less than or equal to k.

C(n) : the no. of phrases in a distinct parsing of a sequence of length n

this number is maximized when all the phrases are as short as possible.

$n=n_k \rightarrow$ all the phrases are of length $\leqq$ k, thus

$$C(n) \leq \sum_{j=1}^{k} 2^j = 2^{K+1} - 2 \leq 2^{k+1} = \frac{n_k - 2}{k-1} \leq \frac{n_k}{k-1}$$

If $n_k \leqq n < n_{k+1}$, we write $n=n_k+\Delta$, then

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

Then the parsing into shortest phrases has each of the phases of length $\leqq$ k and $\frac{\Delta}{k+1}$ phrases of the length k+1.

$$C(n) = C(n_k) + \frac{\Delta}{k+1} \leq \frac{n_k}{K-1} + \frac{\Delta}{k+1} \leq \frac{n_k + \Delta}{k-1} = \frac{n}{K-1}$$

$$\Rightarrow C(n) \leq \frac{n}{K-1}$$

We now bound the size of k for a given n.

Let $n_k \leqq n < n_{k+1}$. Then

$n \geqq n_k = (k-1)2^{k+1}+2 \geqq 2^k$

$\rightarrow \quad k \leqq \log n$

$n \leqq n_{k+1} = k \cdot 2^{k+2}+2 \leqq (k+2)2^{k+2}$

$\leqq [\ (\log n+2) \cdot 2^{k+2}\ ]$

$$\Rightarrow k + 2 \geq \log \frac{n}{\log n + 2}$$

for all n $\geqq$ 4

$$k-1=(k+2)-3\geq \log n-\log(\log n+2)-3$$

$$=\left(1-\frac{\log(\log n+2)+3}{\log n}\right)\log n$$

$$\geq\left(1-\frac{\log(2\log n)+3}{\log n}\right)\log n$$

$$=\left(1-\frac{\log(\log n)+4}{\log n}\right)\log n$$

$$=(1-\varepsilon_n)\log n$$

$$\Rightarrow \quad C(n)\leq \frac{n}{K-1}=\frac{n}{(1-\varepsilon_n)\log n}$$

where $\quad \varepsilon_n=\dfrac{\log(\log n)+4}{\log n}\to 0$ as $n\to\infty$

Lemma 2 :

Let Z be a positive integer valued r.v. with mean μ .

Then the entropy H(z) is bound by

$H(z) \leqq (\mu+1)\log(\mu+1) - \mu \log \mu$

pf : H W.

Let $\{X_i\}_{i=-\infty}^{\infty}$ be a stationary ergodic process with pmf $p(x_1,x_2,\ldots x_n)$. For fixed integer k, defined the k^{th} order Markov approximation to P as

$$Q_k\left(x_{-(K-1)},\ldots x_0, x_1,\ldots, x_n\right) \underline{\underline{\Delta}} P\left(x_{-(k-1)}^{0}\right)\prod_{j=1}^{n} P\left(x_j \middle| x_{j-k}^{j-1}\right)$$

where $x_i^j \underline{\underline{\Delta}} \left(x_i, x_{i+1},\ldots, x_j\right), \quad i \le j$, and the initial state $x_{-(k-1)}^{0}$ will be part of the specification of Q_k.

Since $P\left(X_n \middle| X_{n-k}^{n-1}\right)$ is itself an ergodic process, we have

$$-\frac{1}{n}\log Q_K\left(X_1, X_2,\ldots, X_n \middle| X_{-(k-1)}^{0}\right) = -\frac{1}{n}\sum_{j=1}^{n}\log P\left(X_j \middle| X_{j-k}^{j-1}\right)$$

$$\to -E\log P\left(X_j \middle| X_{j-k}^{j-1}\right) \underline{\underline{\Delta}} H\left(X_j \middle| X_{j-k}^{j-1}\right)$$

We will bound the rate of the L-Z code by the entropy rate of the k-th order Markov approximation for all k. The entropy rate of the Markov approximation $H\left(X_j \middle| X_{j-k}^{j-1}\right)$ converges to the entropy rate of the process as $k \to \infty$ and this will prove the result.

