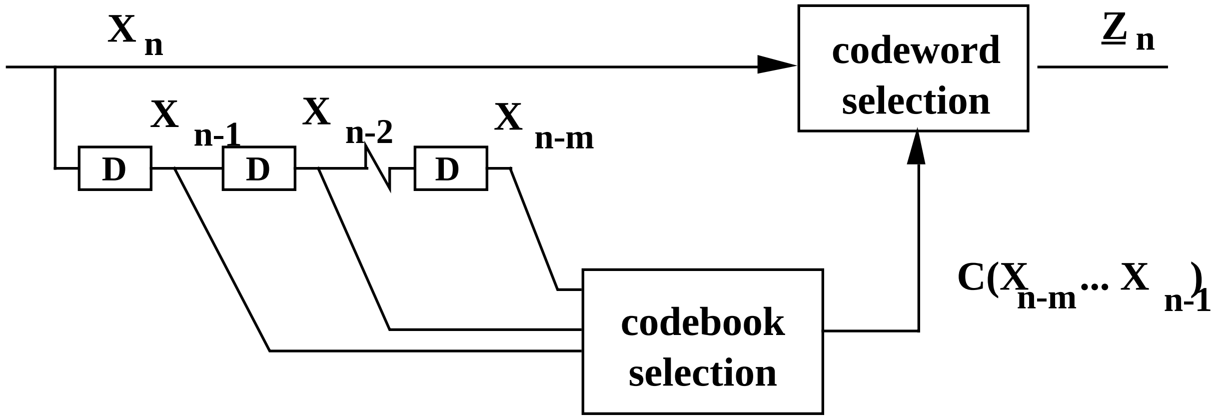
Lecture on Arithmetic Coding

EECS 651, Winter 2001

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

$$H(X_{m+1}|X_1 \ldots X_m) \leq \text{Rate}^* \leq H(X_{m+1}|X_1 \ldots X_m) + 1$$

where conditional is entropy is

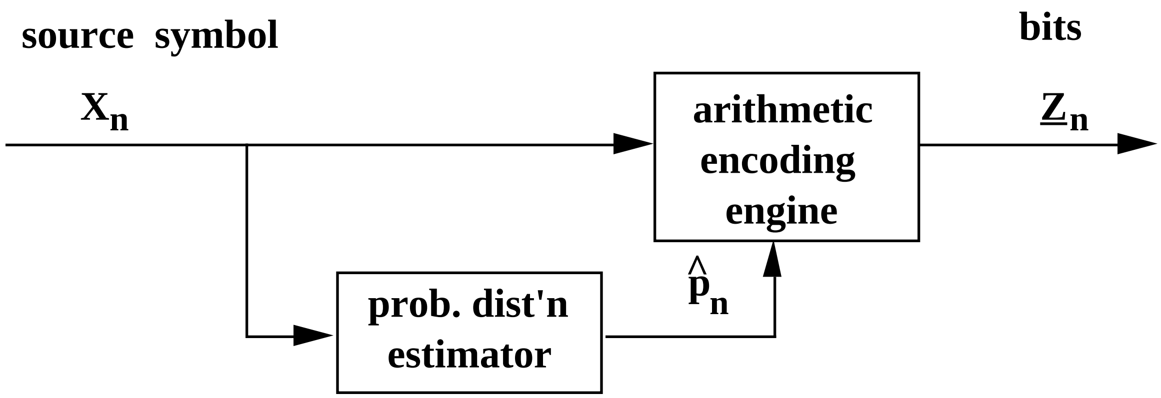
$$H(X_{m+1}|X_1 \ldots X_m) = - \sum_{x_1 \ldots x_{m+1}} P(x_1 \ldots x_{m+1}) \log P(x_{m+1}|x_1 \ldots x_m)$$

For stationary sources

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

Complexity is of the same order as block-to-variable length codes with blocklength m+1.

