Lossless Source Coding

Lempel-Ziv Coding

Lempel-Ziv 1978 (LZ78)

LZ78

[LZ78] J. Ziv and A. Lempel, “Compression of individual sequences via variable rate coding,” *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 530–536, Sept. 1978

- The analysis here follows [Cover & Thomas ’91, ‘06] , attributed to [Wyner & Ziv]. It differs from the original proof in [LZ78]
- The main mechanism in LZ schemes is *pattern matching*: find string patterns that have occurred in the past, and compress them by encoding a reference to the previous occurrence

... r e s t o r e o n e s t o n e ...

- In LZ77, back references are allowed to *any* location in the dictionary window
- In LZ78, back references are allowed only to *certain points* in the past window, determined by *incremental parsing*

LZ78: Incremental Parsing

- The scheme is based on the notion of *incremental parsing*
- Parse the input sequence x_1^n into *phrases*, each new phrase being *the shortest substring that has not appeared so far as a phrase*
- Phrases are collected into an indexed *dictionary*, initialized to $\{0{:}\,\lambda\}$

x_1^n = 1,0,1 1,0 1,0 1 0,0 0,1 0, ... (assume $A = \{0,1\}$)

index: phrase	index: phrase
$0{:}\ \lambda$	$5{:}\ 010$
$1{:}\ 1$	$6{:}\ 00$
$2{:}\ 0$	$7{:}\ 10$
$3{:}\ 11$	$\vdots : \ \vdots$
$4{:}\ 01$	

- Each new phrase is of the form wb, w $=$ a previous phrase, $b \in \{0,1\}$
  - a new phrase can also be described as (i, b) , where $i = \mathrm{index}(w)$

Incremental Parsing (cont.)

$$x_1^n = 1,0,1\ 1,0\ 1,0\ 1\ 0,0\ 0,1\ 0,\ \ldots$$

- New phrase wb described as (i, b) , where $i = \text{index}(w)$, $b \in \{0, 1\}$

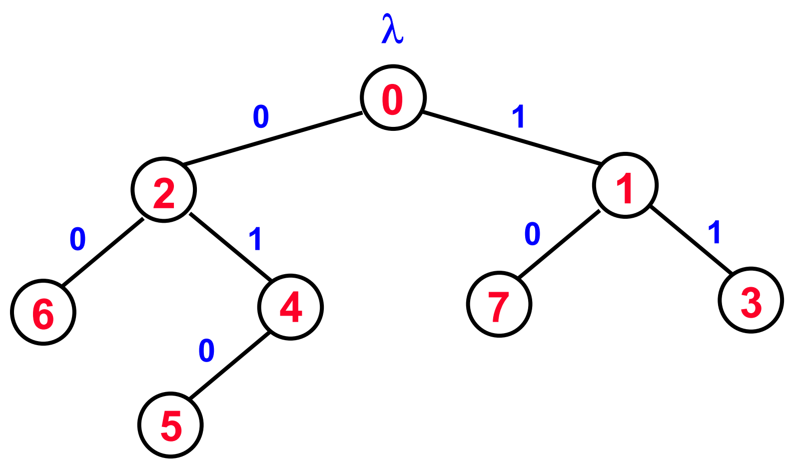
index:	phrase	descr
0:	λ	
1:	1	$(0, 1)$
2:	0	$(0, 0)$
3:	11	$(1, 1)$
4:	01	$(2, 1)$

index:	phrase	descr
5:	010	$(4, 0)$
6:	00	$(2, 0)$
7:	10	$(1, 0)$
$\vdots$:	$\vdots$	$\vdots$

- Let $c(n) =$ number of phrases in x_1^n
  - we assume, for simplicity, that the input ends at a phrase boundary (comma)
    - assumption is easily removed; it does not affect the main results
  - *we encode by describing the sequence of phrases*
  - a phrase description (i, b) can be encoded with $\leq 1 + \lceil \log c(n) \rceil$ bits
  - in the example, 28 bits to describe 13 : bad deal! it gets better as $n \to \infty$
  - decoding is straightforward: decoder builds *the same dictionary*, in lockstep with encoder
  - in practice, we do not need to know $c(n)$ before we start encoding
    - use increasing length codes that the decoder can keep track of

The Parsing Tree

$x_1^n = $ 1,0,1 1,0 1,0 1 0,0 0,1 0, ...



index	*phrase*
0	λ
1	0,1
2	0,0
3	1,1
4	2,1
5	4,0
6	2,0
7	1,0
⋮	⋮

dictionary

- coding could be made more efficient by "recycling" codes of nodes that have a complete set of children (e.g., 1, 2 above)
- will not affect asymptotics
- many (many many) tricks and hacks exist in practical implementations

Incremental Parsing: How Many Phrases?

Lemma: $c(n) \leq \frac{n}{(1-\epsilon_n)\log n}$, $\epsilon_n \to 0$ as $n \to \infty$ $\quad [\epsilon_n = o(1)]$

Proof: $c(n)$ is max when we take all phrases as short as possible.

