Ziv-Lempel 78 : A Universal Compressor

September 8, 2017

1 Compressibility Ratio

$$\rho_E(x[1\ldots n]) = \frac{L(y[1\ldots n])}{n\log|A|}. \tag{1}$$

$$\rho_{E(s)}(x[1\ldots n]) = \min_{E\in E(s)} \rho_E(x[1\ldots n]). \tag{2}$$

$$\rho_{E(s)}(x) \;=\; \limsup_{n\to\infty} \rho_{E(s)}(x[1\ldots n]). \tag{3}$$

$$\rho_E(x) \;=\; \lim_{s\to\infty} \rho_{E(s)}(x). \tag{4}$$

Why does the limit above exist?

It is clear that for every individual sequence, $0 \le \rho(x) \le 1$. Why?

$\rho(x)$ is called the *finite-state compressibility* of x.

Since ZL78 is a combinatorial, deterministic compression scheme, we resort to a combinatorial argument of its optimality. We have to argue that the compression ratio of ZL78 is at most a combinatorial quantity, and the compression ratio of any finite-state compressor with a fixed number of states is at least that combinatorial quantity.

The combinatorial concept that underlies this quantity is defined by $c(x[1\ldots n])$, the maximum number of distinct phrases that $x[1\ldots n]$ can be parsed into.

Homework

1. Consider the problem $\langle x[1\ldots n], k\rangle$ of whether $x[1\ldots n]$ can be parsed into k unique phrases. Show that this problem is NP-complete.

2 Coding Theorem

Theorem 1. *$\forall n > 0$ $\exists$ an information lossless finite-state encoder $\mathcal{E}$ with $s(n)$ states with the following properties.*

1. *For any block length n and every input block $x[1\ldots n]$, we have*

$$\rho_{\mathcal{E}}(x[1\ldots n]) \le \frac{c(x[1\ldots n]}{nlog|A|} log(2|A|(c(x[1\ldots n]) + 1)). \tag{5}$$

2. *For every finite s,*

$$\rho_{\mathcal{E}}(x[1\ldots n]) \leq \rho_{E(s)}(x[1\ldots n]) + \delta_s(n), \tag{6}$$

where

$$\lim_{n\to\infty} \delta_s(n) = 0. \tag{7}$$

3. *Given an infinite sequence x, let $\rho_E(x,n)$ denote the compression ratio attained by $\mathcal{E}$ when compressing x in successive blocks of length n. Then for any $\epsilon > 0$,*

$$\rho_{\mathcal{E}}(x,n) \leq \rho(x) + \delta_\epsilon(x,n). \tag{8}$$

where

$$\lim_{n\to\infty} \delta_\epsilon(x,n) = \epsilon. \tag{9}$$

We show that the Ziv-Lempel78 algorithm is such an encoder $\mathcal{E}$.

Proof. Suppose the incremental parsing of the input by the Ziv-Lempel78 Algorithm into unique phrases is

$$x[1\ldots n] = x[n_0+1\ldots n_1]x[n_1+1\ldots n_2]\ldots x[n_p+1\ldots n_{p+1}] \tag{10}$$

where each $x[n_{j-1}+1\ldots n_j]$ is a unique phrase. Morever, by the property of the algorithm, we know that for each such j^{th} phrase, there is a unique i^{th} phrase, $i<j$, such that $x[n_{i-1}+1\ldots n_i] = x[n_{j-1}+1\ldots n_j-1]$, *i.e.*, the longest proper prefix of the j^{th} phrase is equal to the i^{th} phrase. We will say that $\pi(j)=i$, *i.e.*, the "predecessor" of j is i.

Moreover, $n_1 = 1$.

Let us adopt the convention that $x[n_{-1}+1\ldots n_0] = \lambda$ is the first phrase the parsing of any string, and the notation that $d(w)$ for any non-empty string w is the longest proper prefix of w.

At a particular stage of the algorithm, suppose the j^{th} phrase has been identified. Let $\pi(j)$ be its predecessor. The encoding of the j^{th} phrase is the integer

$$I(x[n_{j-1}+1\ldots n_j]) = \pi(j)|A| + I_A(x[n_j]), \tag{11}$$

where $I_A : A \to \{0\ldots|A|-1\}$ is the natural mapping from letters in A to the natural numbers. Since $0 \leq \pi(j) \leq j-1$, it follows that

$$\begin{aligned} 0 \leq I(x[n_{j-1}+1\ldots n_j]) \;&=\; \pi(j)|A| + I_A(x[n_j]) \\ &\leq\quad (j-1)|A| \;+\; (|A|-1) \;=\; j|A|-1. \end{aligned} \tag{12}$$

Hence the number of bits required to encode the j^{th} phrase is $L_j = \lceil \log j|A| \rceil$.