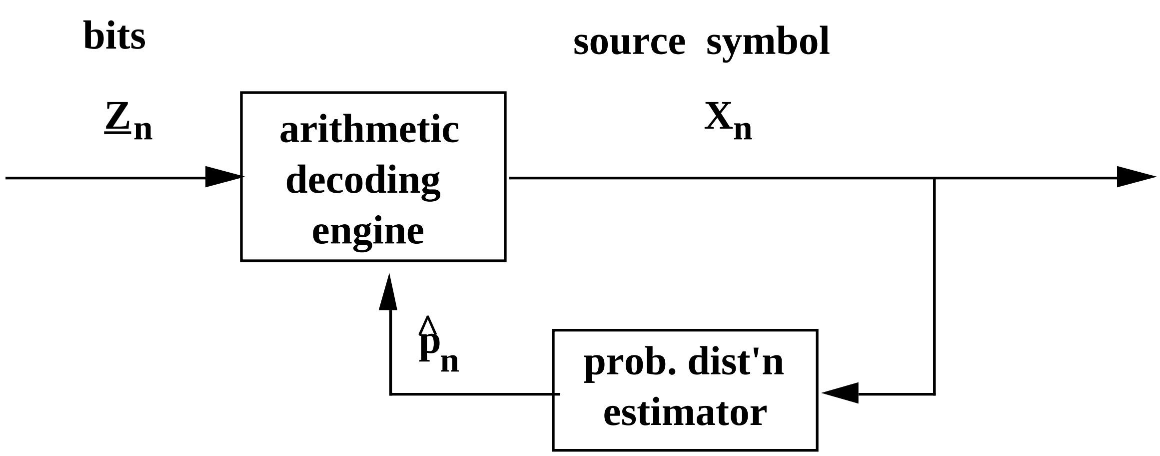
Arithmetic Coding



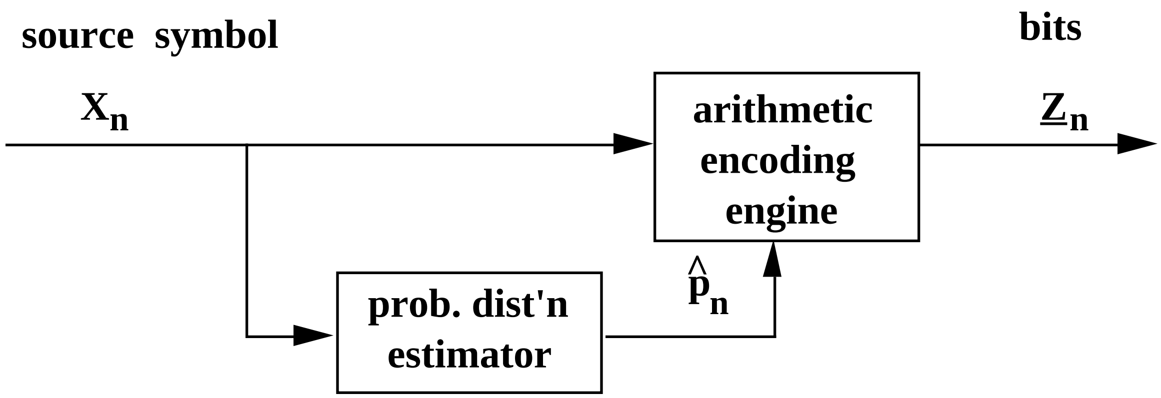
$\hat{p}_n = \{ \hat{p}_n(x) : x \in A \}$, $\hat{p}_n(x)$ = estimate of $P(X_n = x \mid \underline{X}^-)$ from past X's, where $\underline{X}^-$ denotes recent past X_n's

$\underline{Z}_n$ = bits emitted in response to X_n

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



Arithmetic Coding



Key example: When encoding sequence $(x_1, x_2, \ldots)$

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

based on $x_1,\ldots,x_{n-1}$

If source is stationary & ergodic and probability estimates are perfect

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

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

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

$$\cong H(X_{m+1}|\ X_1\ldots.X_m)$$

with high probability, by ergodic thm. (law of large numbers)

Probability Estimation

Basic idea

Maintain table of conditional frequency of occurrence of x_{m+1} after $x_1,\ldots,x_m$, all $x_1,\ldots,x_m,x_{m+1}$