Taking the 2 phrases of length 1, 4 of length 2, ... , up to 2^k of length k, we get overall length $n_k = \sum_{j=1}^{k} j 2^j = (k-1)2^{k+1} + 2$,

and a number of phrases $c_k = \sum_{j=1}^{k} 2^j = 2^{k+1} - 2$.

We have $c_k \leq \frac{n_k}{k-1}$.

Choose k such that $n_k \leq n < n_{k+1}$ and write $n = n_k + \Delta$. Add $\left\lfloor \frac{\Delta}{k+1} \right\rfloor$ phrases of length $k+1$, and possibly one shorter (repeated) tail phrase, to reach n. Then,

$$c(n) \leq c_k + \frac{\Delta}{k+1} + 1 \leq \frac{n_k}{k-1} + \frac{\Delta}{k+1} + 1 \leq \frac{n_k + \Delta}{k-1} + 1 = \frac{n + k - 1}{k-1} \leq \frac{n}{(1-\epsilon_n)\log n}$$

with $\epsilon_n = O(\log\log n / \log n)$.

[why: if $x = k\, 2^k$ then $\log x - \log\log x \leq k \leq \log x - \log\log x + o(1)$] ■

Corollary: $\frac{c(n)}{n} \leq \frac{1+o(1)}{\log n}$

Universality of LZ78

- To establish the universality of LZ78, we will make a connection between its combinatorial structure (the incremental parsing), and probabilistic notions
- A k-*th order Markov probability assignment* Q_k on A^n is defined by
  - a distribution $Q_{init}(s_1)$ on an *initial state* $s_1 = x^0_{-(k-1)}$ (could be fixed, i.e., a single-mass PMF)
  - a collection of conditional probability distributions

$$Q(\cdot \mid a_1^k) \quad \text{for all} \quad a_1^k \in A^k$$

- The probability assigned by Q_k to $x_1^n \in A^n$ (with initial state s_1) is

$$Q_k(x_0, x_1, \ldots, x_n | s_1) \triangleq Q_{init}(s_1) \prod_{j=1}^{n} Q\left(x_j \mid x_{j-k}^{j-1}\right)$$

we will assume, for simplicity, an arbitrary fixed initial state s_1, so $Q_{init}(s_1) = 1$.

Universality of LZ78 (cont.)

- Assume x_1^n is parsed into c *distinct* phrases $y_1, y_2, \dots, y_c$. Define:
  - v_i = index of start of $y_i = (x_{v_i}, \dots, x_{v_{i+1}-1})$
  - $s_i = (x_{v_i-k}, \dots, x_{v_i-1})$ = the k bits preceding y_i in x_1^n (a *state*). We have

$$Q_k(y_i|s_i) = \prod_{j=0}^{v_{i+1}-v_i-1} Q\left(x_{v_i+j} \,\middle|\, x_{v_i+j-k}^{v_i+j-1}\right)$$

$$Q_k(x_1^n|s_1) = \prod_{i=1}^{n} Q\left(x_i|x_{i-1}^{i-k}\right) = \prod_{i=1}^{c} Q_k(y_i|s_i)$$

  - c_{ls} = number of phrases y_i of length l and preceding state $s \in \{0,1\}^k$
  - we have $\sum_{l,s} c_{ls} = c$ and $\sum_{l,s} l\, c_{ls} = n$

Ziv's inequality: For any distinct parsing of x_1^n, and any Q_k, we have

$$\log Q_k(x_1, x_2, \dots, x_n|s_1) \le -\sum_{l,s} c_{ls} \log c_{ls}$$

The lemma upper-bounds the probability of *any sequence* under *any probability assignment* from the class, based on properties of *any distinct parsing* of the sequence (including the incremental parsing)

Universality of LZ78 (proof of Ziv's inequality)

Proof of Ziv's inequality:

$$Q_k(x_1, x_2, \dots, x_n | s_1) = Q_k(y_1, y_2, \dots, y_c | s_1) = \prod_{i=1}^{c} Q(y_i | s_i)$$

$$\log Q_k(x_1, x_2, \dots, x_n | s_1) = \sum_{i=1}^{c} \log Q(y_i | s_i)$$

$$= \sum_{l,s} \sum_{i:|y_i|=l, s_i=s} \log Q(y_i|s)$$

$$= \sum_{l,s} c_{ls} \underbrace{\sum_{i:|y_i|=l, s_i=s} \frac{1}{c_{ls}} \log Q(y_i|s)}_{E \log Q(y_i|s)}$$

$E \log X \le \log EX$

Here X is uniformly distributed on the c_{ls} numbers $Q(y_i|s)$

Jensen

$$\le \sum_{l,s} c_{ls} \log \underbrace{\left(\sum_{i:|y_i|=l, s_i=s} \frac{1}{c_{ls}} Q(y_i|s) \right)}_{\log EQ(y_i|s)}$$

Since the y_i are distinct, we have $\sum_{i:|y_i|=l, s_i=s} Q(y_i|s) \le 1$

$$\Longrightarrow \log Q_k(x_1, x_2, \dots, x_n | s_1) \le \sum_{l,s} c_{ls} \log \frac{1}{c_{ls}} = -\sum_{l,s} c_{ls} \log c_{ls} \quad \blacksquare$$