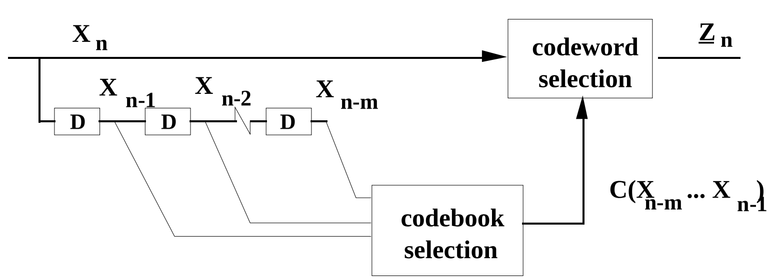
Arithmetic Coding

- A lossless coding technique that encodes data at rates close to entropy or entropy-rate, without the +1/k penalty or the high complexity of block to variable-length coding.
- It can be used with probability estimation methods enable it to adapt to unknown or changing source probability distributions. Thus, it can be 'universal'.
- Arithmetic coding uses a coding strategy somewhat like conditional coding, so we begin by reviewing conditional coding.

Conditional Coding



- For every m-tuple $X_1 \ldots X_m$, there is a prefix code $C(X_1,\ldots,X_m)$ optimized for conditional probabilities $\{ P(x|X_1 \ldots X_m) : x \in A \}$.
- Overall rate

  $$H(X_{m+1}|X_1 \ldots X_m) \le R^* \le H(X_{m+1}|X_1 \ldots X_m) + 1$$

  where conditional is entropy is

  $$H(X_{m+1}|X_1 \ldots X_m) = - \sum_{x_1 \ldots x_{m+1}} P(x_1 \ldots x_{m+1}) \log P(x_{m+1}|x_1 \ldots x_m)$$

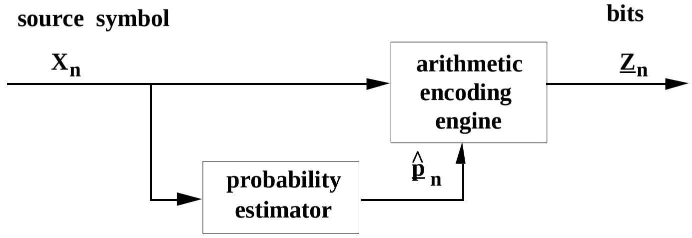
- For stationary sources

  $H(X_{m+1}|X_1 \ldots X_m) \downarrow H_\infty(X)$ (faster than $H_m(X) \downarrow H_\infty(X)$)

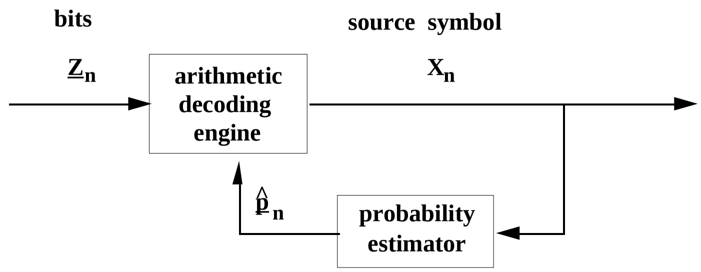
- Complexity is of the same order as block-to-variable length codes.

Arithmetic Coding

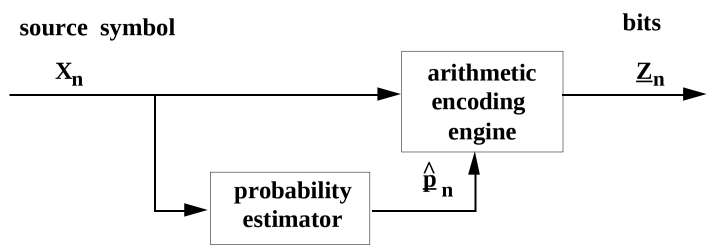
- Encoder:



