LZ78: Summary so far

- *Incremental parsing*: Input sequence x_1^n parsed into $c(n)$ *distinct phrases* (except maybe the very last one, which is immaterial to the main asymptotic results).
- Phrases are collected into a *dictionary*, which are conveniently represented by a *tree*.
- Upper bound on number of phrases:

$$\frac{c(n)}{n} = \frac{1+o(1)}{\log n}$$

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

- Each phrase can be encoded with $\lceil \log c(n) \rceil + 1$ bits, yielding a total code length

$$L(x_1^n) = c(n)(\lceil \log c(n) \rceil + 1)$$

- Ziv's inequality connects the incremental parsing with k-*th order Markov probability assignments*

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

written as code lengths

where c_{ls} = number of phrases of length l that occur following a given k-tuple s in x_1^n.

Universality for Individual Sequences: Theorem

Theorem: *For any sequence* x_1^n *and for any* k*-th order probability assignment* Q_k, *we have*

$$\frac{c(n)\log c(n)}{n} \le -\frac{1}{n}\log Q_k(x_1^n|s_1) + \frac{\big(1+o(1)\big)k}{\log n} + O\left(\frac{\log\log n}{\log n}\right)$$

Auxiliary lemma (maximal entropy):

Let X *be a random variable over* $\mathbb{Z}_{\geq 0}$ *with PMF* p *such that* $E_p X = \mu$. Then $H(X)$ *is maximized when* $p(x) = \exp(\lambda_0 + \lambda_1 x)$ *satisfying the constraint*.

- Proof: Consider a PMF q satisfying the constraint. Then show $H(p) - H(q) = D(q||p)$ using $E_p X = E_q X = \mu$ and $\sum_x p(x) = \sum_x q(x) = 1$. ■

Corollary: *For* X *as above,*

$$H(X) \le (\mu+1)\log(\mu+1) - \mu\log\mu$$

- Proof: Solve for λ_1 and λ_0 in terms of μ, and write $H_p(X)$ explicitly. ■

Universality for Individual Sequences: Proof

Define $\pi_{ls} \triangleq \frac{c_{ls}}{c}$. Then, $\sum_{l,s} \pi_{ls} = 1$ and $\sum_{l,s} l\, \pi_{ls} = \frac{n}{c}$

(recall $\sum_{l,s} c_{ls} = c$ and $\sum_{l,s} l\, c_{ls} = n$).

Define r.v. $U, V \sim P(U = l, V = s) = \pi_{ls}$.

We have $EU = \frac{n}{c}$ and $H(V) \leq k$ (V defined over binary k-tuples).

From Ziv's lemma:

$$-\log Q_k(x_1^n | s_1) \geq \sum_{l,s} c_{ls} \log \frac{c_{ls} c}{c} = \sum_{l,s} c_{ls} \log c + \sum_{l,s} c_{ls} \log \frac{c_{ls}}{c}$$

$$= c \log c + c \sum_{l,s} \pi_{ls} \log \pi_{ls}$$

$$\Rightarrow \qquad -\frac{1}{n} \log Q_k(x_1^n | s_1) \geq \frac{c}{n} \log c - \frac{c}{n} H(U, V)$$

$$\geq \frac{c}{n} \log c - \frac{c}{n} \big(H(U) + H(V) \big) \qquad (\star)$$

Universality for Individual Sequences: Proof

$$-\frac{1}{n}\log Q_k(x_1^n|s_1) \geq \frac{c}{n}\log c \; - \frac{c}{n}\big(H(U)+H(V)\big) \qquad (\star)$$

By the maximum entropy theorem for mean-constrained r.v. applied to U,

recalling $EU = \frac{n}{c}$ and $\frac{c}{n} = \frac{1+o(1)}{\log n}$

$$H(U) \leq \left(\frac{n}{c}+1\right)\log\left(\frac{n}{c}+1\right) - \frac{n}{c}\log\frac{n}{c}$$

$$\Rightarrow \quad \frac{c}{n}H(U) \leq \left(1+\frac{c}{n}\right)\log\left(\frac{n}{c}+1\right) - \log\frac{n}{c}$$

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

$$= \underbrace{\frac{c}{n}\log\frac{n}{c}}_{O\left(\frac{\log\log n}{\log n}\right)} + \underbrace{\left[\log\left(\frac{n}{c}+1\right) - \log\frac{n}{c}\right]}_{O(c/n) = O(1/\log n)}\left(\frac{c}{n}+1\right) = O\left(\frac{\log\log n}{\log n}\right)$$

with $x \to \infty$, we have

$$\log(x+1) - \log x$$

$$= \log\left(1+\frac{1}{x}\right) = O\left(\frac{1}{x}\right)$$

Together with $(\star)$ and $H(V) \leq k$,

$$-\frac{1}{n}\log Q_k(x_1^n|s_1) \geq \frac{c\log c}{n} - \frac{\big(1+o(1)\big)k}{\log n} - O\left(\frac{\log\log n}{\log n}\right)$$