The total length of the encoding of $x[1 \dots n]$ is

$$\begin{aligned} L &= \sum_{j=1}^{p+1} L_j \\ &= \sum_{j=1}^{p+1} \lceil \log j|A| \rceil \\ &\leq \sum_{j=1}^{p+1} \log j|A| + 1 \\ &= \sum_{j=1}^{p+1} \log(2j|A|) \\ &\leq (p+1)(\log(p+1) + \log(2|A|)). \end{aligned}$$

Now, $p \leq c(x[1 \dots n])$. It follows that

$$\rho_{\mathcal{E}}(x[1 \dots n]) \leq \frac{(c(x[1 \dots n]) + 1)\ \log[2|A|(c(x[1 \dots n]) + 1)]}{n \log |A|}. \tag{13}$$

This proves (i).

We shall prove that

$$\rho_{E(s)}(x[1 \dots n]) \quad \geq \quad \frac{(c(x[1 \dots n]) + s^2)\ \log[c(x[1 \dots n]) + s^2]}{n \log |A|} + \frac{2s^2}{n \log |A|}. \tag{14}$$

From the above two inequalities, we conclude that

$$\rho_{\mathcal{E}}(x[1 \dots n]) \ \leq \ \rho_{E(s)}(x[1 \dots n]) + \delta_s(n), \tag{15}$$

where $\lim_{n \to \infty} \delta_s(n) = 0$. This proves (ii). □

3 Converse-to-Coding Theorem

Here, we derive a lower bound for the compression ratio of any ILFSC and any string x.

Theorem 2. *For any $x \in A^n$, we have*

$$\rho_{E(s)}(x) \geq \frac{c(x) + s^2}{n \log \alpha} \log \frac{c(x) + s^2}{4s^2} + \frac{2s^2}{n \log \alpha}.$$

Proof. Let

$$x = x[1 \dots n_1]x[n_1 + 1 \dots n_2] \dots x[n_c - 1 \dots n_c]$$

be a parsing of x into the maximum number of unique phrases possible, *i.e.* $c(x)$ phrases.

Suppose we denote the corresponding output phrases by $y[n_{j-1} + 1 \dots n_j]$, for $1 \leq j \leq c$. The output phrases need not be unique, since uniqueness is assured only given the triple

$$(\text{start state}, \text{output phrase}, \text{final state}).$$

We group together the input phrases depending on the length of its output phrase. Let c_j be the number of distinct input phrases which have exactly j-long output phrases. We know $c_j \leq s^2 2^j$, since the output alphabet is binary, and there are s^2 possible pairs of states to form the triples.

To obtain a lower bound on the output length, $L(y[1 \ldots n])$. To overestimate $L(y)$, it is sufficient to overestimate c_j, $j = 0, 1, 2, \ldots$ at the expense of $\sum_{i>j} c_i$, provided the sum of all c_j remains c.

Now,

$$q = \sum_{j=0}^{k} 2^j \ + \ \Delta_k.$$

We can assume that

$$c_j = 2^2 2^j \text{ for } 0 \leq j \leq k,$$

and

$$c_{k+1} = s^2 \Delta_k + r.$$

Hence

$$c = s^2 \left(\sum_{j=0}^{k} 2^j + \Delta_k \right) \ = \ s^2 (2^{k+1} + t), \tag{16}$$

where

$$t = \Delta_k - 1 + \frac{r}{s^2}.$$

Now,

$$\begin{aligned} L(y) &\geq s^2 \sum_{j=0}^{k} j 2^j \ + \ (k+1)(s^2 \Delta_k + r) \\ &= s^2 [(k-1)2^{k+1} + 2] \ + \ (k+1)(s^2 \Delta_k + r) \\ &= s^2 (k-1)(2^{k+1} + t) + s^2 (k + 3 + 2t) \\ &= (k-1)(c + s^2) + 2s^2 (t+2). \end{aligned}$$

From (16), we have

$$(k-1) \ = \ \log \frac{c - s^2 t}{s^2} - 2 \ = \ \log \frac{c + s^2}{4s^2} - \log \left[1 + \frac{(t+1)s^2}{c - s^2 t} \right].$$

Now, substituting this back into the estimate of $L(y)$, we get

$$L(y) \ \geq \ (c + s^2) \left(\log \frac{c + s^2}{4s^2} + \tau \right) \tag{17}$$

where

$$\tau = \frac{2s^2}{c + s^2} + 2 \frac{\phi}{1 + \phi} - \log(1 + \phi), \tag{18}$$

with

$$\phi = \left(\Delta_k + \frac{r}{s^2} \right) \frac{1}{2^{k+1}}. \tag{19}$$

This establishses that

$$L(y) \geq (c + s^2) \log \frac{c + s^2}{4s^2} + 2s^2.$$

□