Lecture 10: Shannon-Fano-Elias Code, Arithmetic Code

- Shannon-Fano-Elias coding
- Arithmetic code
- Competitive optimality of Shannon code
- Generation of random variables

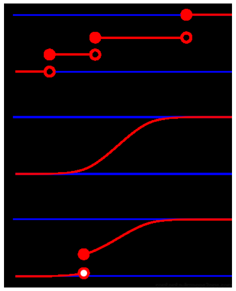
Dr. Yao Xie, ECE587, Information Theory, Duke University

CDF of a random variable

- Cumulative distribution function (CDF)

$$F(x) = \sum_{p_i < x} p_i, \quad F(x) = p(X \le x) = \int^{x} f(u)du$$

- 1) $F(x)$ is monotonic, right-continuous, 2) $F(x) \to 1$ when $x \to \infty$ and 3) $F(X) \to 0$ when $x \to -\infty$



Transform of random viable by CDF

- Random variable $F(X)$ (for X continuous) is uniformly distributed
  Proof:

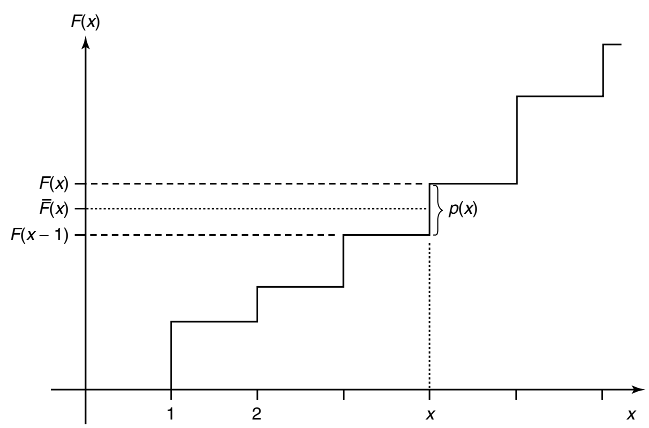
$$
\begin{aligned}
p\{F(X) \le t\} &= p\{F^{-1}[F(X)] \le F^{-1}(t)\} \\
&= p(X \le F^{-1}(t)) \\
&= F(F^{-1}(t)) = t.
\end{aligned}
$$

- This means $F^{-1}(U)$ when U is uniform[0, 1] has distribution $p(x)$

- Example: How to generate Bernoulli random variable

Shannon-Fano-Elias Coding

- Pick a number from the disjoint interval: $\bar{F}(x) = \sum_{a<x} p(a) + \frac{1}{2}p(x)$
- Truncate the real number to enough bits such that the codewords are unique
- We can show that $l(x) = \lceil \log \frac{1}{p(x)} \rceil + 1$ is enough cold length such that the codewords are unique



- Using $\lfloor \bar{F}(x) \rfloor_{l(x)}$ as the codeword $F(X)$

x	$p(x)$	$F(x)$	$\overline{F}(x)$	$\overline{F}(x)$ in Binary	$l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$	Codeword
1	0.25	0.25	0.125	0.001	3	001
2	0.25	0.5	0.375	0.011	3	011
3	0.2	0.7	0.6	$0.1\overline{0011}$	4	1001
4	0.15	0.85	0.775	$0.110\overline{0011}$	4	1100
5	0.15	1.0	0.925	$0.111\overline{0110}$	4	1110

Codewords are unique

$$F(x) - \lfloor \bar{F}(x) \rfloor_{l(x)} < \frac{1}{2^{l(x)}}$$

For $l(x) = \lceil \log \frac{1}{p(x)} \rceil + 1$

$$\begin{aligned}
\frac{1}{2^{l(x)}} &= \frac{1}{2^{\lceil \log \frac{1}{p(x)} \rceil + 1}} && (1) \\
&< \frac{1}{2} 2^{\log \frac{1}{p(x)}} = \frac{1}{2} p(x) && (2) \\
&= \bar{F}(x) - F(x-1) && (3)
\end{aligned}$$

Codes are prefix codes

- If the $\lfloor \bar{F}(x) \rfloor_{l(x)} = 0.z_1 z_2 \ldots z_l$
- If it is a prefix of another code, the code has the form

$$z^* = 0.z_1 z_2 \ldots z_l z_{l+1}$$

- this z^* lies somewhere between

$$[0.z_1 z_2 \ldots z_l, 0.z_1 z_2 \ldots z_l + \frac{1}{2^l})$$

- by construction, there is no such codeword

Summary of Shannon-Fano-Elias codes

- Code word for $x \in \mathcal{X}$ is

$$C(x) = \lfloor \bar{F}(x) \rfloor_{l(x)}$$

- $l(x) = \lceil \log \frac{1}{p(x)} \rceil + 1$

- expected code length

$$\sum p(x) l(x) = \sum p(x) \lceil \log \frac{1}{p(x)} \rceil + 1 \leq H(X) + 2 \text{ bits}$$

x	$p(x)$	$F(x)$	$\overline{F}(x)$	$\overline{F}(x)$ in Binary	$l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$	Codeword
1	0.25	0.25	0.125	0.001	3	001
2	0.5	0.75	0.5	0.10	2	10
3	0.125	0.875	0.8125	0.1101	4	1101
4	0.125	1.0	0.9375	0.1111	4	1111