- $\hat{\underline{p}}_n = \{ \hat{p}_n(x) : x \in A \}$, $\hat{p}_n(x)$ = estimate of $P(X_n = x)$ derived from previous X's
- $\underline{Z}_n$ = bits emitted in response to X_n, i.e. after X_n arrives and before X_{n-1}
- $X_1 \ldots X_n \rightarrow Z_1 \ldots Z_{L_n}$ where $L_n \cong -\log \prod_{i=1}^{n} \hat{p}_n(X_i)$. This is the key property.
- Decoder:



Arithmetic Coding



Key example: Suppose

$\hat{p}_n(x)$ = estimate of $P(X_n = x \mid X_{n-m} \ldots X_{n-1})$ based on $X_1, \ldots, X_{n-1}$

If source is stationary & ergodic and probability estimates are perfect

$$\frac{L_n}{n} \cong -\frac{1}{n} \log \prod_{i=1}^{n} \hat{p}_n(X_i) \cong -\frac{1}{n} \log \prod_{i=1}^{n} \Pr(X_i|X_{i-m} \ldots X_{i-1})$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \log \Pr(X_i|X_{i-m} \ldots X_{i-1})$$

$\cong E - \log \Pr(X_i|X_{i-m} \ldots X_{i-1})$ is n is large, by law of large numbers

$= H(X_{m+1}| X_1 \ldots X_m)$ with high probability, by ergodic thm.

Probability Estimation

- Basic idea

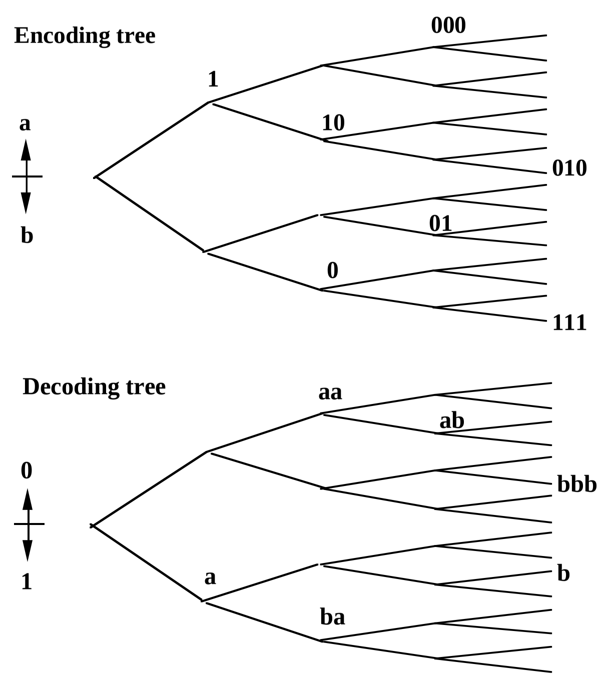
  For each $x_1,..,x_m$ build table of conditional frequency of each x_{m+1}.

- Can reduce size of tables

  If the conditional frequency tables for $x_1,x_2,...,x_m$ don't depend significantly on x_1, keep a table just for $x_2,...,x_m$.

Arithmetic Coding Engine

- Restrict attention to binary sources.
- Irregular nature of encoding and decoding is illustrated by the following encoding and decoding trees. They are not actuallly used in encoding and decoding.



Four Levels of Explanation

- Infinite sequence to infinite sequence

  $\underline{X} = X_1 X_2 \ldots \rightarrow \underline{Z} = Z_1 Z_2 \ldots$

- Finite sequence to finite sequence

  $X^n = X_1 \ldots X_n \rightarrow Z^n = Z_1 \ldots Z_{L_n}$

- Incremental encoding and decoding

  after $X^{n-1} = X_1 \ldots X_{n-1} \rightarrow Z^{L_n} = Z_1 \ldots Z_{L_{n-1}}$, $X_n \rightarrow Z_{L_{n-1}} \ldots Z_{L_n}$

- Incremental encoding with finite precision arithmetic

  we'll skip this

- For purposes of discussion and illustration:

  Assume binary, stationary, memoryless source with known probability distribution. (No prob. estimation.)