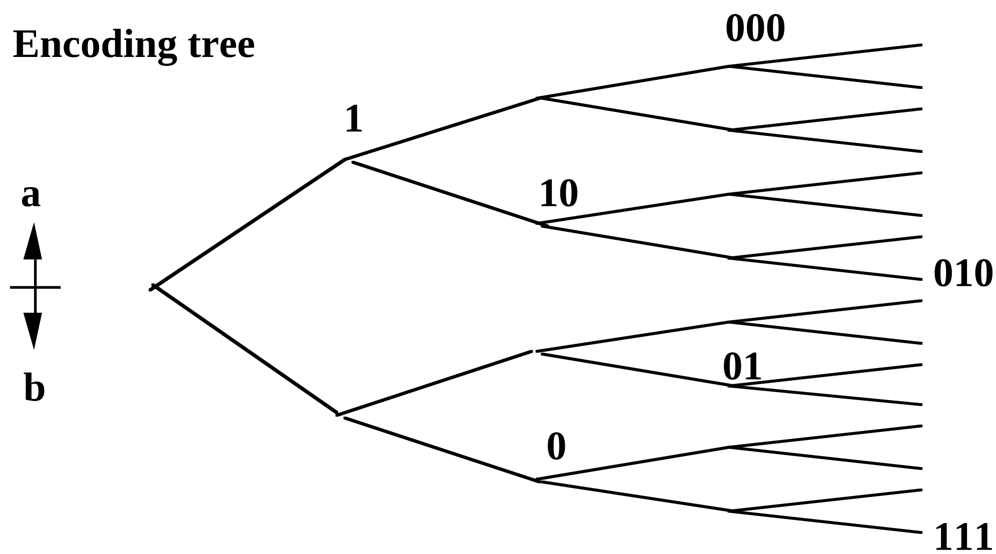
Reduce size of tables

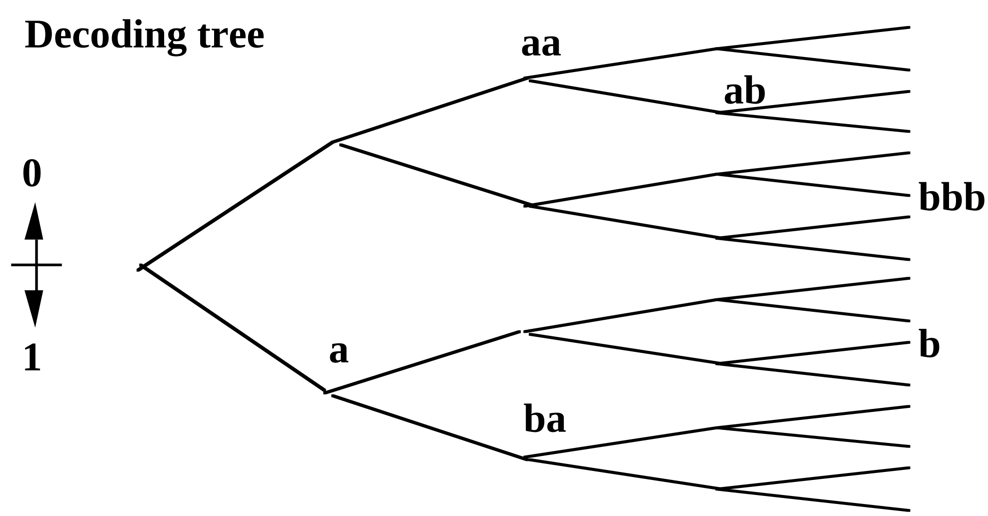
Keep only the most important conditioning events. (Lump events with similar conditional frequencies.)

Arithmetic Coding Engine

Restrict attention to binary sources.