S'pose $X_{-(k-1)}^{n} = x_{-(k-1)}^{n}$, and s'pose that x_1^n is parsed into C distinct phrases, $y_1, y_2, \ldots, y_c$. Let ν_i be the index of the start of the i-th phrase, i.e., $y_i = x_{\nu_i}^{\nu_{i+1}-1}$. For each i=1,2,…,c, defines $S_i = x_{\nu_i - k}^{\nu_i - 1}$. Thus, S_i is the k bits of x preceding y_i, of course, $S_1 = x_{-(k-1)}^{0}$

Let C_{ls} be the number of phrases y_i with length *l* and preceding state S_i=S for *l*=1,2,… and $s \in X^k$, we then have $\sum_{l,s} C_{ls} = C$

and $\sum_{l,s} lC_{ls} = n$

Lemma 3 : (Ziv's inequality)

For any distinct parsing (in particular, the L-Z parsing) of the string $x_1,x_2,\ldots,x_n$, we have

$$\log Q_k\left(x_1, x_2,..., x_n \middle| s_1\right) \le -\sum_{l,s} C_{ls} \log C_{ls}$$

Note that the right hand side does not depend on Q_k.

proof : we write

$$Q_k\left(x_1, x_2,..., x_n \middle| s_1\right) = Q\left(y_1, y_2,..., y_c \middle| s_1\right)$$

$$= \prod_{i=1}^{C} P\left(y_i \middle| s_i\right)$$

$$\text{or } \log Q_k\left(x_1, x_2, \ldots, x_n \middle| s_1\right) = \sum_{i=1}^{C} \log P\left(y_i \middle| s_i\right)$$

$$= \sum_{l,s} \sum_{i:|y_i|=l, s_i=s} \log P\left(y_i \middle| s_i\right)$$

$$= \sum_{l,s} C_{ls} \sum_{\substack{i:|y_i|=l \\ s_i=S}} \frac{1}{C_{ls}} \log P\left(y_i \middle| s_i\right)$$

$$\leq \sum_{l,s} C_{ls} \log \left(\sum_{\substack{i:|y_i|=l \\ |s_i|=S}} \frac{1}{C_{ls}} P\left(y_i \middle| s_i\right) \right)$$

Now since the y_i are distinct, we have $\sum_{\substack{i:|y_i|=l \\ |s_i|=S}} P\left(y_i \middle| s_i\right) \leq 1$

$$\Rightarrow \quad \log Q_K\left(x_1, x_2, \ldots, x_n \middle| s_1\right) \leq \sum_{l,s} C_{ls} \log \frac{1}{C_{ls}}$$

Theorem :

Let $\{X_n\}$ be a stationary ergodic process with entropy rate H(***X***), and let C(n) be the number of phrases in a distinct parsing of a sample of length n from this process. Then

$$\limsup_{n\to\infty} \frac{C(n)\log C(n)}{n} \le H(X)$$

with probability 1.

$$\text{pf}: \log Q_k\left(x_1, x_2, ..., x_n \middle| s_1\right) \le -\sum_{ls} C_{ls} \log \frac{C_{ls} \cdot C}{C}$$

$$= -C \log C - C \sum_{l,s} \frac{C_{ls}}{C} \log \frac{C_{ls}}{C}$$

writting $\Pi_{ls} = \frac{C_{ls}}{C}$, we have

$$\sum_{l,s} \Pi_{l,s} = 1, \quad \sum_{l,s} l\Pi_{l,s} = \frac{n}{C}$$

We now define r.v.'s U,V, such that

$$\Pr(U=l, V=s) = \Pi_{l,s}$$

Thus E(U)=n/c, and

$$\log Q_k(x_1, x_2, \ldots, x_n \mid s_1) \le CH(U,V) - c \log c$$

$$\Rightarrow -\frac{1}{n}\log Q_K(x_1, x_2, \ldots, x_n \mid s_1) \ge \frac{C}{n}\log C - \frac{C}{n}H(U,V)$$