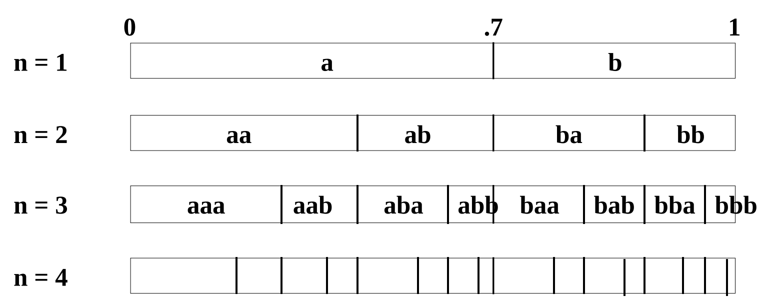
  $P(X = a) = p$, $P(X = b) = 1-p$ ($p \cong .6$ in pictures)

  $P(X_1 \ldots X_n = a\ a\ b\ a\ b \ldots) = p^{N_a} (1-p)^{N_b}$

AC-7

Interval Partitioning

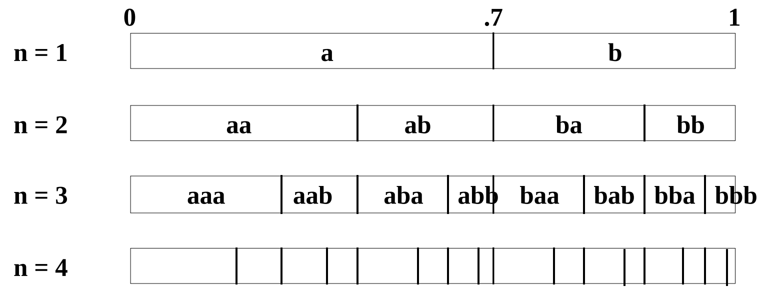
- For each n , hierarchically partition the unit interval $[0,1)$ according to probability of n-tuples. The interval corresponding to the sequence $(x_1,\ldots,x_{n-1})$, is divided into two intervals, one for the $(x_1,\ldots,x_{n-1},a)$ whose length fraction p of the lenth of the interval for $(x_1,\ldots,x_{n-1})$ and one for $(x_1,\ldots,x_{n-1},b)$ whose length is fraction $1-p$ of the length of the ineral for $(x_1,\ldots,x_{n-1})$. The first interval is to the left of the second.



- Notation: The encoding interval for X^n

  $J(X^n) = [A(X^n), B(X^n)) = [A_n, B_n)$ = interval assoc. with $X^n = (X_1 \ldots X_n)$

AC-8



- $J(X^n) = [\, A(X^n), B(X^n) \,) = [\, A_n, B_n \,)$ interval associated with $X^n = (X_1 \ldots X_n)$
- Key facts :

  (1) length of $J(X^n) = \prod_{i=1}^{n} P_i(X_i) = P(X^n)$

  The last equality derives from the IID assumption of our example. We write $P_i(X_i)$ because in real situations, the probability distribution of X_i often changes with i.

  (2) $J(X^n) \subset J(X^{n-1})$, $\quad J(X^n) = J(X^{n-1}a) \cup J(X^{n-1}b)$

  (3) $$J(X^n) = \begin{cases} [A_{n-1}, A_{n-1}+p(B_{n-1}-A_{n-1})), & \text{if } X_n=a \\ [A_{n-1}+p(B_{n-1}-A_{n-1}), B_{n-1}), & \text{if } X_n=b \end{cases}$$