Irregular nature of encoding and decoding.
(the following are not the actual trees)





Levels of Explanation

Infinite sequence to infinite sequence

$X_1\ X_2 \ldots \rightarrow Z_1\ Z_2 \ldots$

Finite sequence to finite sequence

$X_1 \ldots X_n \rightarrow Z_1 \ldots Z_{L_n}$

Incremental encoding and decoding

after $X_1 \ldots X_{n-1} \rightarrow Z_1 \ldots Z_{L_{n-1}}$,

$X_n \rightarrow Z_{L_{n-1}} \ldots Z_{L_n}$

Incremental encoding with finite precision arithmetic

we'll skip this

For purposes of discussion:

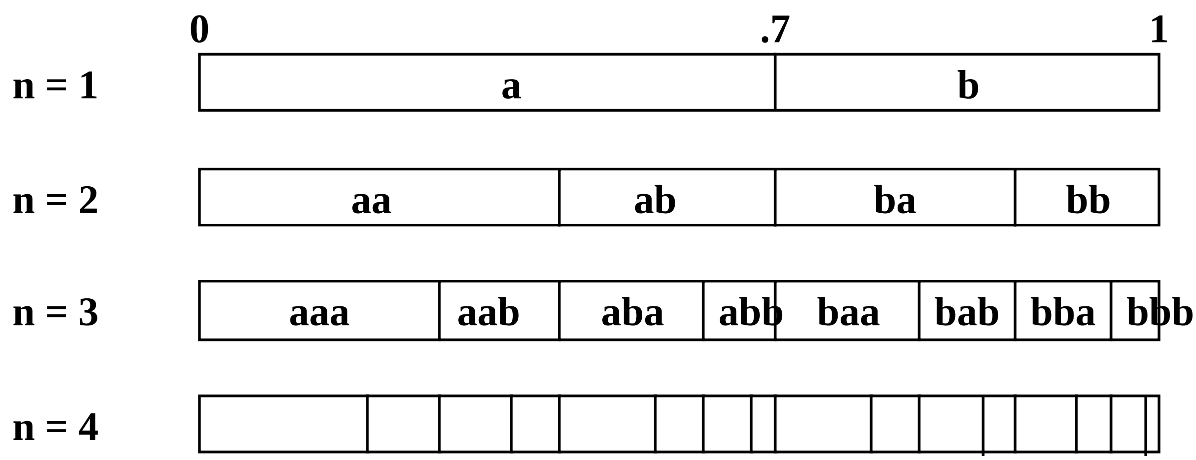
Assume binary, stationary, memoryless source with known probability distribution. (No prob. estimation.)

$P(X = a) = p$, $P(X = b) = 1-p$ ($p \cong .7$ 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.



Notation

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

Key facts

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

$$J_{X^n} \subset J_{X^{n-1}}$$

$$J_{X^n} = J_{X^{n-1}a} \cup J_{X^{n-1}b}$$

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

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 number

$$Z = J_{X_1} \cap J_{X_2} \cap J_{X_3} \ldots \qquad (Z \in [0,1])$$

$$= \lim_{n \to \infty} A_n = \lim_{n \to \infty} B_n$$

i.e.,

$$Z = .\, Z_1 Z_2 Z_3 \ldots \qquad (Z_n \in \{0,1\})$$

Cannot deducing the rate of the code from just the infinite sequence to infinite sequence encoding rule.

Facts:

1. $Z \in J_{x^n}$ **all n**

2. $Z \in J_{u^n}$ **if and only if** $u_1 \ldots u_n = x_1 \ldots x_n$

Finite sequence to finite sequence encoding

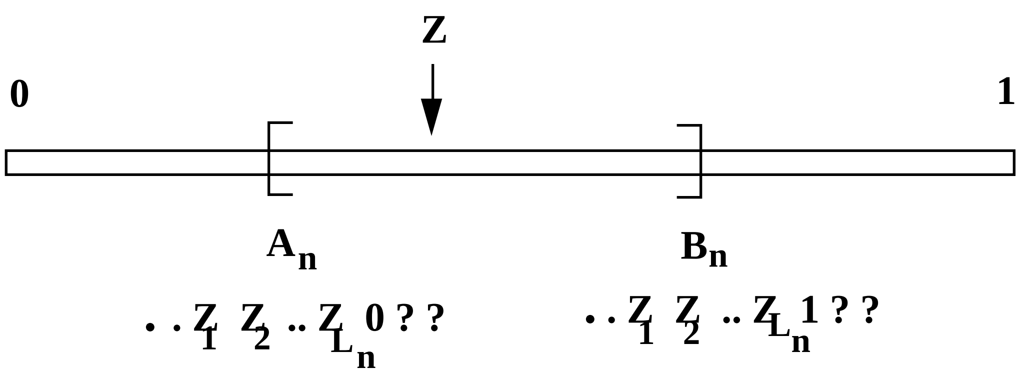
Encode $X_1 X_2 \ldots X_n$ **into** $Z_1 Z_2 \ldots Z_{L_n}$