- $L = 2.75$
- entropy $=$ 1.75 bits
- Huffman code $=$ 1.75 bits

- Apply Shannon-Fano-Elias coding to a sequence of random variables?

$$C(X_1X_2\cdots X_n) = ?$$

- We need joint CDF of $X_1X_2\cdots X_n$

- Arithmetic codes

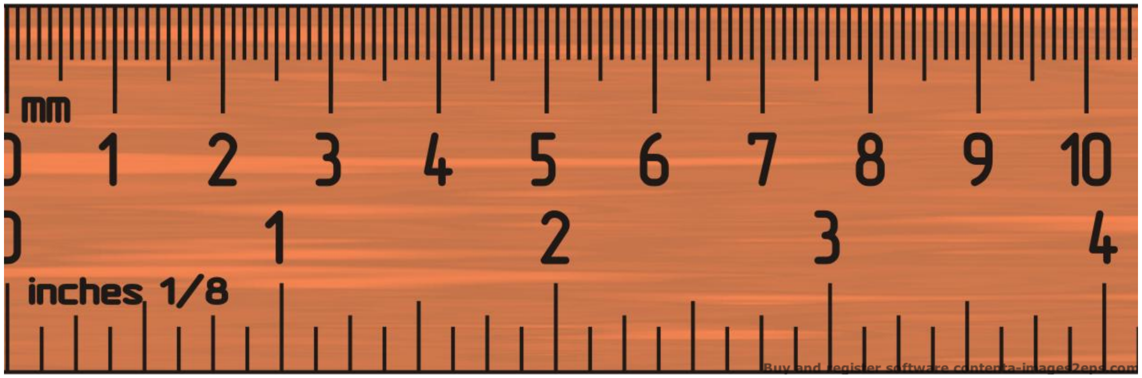
Arithmetic codes

- Huffman coding is optimal for encode a a random variable with known distribution

- Arithmetic code: use an subinterval of unit interval to code

- Basis for many practical compression schemes: JPEG, FAX

Encode and decode by a variable-scale ruler



Properties of arithmetic code

- Code a sequence of random variables on-the-fly
- Decode on-the-fly
- Code(extension to a sequence) can be calculated simply from Code(original sequence)
- Shannon-Fano-Elias for a sequence of random variables

- A message is represented by an interval of real numbers between 0 and 1
- As messages becomes longer, the interval needed to represent it becomes smaller
- The number of bits needed to specify the interval grows