■

Universality for Individual Sequences: Discussion

- The theorem holds for *any k-th order probability assignment Q_k*, and, in particular, the k-th order empirical distribution of x_1^n, which gives an ideal code length equal to the empirical entropy

$$-\frac{1}{n}\log \hat{P}_k(x_1^n) = \widehat{H}_k(x_1^n)$$

- The asymptotic $O\left(\frac{\log\log n}{\log n}\right)$ term in the redundancy has been improved to $O\left(\frac{1}{\log n}\right)$ — this is the best possible upper bound

- Universal schemes based on context modeling and arithmetic coding can achieve a faster convergence rate: $O\left(\frac{\log n}{n}\right)$ in the class of finite memory Markov sources.

Compressibility

- *Finite-memory compressibility*

Q_k is optimized for x_1^n, for each k

we must have $n \to \infty$ before $k \to \infty$, otherwise definitions are meaningless!

$$FM_k(x_1^n) = \inf_{Q_k, s_1} \left(-\frac{1}{n} \log Q_k(x_1^n | s_1) \right)$$ k-th order, finite sequence

$$FM_k(x_1^\infty) = \limsup_{n\to\infty} FM_k(x_1^n)$$ k-th order, infinite sequence

$$FM(x_1^\infty) = \lim_{k\to\infty} FM_k(x_1^n)$$ *FM compressibility*

- *Lempel-Ziv compression ratio*

$$LZ(x_1^n) = \frac{1}{n} c(n) \left(\lceil \log c(n) \rceil + 1 \right)$$ finite sequence

$$LZ(x_1^\infty) = \limsup_{n\to\infty} LZ(x_1^n)$$ *LZ compression ratio*

Theorem: For any sequence x_1^∞, $LZ(x_1^\infty) \le FM(x_1^\infty)$

Probabilistic Setting

Theorem: Let $X_{-\infty}^{\infty}$ be a stationary ergodic random process. Then,

$$LZ(X_1^{\infty}) \leq H(X_1^{\infty}) \text{ with probability } 1$$

Proof: via approximation of the stationary ergodic process with Markov processes of increasing order, and the previous theorems.

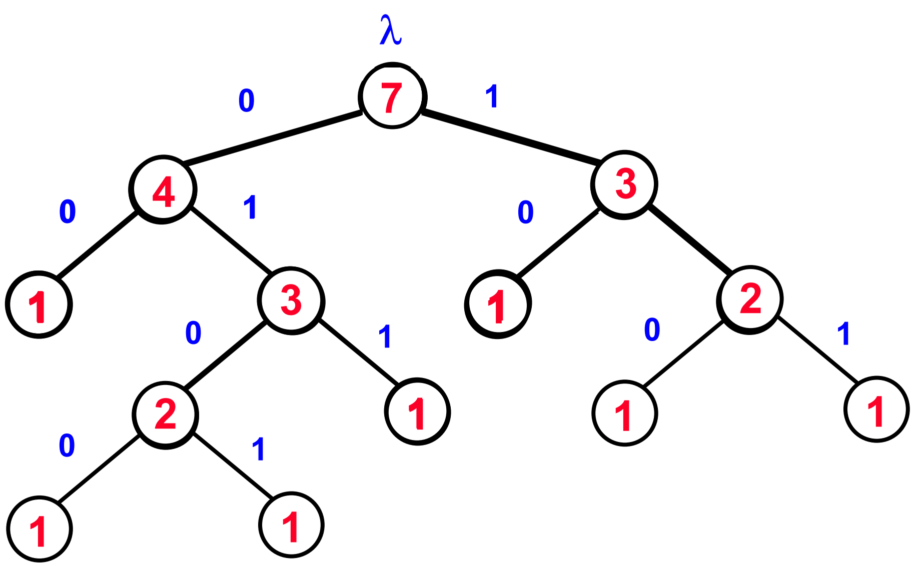
$$Q_k\left(x_{-(k-1)}^{0} x_1^{n}\right) \triangleq P_X\left(x_{-(k-1)}^{0}\right) \prod_{j=1}^{n} P_X\left(x_j \middle| x_{j-k}^{j-1}\right), \qquad X \sim P_X$$

$$H\left(X_j \middle| X_{j-k}^{j-1}\right) \xrightarrow{k \to \infty} H(X)$$

Markov k-th order approximation of X

The LZ Probability Assignment

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



In general,

$$P(x_1^n) = \frac{1}{(c(n)+1)!}$$

$$-\log P = c(n)\log c(n) + o(c(n)\log c(n))$$

- Slightly different tree evolution
  anticipatory parsing: when a new phrase is parsed, add both children to the tree (keps it *complete*)
- A *weight* is kept at every node
  - number of times the node was traversed through + 1
- A node act as a conditioning state, assigning to its children probabilities proportional to their weight
- Example: string <u>101101010</u> 011
  s

