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

• Overall rate

\[
H(X_{m+1}|X_1...X_m) \leq R^* \leq H(X_{m+1}|X_1...X_m) + 1
\]

where conditional is entropy is

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

• For stationary sources

\[
H(X_{m+1}|X_1...X_m) \downarrow H_\infty(X) \quad \text{(faster than } \downarrow H_m(X) \downarrow H_\infty(X) \text{)}
\]

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

\[ \hat{p}_n = \{ \hat{p}_n(x) : x \in A \}, \quad \hat{p}_n(x) = \text{estimate of } P(X_n = x) \text{ from previous } X\text{'s} \]

\[ Z_n = \text{bits emitted in response to } X_n \]

\[ X_1 \ldots X_n \rightarrow Z_1 \ldots Z_{L_n} \text{ where } L_n = -\log \prod_{i=1}^{n} \hat{p}_n(X_i) \]

Key example:

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

If source is stationary & ergodic and probability estimates are perfect

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

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

\[ \equiv H(X_{m+1} | X_1, \ldots, X_m) \text{ with high probability, by ergodic thm.} \text{ (law of large numbers)} \]
Probability Estimation

- Basic idea
  For each $x_1,\ldots,x_m$ build table of conditional frequency of each $x_{m+1}$.

- Reduce size of tables
  If the conditional frequency tables for $x_1,x_2,\ldots,x_m$ don't depend significantly on $x_1$, keep a table just for $x_2,\ldots,x_m$.

Arithmetic Coding Engine

- Restrict attention to binary sources.
- Irregular nature of encoding and decoding.

```
Encoding tree  
```
```
Decoding tree
```

Four Levels of Explanation

• Infinite sequence to infinite sequence
  \[ X_1 X_2 \ldots \to Z_1 Z_2 \ldots \]

• Finite sequence to finite sequence
  \[ X_1 \ldots X_n \to Z_1 \ldots Z_{L_n} \]

• Incremental encoding and decoding
  after \[ X_1 \ldots X_{n-1} \to Z_1 \ldots Z_{L_{n-1}} \], \[ X_n \to Z_{L_{n-1} \ldots L_n} \]

• Incremental encoding with finite precision arithmetic
  we’ll skip this

• For purposes of discussion:
  Assume binary, stationary, memoryless source with known probability distribution. (No prob. estimation.)
  \[ P(X = a) = p, \quad P(X = b) = 1-p \quad (p \approx .7 \text{ in pictures}) \]
  \[ P(X_1 \ldots X_n = a \ a \ b \ a \ b \ldots) = p^{N_a} (1-p)^{N_b} \]

Interval Partitioning

• For each \( n \), partition unit interval \([0,1]\) according to probability of \( n \)-tuples.

\[
\begin{array}{c|c|c}
 n = 1 & 0 & .7 & 1 \\
\hline
 a & b \\
 n = 2 & & & \\
 aa & ab & ba & bb \\
 n = 3 & & & \\
 aaa & aab & aba & ab\bar{b} & baa & bab & bba & b\bar{b} \\
 n = 4 & & & \\
 \\
\end{array}
\]

• The encoding interval for \( X^n \)
  \[ J_{X^n} = [A_n, B_n] = \text{interval assoc. with } X^n = (X_1 \ldots X_n) \]

• Key facts:
  \[ P(X^n) = \prod_{i=1}^{n} P(X_i) = \text{length of } J_{X^n} \]
  \[ J_{X^n} \subset J_{X^{n-1}}, \quad J_{X^n} = J_{X^{n-1}a} \cup J_{X^{n-1}b} \]
  \[ J_{X^n} = \begin{cases} 
 [A_{n-1}, A_{n-1}+p(B_{n-1}-A_{n-1})], & \text{if } X_n=a \\
 [A_{n-1}+p(B_{n-1}-A_{n-1}), B_{n-1}], & \text{if } X_n=b 
\end{cases} \]
Infinite Sequence to Infinite Sequence Encoding

- Encode $X_1 X_2 \ldots$ into $Z_1 Z_2 \ldots$,
  where $Z_1 Z_2 \ldots$ is the binary expansion of the unique number $Z$ in
  $J_{X_1} \cap J_{X_2} \cap J_{X_3} \ldots$
i.e., if $Z = .Z_1 Z_2 Z_3 \ldots$, send $Z_1, Z_2, \ldots$
- 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 $n$
  - $Z \in J_{u_n}$ if and only if $u_1 \ldots u_n = x_1 \ldots x_n$