Example

$\mathcal{X} = \{a, e, i, o, u, !\}$
Message $= \{eaii!\}$

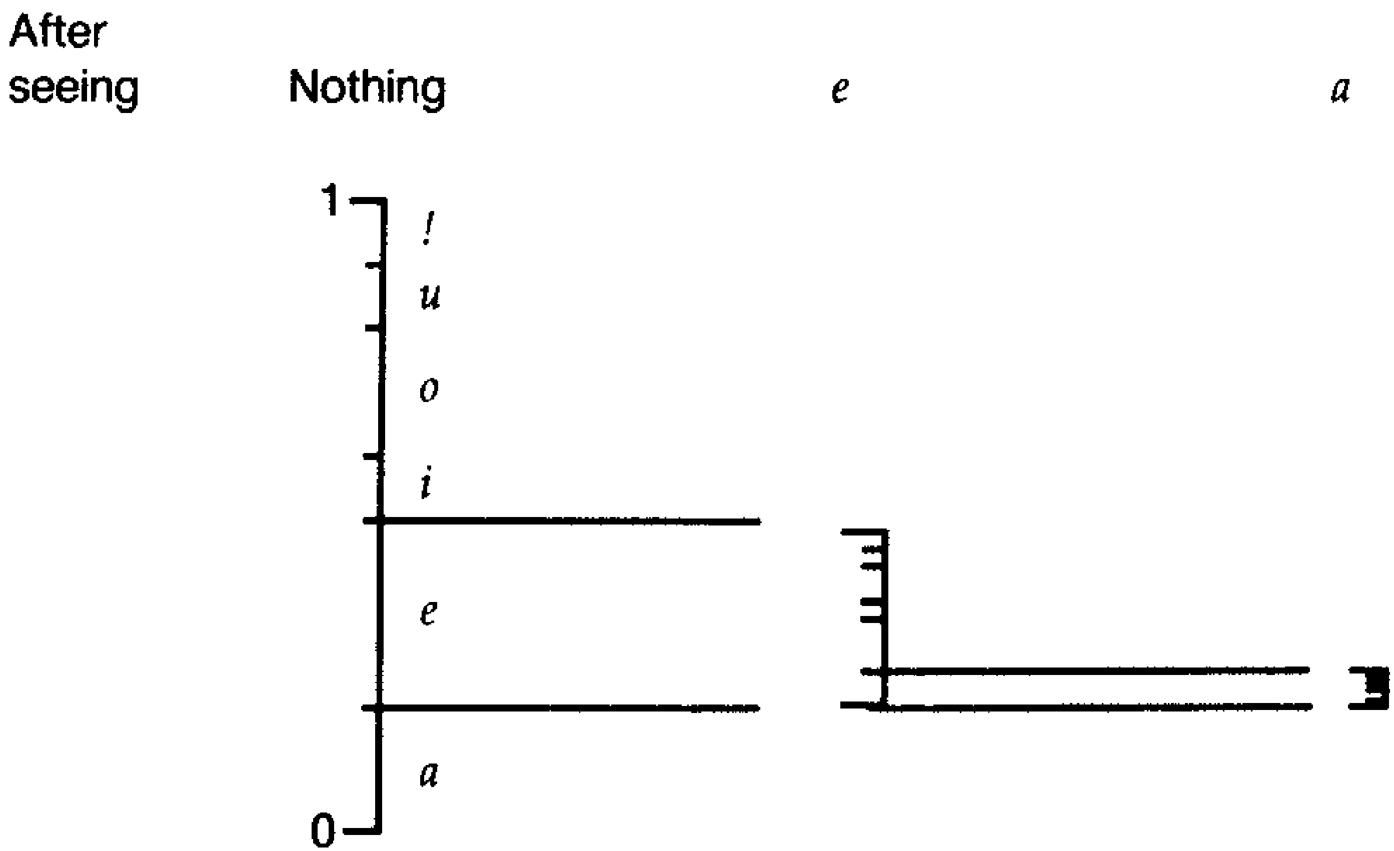
TABLE I. Example Fixed Model for Alphabet $\{a, e, i, o, u, !\}$

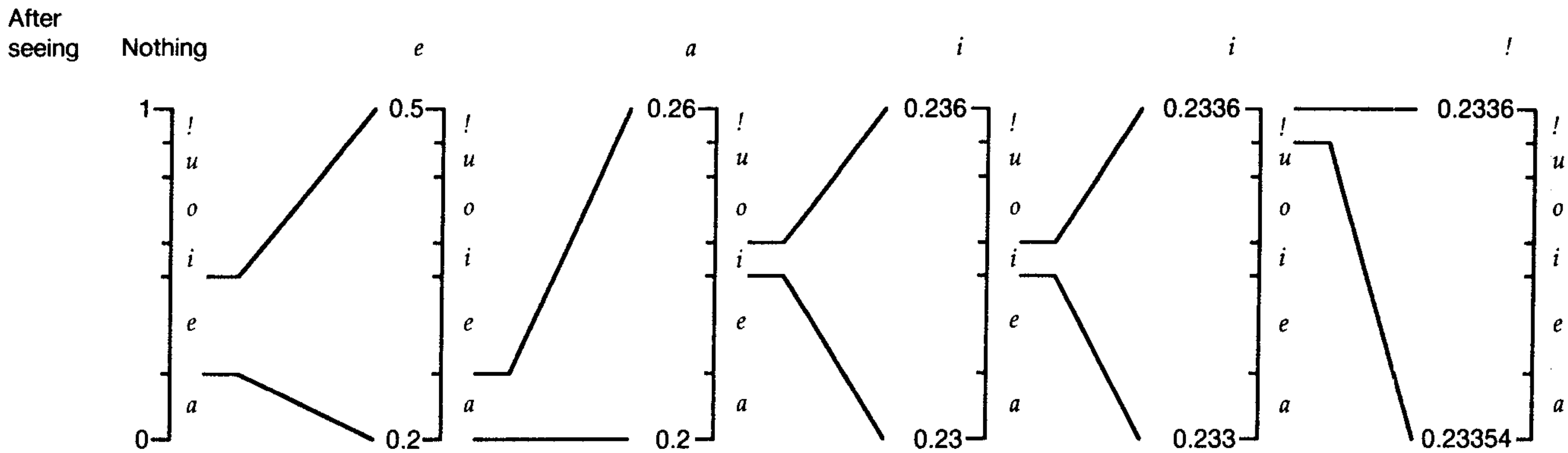
Symbol	Probability	Range
a	.2	[0, 0.2)
e	.3	[0.2, 0.5)
i	.1	[0.5, 0.6)
o	.2	[0.6, 0.8)
u	.1	[0.8, 0.9)
!	.1	[0.9, 1.0)

Reference: Arithmetic coding for data compression, by I. Witten, R. M. Neal and G. Cleary, Communications of the ACM, 1987.