Since $H(U,V) \leqq H(U) + H(V)$

and $H(V) \leqq \log |\boldsymbol{X}|^k = k$

From Lemma 2, we have

$$H(U) \le (EU+1)\log(EU+1) - EU \log EU$$

$$= \left(\frac{n}{C}+1\right)\log\left(\frac{n}{C}+1\right) - \frac{n}{C}\log\frac{n}{C}$$

$$= \log\frac{n}{C} + \left(\frac{n}{C}+1\right)\log\left(\frac{C}{n}+1\right)$$

Thus, $\frac{C}{n}H(U,V) \le \frac{C}{n}k + \frac{C}{n}\log\frac{n}{C} + O(1)$.

For a given n, the maximum of $\frac{C}{n}\log\frac{n}{C}$ is attained for the maximum value of C $\left(\text{for } \frac{C}{n} \le \frac{1}{e}\right)$. But from Lemma 1,

$$C \le \frac{n}{\log n}\left(1+O(1)\right).$$ Thus

$$\frac{C}{n}\log\frac{n}{C} \le O\left(\frac{\log\log n}{\log n}\right)$$

and therefore $\frac{C}{n}H(U,V) \to 0$ as $n \to \infty$

Therefore,

$$\frac{C(n)\log C(n)}{n} \le \frac{-1}{n}\log Q_k\left(x_1, x_2, \ldots, x_n \middle| s_1\right) + \varepsilon_k(n)$$

where ε k(n) $\rightarrow$0 as n$\rightarrow\infty$. Hence, with probability 1,

$$\limsup_{n\to\infty} \frac{C(n)\log C(n)}{n} \le \lim_{n\to\infty} -\frac{1}{n}\log Q_k\left(X_1, X_2, \ldots, X_n \middle| X_{-(k-1)}^{0}\right)$$
$$= H\left(X_0 \middle| X_{-1}, \ldots, X_{-k}\right) \to H(X) \quad as \quad k \to \infty.$$

Theorem : Let $\{X_i\}_{i=-\infty}^{\infty}$ be a stationary ergodic stochastic process. Let l(X1,X2,…Xn) be the L-Z codeword length associated with X1,X2,…,Xn. Then

$$\lim_{n\to\infty}\frac{1}{n}l(X_1,X_2,...,X_n)\le H(X) \quad \text{with probability } 1.$$

pf : $l(X_1,X_2,...,X_n)=C(n)(\log C(n)+1)$

$$\text{by Lemma1,} \quad \limsup\frac{C(n)}{n}=0$$

$$\Rightarrow \limsup_{n\to\infty}\frac{l(X_1,X_2,...,X_n)}{n}=\limsup_{n\to\infty}\left(\frac{C(n)\log C(n)}{n}+\frac{C(n)}{n}\right)$$

$$\le H(X) \quad \text{, with probability } 1.$$

The L-Z code is a universal code, in which, the code does not depend on the distribution of the source!!

Optimal Variable Length-to-Fixed Length Code

Algorithm :