  $P(0|s)$ = 4/7

  $P(1|s0)$ = 3/4

  $P(1|s01)$ = 1/3

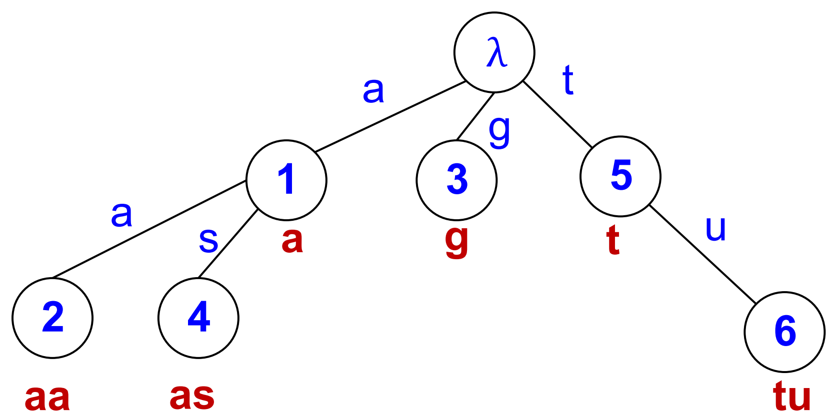
  $P(011|s)$ = (4/7)*(3/4)*(1/3) = 1/7

  Notice `telescoping'
- Similarly, P(010|101101) = 1/6, etc.
- $\Rightarrow$ $P(s011)$ = 1/(7!)

every lossless compression algorithm defines a prob. assignment, even if it wasn't meant to!

Other Properties

- Individual sequences result applies also to FSM probability assignments
- The "worst sequence"
  - *counting sequence* 0 1 00 01 10 11 000 001 010 011 100 101 110 111 ..
  - maximizes $c(n)$, incompressible with LZ78
- Generalization to larger alphabets is straightforward



  - Data structure must be efficient to accommodate possibly small subsets of the alphabet occurring at each node

Other Properties

- *LZW modification*: extension symbol b not sent. It is determined by the first symbol of the next phrase instead [Welch 1984]
  - dictionary is initialized with all single-symbol strings
  - works very well in practice
  - breakthrough in popularization of LZ, led to UNIX *compress*
- In real life we use *bounded dictionaries*, and need to reset them from time to time
  - E.g.: a dictionary for 2^{16} entries. Once all the entries are used, we may
    - freeze the dictionary and continue with it until the input is exhausted
    - erase the dictionary and start from scratch (full reset)
    - erase part of the dictionary and fill with new entries
    - delay the reset until compression ratio deteriorates
    - …

Lempel-Ziv in the Real World

- The most popular data compression algorithm in use
  - virtually every computer in the world runs some variant of LZ
  - LZ78
    - compress
    - GIF
    - TIFF
  - LZ77
    - gzip, pkzip (LZ77 + Huffman for pointers and symbols)
    - png
    - 7-zip
  - many more implementations in software and hardware
    - most modern operating systems include compression libraries with LZ
    - software distribution
    - tape drives
    - printers
    - network routers
    - various commercially available VLSI designs
    - ...

Some comparisons

- Input file: Don Quijote de la Mancha, HTML
  file size: 2,261,865 bytes

Compressor	Output bytes	bits/symbol
Huffman	1,284,323	4.54
vanilla LZ77	1,310,561	4.63
gzip -1	994,295	3.52
gzip -9	816,909	2.89

Some comparisons

- Input file: Don Quijote de la Mancha, HTML
  file size: 2,261,865 bytes

Compressor	Output bytes	bits/symbol
Huffman	1,284,323	4.54
vanilla LZ77	1,310,561	4.63
gzip -1	994,295	3.52
gzip -9	816,909	2.89
LZ78 (LZW with 16 bit dict)	839,560	2.97

Some comparisons

- Input file: Don Quijote de la Mancha, HTML
  file size: 2,261,865 bytes

Compressor	Output bytes	bits/symbol
Huffman	1,284,323	4.54
vanilla LZ77	1,310,561	4.63
gzip -1	994,295	3.52
gzip -9	816,909	2.89
LZ78 (LZW with 16 bit dict)	839,560	2.97
LZMA (7z)	639,295	2.26

Universality is great, but …

- Input file: Mars rock image
  file size: 693,904 bytes

Compressor	Output bytes	bits/symbol
Uncompressed	693,904	8.00
gzip-9	627,858	7.23
LZW	668,327	7.70
7-Zip	524,622	6.05
JPEG-LS	465,353	5.36

Universality is great, but …

- Input file: Tools image
  file size: 1,828,817 bytes

Compressor	Output bytes	bits/symbol
Uncompressed	1,828,817	8.00
gzip-9	1,639,673	7.17
LZW	1,775,923	7.77
7-Zip	1,367,617	5.98
JPEG-LS	1,235,563	5.40