Infinite Sequence to Infinite Sequence Encoding

- Notice that as n increases $J(X^n)$ narrows to a number denoted Z
- Encode $\underline{X} = X_1 X_2 \ldots$ into $\underline{Z} = Z_1 Z_2 \ldots$,

  where $Z_1 Z_2 \ldots$ is the binary expansion of the unique number Z in

  i.e., if $Z = .Z_1 Z_2 Z_3 \ldots$, send $\underline{Z} = Z_1, Z_2, \ldots$
- Notes:
  - $Z = \lim_{n\to\infty} A_n = \lim_{n\to\infty} B_n$
  - One cannot deduce the rate of the code from just the infinite sequence to infinite sequence encoding rule.
  - $Z \in J(X^n)$ all n
  - If $Z \in J(u_1,\ldots,u_n)$ for some $u_1,\ldots,u_n$, then $x_1 \ldots x_n = u_1 \ldots u_n$
  - Thefore, $Z \in J(u_1,\ldots,u_n)$ iff $u_1,\ldots,u_n$, then $x_1 \ldots x_n = u_1 \ldots u_n$
  - The above two facts are useful when decoding.

Finite Sequence to Finite Sequence Encoding

- Key observation: $Z_1 = 0$ iff $Z \le .5$, $Z_1 = 1$ iff $Z > .5$
- How much of $X_1, X_2, \ldots$ do we have to see before we can tell if $Z \le$ or $> .5$?
- That is, what is the smallest value of n such that $X_1,\ldots,X_n$ determine Z_1?
- One cannot say in advance. That is, the value of n depends on $X_1, X_2, \ldots$.
- However, as n increases, we will at some point find the smallest n such that either

  $B_n \le .5$, in which case $Z_1 = 0$,

  or

  $A_n \ge .5$, in which case, $Z_1 = 1$.

  This is the n after which the encoder produces Z_1. That is, this is the n for which $X_1,\ldots,X_n$ determines Z_1.
- Similarly, Z_2 will be produced by the encoder at the first time n such that either

  $[A_n,B_n)\subset[0,.25]$ or $[A_n,B_n)\subset[.25,.5]$ or $[A_n,B_n)\subset[.5,.75]$ or $[A_n,B_n)\subset[.75,1]$

  in which case

  Z_2 = 0 or 1 or 0 or 1
- The general case is discussed on the next page

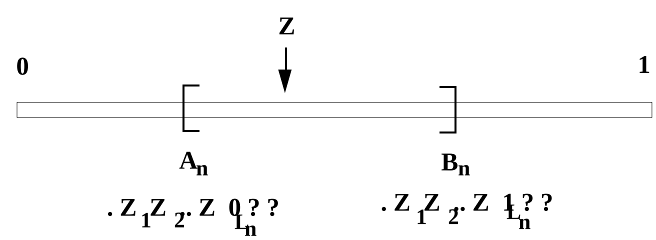
AC-11

Finite Sequence to Finite Sequence Encoding

To encode $X_1 X_2 \ldots X_n$ into $Z_1 Z_2 \ldots Z_{L_n}$ and to find L_n:

1. Find $J(X^n) = [L_n,R_n)$
2. Let $Z_1 Z_2 \ldots Z_{L_n}$ = Greatest Common Prefix (GCP) of binary expansions of A_n & B_n

$A_n = .\, Z_1 Z_2 Z_3 \ldots Z_{L_n}\, 0\, a_{L_n+2} \ldots$ $\qquad$ $B_n = .\, Z_1 Z_2 Z_3 \ldots Z_{L_n}\, 1\, b_{L_n+2} \ldots$



Encoding length and rate

$P(X^n)$ = $B_n - A_n$ = . 0 0 ... 0 0 1 ? ? ? (L_n 0's)

$\le$. 0 0 ... 0 1 0 0 ... (L_n-1 0's)

$= 2^{-L_n}$

$$\Rightarrow \quad \frac{L_n}{n} \le -\frac{1}{n}\log P(X^n). \qquad E\,\frac{L_n}{n} \le H_n(X) = H_\infty(X)$$

AC-12

Finite Sequence to Finite Sequence Decoding

Example: Suppose $Z_1Z_2Z_3 = 0\ 0\ 1$

Then we know $Z = .0\ 0\ 1\ ?\ ?\ ?\ ?$

So we know $.125 \le Z \le .25$

Therefore,

$Z \in [0,.6] \Rightarrow X_1 = a$

$Z \in [0,.36] \Rightarrow X_1X_2 = a\ a$

However, we can't tell whether

$Z \in [0,.215)$ or $[.215,.35)$.

