Arithmetic coding [1960's+ – Witten, Neal, Cleary – CACM 1987]

6.897
L 13.2
Apr. 9, 2003

- clever idea to allow sharing of fractional bits
- code string as real number in $[0,1]$:

  - which piece in partition $1 = p_1 + p_2 + \cdots + p_{|\Sigma|}$ the code is in decides first letter
  - recurse.
- $\text{code}(k\,x) = \sum_{i=1}^{k-1} p_i + p_k \cdot \text{code}(x)$ (k: letter, x: string)
- if the string is infinite, we get a real number
- else we really get a range $(L, R]$
  - $\text{code}(k) = \sum_{i=1}^{k-1} p_i + (0, p_k]$.
  - if we write L, R in binary & take $\text{lcp}(L, R)$ then the number $\text{lcp}(L,R)\,1 \in$ range $(L, R]$

Theorem: $E[\text{code length for } n\text{-char. string over } H] \in [H \cdot n,\ H \cdot n + 1)$

additive error now over whole string not per letter

Proof: Suppose string $x = x_1 x_2 \cdots x_n$.
In fact, code length is exactly $\lceil \lg 1/p_{x_1} + \lg 1/p_{x_2} + \cdots + \lg 1/p_{x_n} \rceil$

$= \lceil \lg 1/\underbrace{p_{x_1} p_{x_2} \cdots p_{x_n}}_{\text{length of final interval}} \rceil$

$= \lceil \lg 1/\underbrace{\lfloor L p_{x_1} \cdot p_{x_2} \cdot \cdots \cdot p_{x_n} \rfloor}_{\text{dividing into this power of 2 leaves a notch in the interval}} \rceil$.

$\leq 1 + \lg 1/p_{x_1} + \lg 1/p_{x_2} + \cdots + \lg 1/p_{x_n}$

$\Rightarrow E[\text{code length}] < 1 + H \cdot n$. □

- ∃ variations to speed up computation & use less precision etc.