step 1. 永遠分最大的 node

step 2. 分到有 2^l leaf nodes 則停

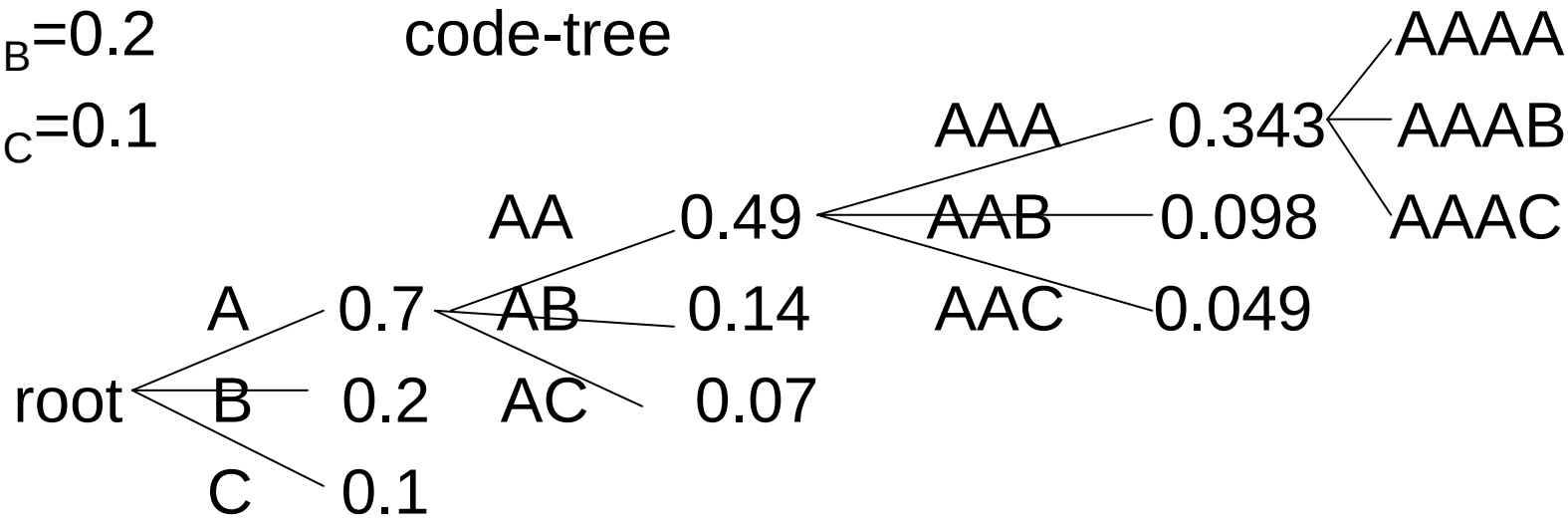
Example : Source data input : A B C C B A A A C

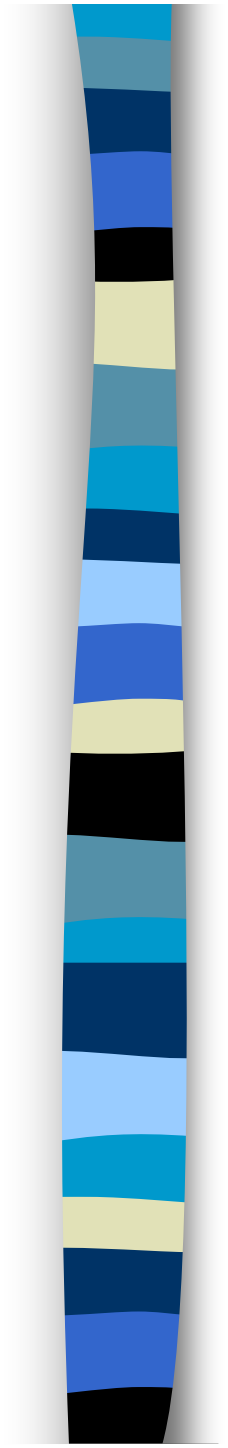
P_A=0.7

P_B=0.2

P_C=0.1

code-tree





A	1011
B	1010
C	1001
AA	1000
AB	0111
AC	0110
AAA	0101
AAB	0100
AAC	0011
AAAA	0010
AAAB	0001
AAAC	0000

A B, C, C, B, A A A C

0111 | 1001 | 1001 | 1010 | 0000

Tunstall ’77

GIT ph.D. Thesis