Therefore, from $Z_1 Z_2 Z_3 = 0\ 0\ 1$, we cannot determine X_3, until more Z_i's are received.

The general case is explained on the next page.

Finite Sequence to Finite Sequence Decoding

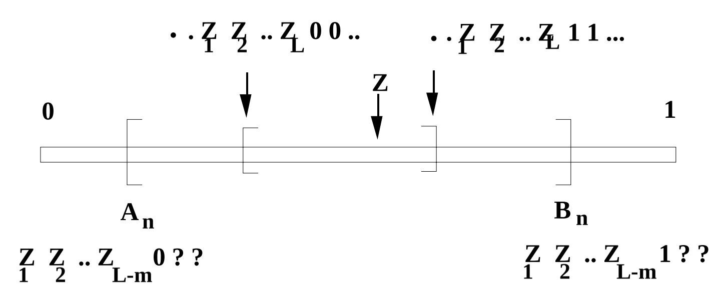
Given $Z_1 Z_2 \ldots Z_L$

1. Let $K(Z^L) = [.Z_1 Z_2 \ldots Z_L\ 0\ 0 \ldots, \ .Z_1 Z_2 \ldots Z_L\ 1\ 1 \ldots)$ = decoding interval for Z^L.
2. Cleary, $Z \in K(Z^L)$
3. Find n (as large as possible) and $X_1 \ldots X_n$ such that encoding interval for $X_1,\ldots,X_n$ contains the decoding interval K_{Z^L}, i.e.

   $J(X^n) \supset K(Z^L)$
4. Decode $Z_1 Z_2 \ldots Z_L$ into $X_1 \ldots X_n$

Key fact:

$Z \in J(Y^n)$ implies $X^n = Y^n$



Incremental Encoding

- Assume we have already encoded X^{n-1} into $Z^{L_{n-1}}$ and $J(X^{n-1}) = [A_{n-1}, B_{n-1})$.
- Binary expansions of A_{n-1} and B_{n-1} need only have been computed out to where they differ. That is, we need only have computed

  $\tilde{A}_{n-1} = .Z_1,\ldots,Z_{L_{n-1}},0,0,\ldots$ and $\tilde{B}_{n-1} = .Z_1,\ldots,Z_{L_{n-1}},1,0,0,\ldots$

  ($\tilde{A}_{n-1}$ and $\tilde{B}_{n-1}$ are called "dyadic" since they have finite binary expansions)
- Given next source symbol X_n
  1. Compute binary expansions of A_n and B_n out to where they disagree. These begin with $Z^{L_{n-1}}$.
  2. Their GCP is $Z^{L_n} = (Z_1,\ldots,Z_{L_{n-1}}, Z_{L_{n-1}+1}, \ldots, Z_{L_n})$
  3. Output $Z_{L_{n-1}+1}, \ldots, Z_{L_n}$ (if not empty).
- As n increases, the arithmetic precision needs to increase.
- Incremental decoding is similar

Arithmetic Coding Credits

Elias (~1960, unpublished, see Abramson's info thy text)

invented the coding by partitioning

Rissanen 1976, Pasco 1976

developed the first incremental methods for finite precision

Further developments

Rissanen, Langdon 1979, 1981, 1981

Rubin 1979

Guazzo 1980

Jones 1981

Cleary & Witten, 1984, 1984, 1987

IBM Q-coder 1988

Comparisons

English text

- Arithmetic coding -- 2.2 bits/symbol (Cleary & Witten)
- Ziv-Lempel -- 3.5 (Pisciotta & Wei)

Ziv-Lempel is generally considered to be less complex. But it's not so clear.

Implementations

Arithmetic coding included in

- JPEG image coding standard (as an option)
- JPEG2000 image coding standard