Finite Sequence to Finite Sequence Encoding

- Key observation: $Z_1 = 0$ iff $Z \leq .5$, $Z_1 = 1$ iff $Z > .5$
- How much of $X_1, X_2, \ldots$ do we have to see before we can tell if $Z \leq$ 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
  
  $B_n \leq .5$, in which case $Z_1 = 0$,
or
  $A_n \geq .5$, 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
Finite Sequence to Finite Sequence Encoding

To encode \( X_1 X_2 ... X_n \) into \( Z_1 Z_2 ... Z_{L_n} \) and to find \( L_n \):

1. Find \( J_X = [A_n, B_n) \)

2. Let \( Z_1 Z_2 ... Z_{L_n} = \) Greatest Common Prefix (GCP) of binary expansions of \( A_n \) & \( B_n \)

\[
A_n = . Z_1 Z_2 Z_3 ... Z_{L_n} a_{L_n+1}...
\]

\[
B_n = . Z_1 Z_2 Z_3 ... Z_{L_n} b_{L_n+1}...
\]

Encoding length and rate

\[
P(X^n) = B_n - A_n = .00 ... 0 0 1 ? ? ? \quad (L_n \text{ 0's})
\]

\[
\leq .00 ... 0 1 0 0 ... \quad (L_n-1 \text{ 0's})
\]

\[
= 2^{-L_n}
\]

\[
\Rightarrow \quad \frac{L_n}{n} \leq \frac{1}{n} \log P(X^n).
\]

\[
E \frac{L_n}{n} \leq H_n(X) = H_\infty(X)
\]

AC-11

Finite Sequence to Finite Sequence Decoding

Example: Suppose \( Z_1 Z_2 Z_3 = 0 0 1 \)

Then we know \( Z = .0 0 1 ? ? ? ? \)

So we know \( .125 \leq Z \leq .25 \)

Therefore,

\[
Z \in [0,.7] \Rightarrow X_1 = a
\]

\[
Z \in [0,.49] \Rightarrow X_1 X_2 = a a
\]

\[
Z \in [0,.343] \Rightarrow X_1 X_2 X_3 = a a a
\]

However, we can’t tell whether

\[
Z \in [0,.240) \ \text{or} \ [.240,.343).
\]

Therefore, from \( Z_1 Z_2 Z_3 = 0 0 1 \), we cannot determine \( x_4 \), until more \( Z \)'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 \( K_{ZL} = [.Z_1 Z_2 \ldots Z_L 0 0 \ldots , .Z_1 Z_2 \ldots Z_L 1 1\ldots ) = \) decoding interval for \( Z_L \)

2. 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 \( K_{ZL} \)

\[ J_{Xn} \supset K_{ZL} \]

3. Decode \( Z_1 Z_2 \ldots Z_L \) into \( X_1 \ldots X_n \)

Key fact

\( Z \in J_{Yn} \) implies \( X^n = Y^n \)

\[ \cdots .Z_2 \ldots Z_L 0 0 \ldots .Z_2 \ldots Z_L 1 1 \ldots \]

\[ \begin{array}{c}
0 \\
0 \\
A_n \\
Z \\
Z \\
\vdots \\
1 \\
B_n \\
Z_L \ldots \ Z_L \ldots \ Z_L \ldots \ Z_L \ldots \ 0 \ ? \ ? \\
Z_L \ldots \ Z_L \ldots \ Z_L \ldots \ Z_L \ldots \ 1 \ ? \ ? \\
\end{array} \]

Incremental Encoding

- Assume we have already encoded \( X^{n-1} \) into \( Z_{L^{n-1}} \) and \( J_{X^{n-1}} = [A_{n-1}, B_{n-1}) \), where \( A_{n-1} \& B_{n-1} \) are dyadic.

- Binary expansions of \( A_{n-1} \) and \( B_{n-1} \) need only have been computed out to where they differ.

- 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 \( Z_{L^n} = (Z_{L^{n-1}}, Z_{L_{n+1}}, \ldots, Z_{L^n}) \)

  3. Output \( Z_{L_{n+1}}, \ldots, Z_{L_n} \) (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
   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
   JPEG2000 image coding standard
Ziv-Lempel included in
   Unix 'compress' command
   zip and gzip
   Macintosh 'stuffit' utility
   new modem standards