Encoding

Initially	[0,	1)
After seeing *e*	[0.2,	0.5)
a	[0.2,	0.26)
i	[0.23,	0.236)
i	[0.233,	0.2336)
!	[0.23354,	0.2336)

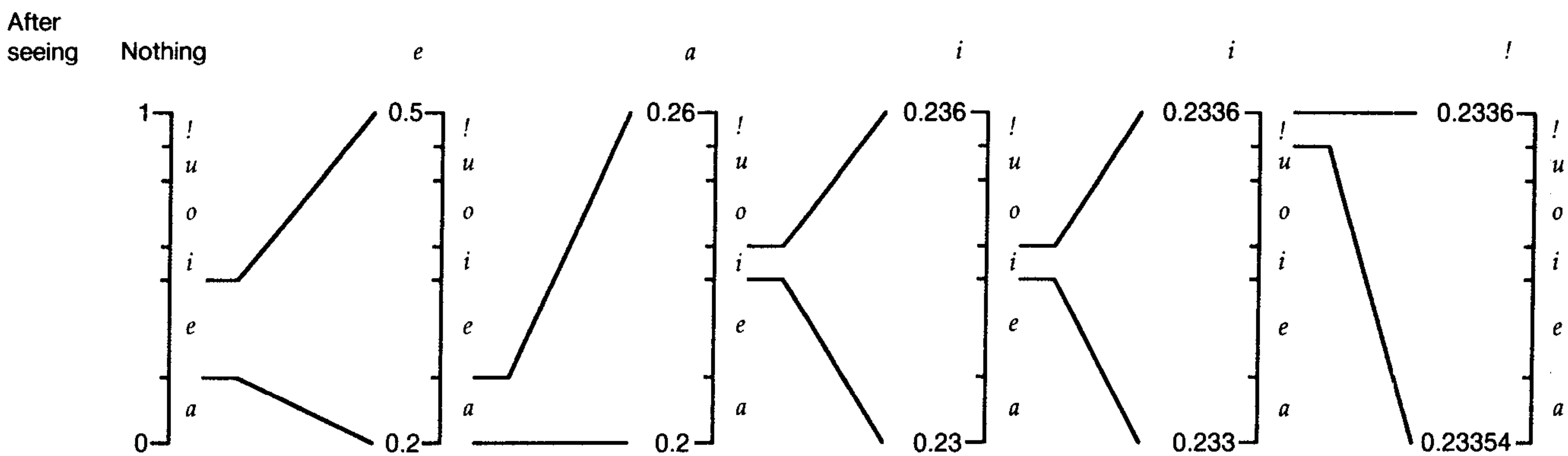

After
seeing
Nothing
e
a
1
0
!
u
o
i
e
a



The final code is a number in $[0.23354, 0.2336)$, say 0.23354

Decoding

- By repeating the same procedure: decode of 0.23354



- Given any length n and $p(x_1 x_2 \cdots x_n)$, code of length $\log \frac{1}{p(x_1 \cdots x_n)} + 2$ bits

Competitive optimality of the Shannon code

- Huffman code is optimal "on average": achieves minimum L
- Individual sequences of Huffman code may be longer
- Another optimality criterion: competitive optimality
- No other code can do much better than the Shannon code most of the time
$$p(l^*(X) < l(X)) \geq p(l^*(X) > l(X))$$
- Huffman codes are not easy to analyze this way because lack of expression for code length

Generation of random viable from coin toss

- Representing a random variable by a sequence of bits such that the expected length of representation is minimized

- Dual problem: how many fair coin tosses are needed to generate a random variable X

- Example 1: $X = a$ w.p. 1/2, $X = b$ w.p. 1/4, $X = c$ w.p. 1/4.

- $h \to a$, $th \to b$, $tt \to c$, average toss: $1.5 = H(X)$

- Dyadic distribution: $E(\text{toss}) = H(X)$

- Example 2: $X = a$ w.p. $2/3$, $X = b$ w.p. $1/3$
- $2/3 = 0.10101010101$, $1/3 = 0.01010101010$
- $h \to a$, $th \to b$, $tth \to a \cdots$
- General distribution: $H(X) \leq E(\text{toss}) < H(X) + 2$

Summary

- Coding by "interval":
  Shannon-Fano-Elias code
  Arithmetic code

- Shannon code has competitive optimality

- Generate random variable by coin tosses