1. **Find** $J_{X^n} = [A_n, B_n]$

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

$$A_n = .\, Z_1\, Z_2\, Z_3 \ldots Z_{L_n}\, a_{L_n+1} \ldots$$

$$B_n = .\, Z_1\, Z_2\, Z_3 \ldots Z_{L_n}\, b_{L_n+1} \ldots$$



Encoding rate

$$P(X^n) = B_n - A_n = .\,0\,0 \ldots 0\,0\,1\,?\,?\,? \qquad (L_n\ 0\text{'s})$$

$$\leq .\,0\,0 \ldots 0\,1\,0\,0 \ldots \qquad (L_n\text{-}1\ \ 0\text{'s})$$

$$= 2^{-L_n}$$

$$\Rightarrow \quad \frac{L_n}{n} \leq -\frac{1}{n} \log P(X^n) \cong H_\infty(X)$$

Finite sequence to finite sequence decoding

Given $Z_1\ Z_2\ \ldots\ Z_L$

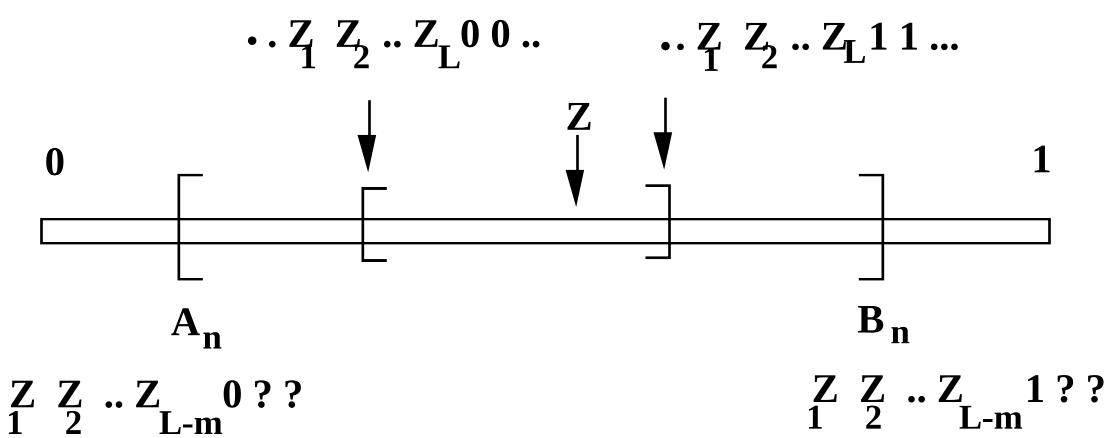
1. Let $K_{Z^L} = [\ .Z_1\ Z_2\ \ldots\ Z_L\ 0\ 0\ \ldots\ ,\ .Z_1\ Z_2\ \ldots\ Z_L\ 1\ 1\ldots\ ]$
2. Find n (as large as possible) and $X_1 \ldots X_n$ such that

   $$J_{X^n} \supset K_{Z^L}$$

3. Decode $Z_1\ Z_2\ \ldots\ Z_L$ into $X_1\ \ldots\ X_n$

Recall Key fact

$Z \in J_{u^n}$ implies $u^n = x^n$



Incremental encoding

Replace p by a dyadic fraction $p = \sum_{i=1}^{q} p_i 2^{-i}$.

Do all arithmetic base 2 .

Assume: Already encoded X^{n-1} into $Z^{L_{n-1}}$

$J_{X^{n-1}} = [A_{n-1}, B_{n-1}]$ (A_{n-1} & B_{n-1} are dyadic)

Binary expansion of A_{n-1} and B_{n-1} need only be computed out to where they disagree.

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 $Z^{L_n} = (Z^{L_{n-1}}, Z_{L_{n-1}+1}, \ldots, Z_{L_n})$
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