

ENGR 76

Information Science and Engineering

Lecture 3: Source Coding II

Probability Review, Huffman Coding, and Block Coding

Siddharth Chandak

Course Announcements and Reminders

Reminders

- Mini-PSet 1 is due tomorrow
- Project 0 due on Friday
 - Issues with Python setup
 - Please reach out on Ed today
 - Make sure everything is working!
 - Project Session
 - Video on Canvas
- No office hours today

Recap

Symbols, Alphabet and Code

- Alphabet: set of symbols
- Code: maps each symbol to a binary sequence
- For example, (A \rightarrow 00, B \rightarrow 01, C \rightarrow 10, D \rightarrow 11)
- Properties of codes:
 - Non-singular
 - Uniquely decodable
 - Prefix-free



Prefix-Free Codes

- No codeword is a prefix of any other codeword
 - (A \rightarrow 0, B \rightarrow 10, C \rightarrow 110, D \rightarrow 111) ✓
 - (A \rightarrow 0, B \rightarrow 1, C \rightarrow 11) ✗
- Allows for instantaneous decoding
 - Each symbol can be decoded as soon as its complete codeword is read (no need to look at future bits)
- No loss of optimality
- Huffman Coding:
 - prefix-free code which minimizes average number of bits per symbol
 - **Will discuss in second half of today's class**

Information Source

Last Class

In last class,

- treated sequence of symbols as given
- did not talk about how they are generated

- A source generates a sequence of symbols each belonging to alphabet \mathcal{X}
- Model sources as **random variables**

Random Variables

Random Variable

- **Random variable:** A quantity that takes different values in alphabet \mathcal{X} with different probabilities
- Alphabet $\mathcal{X} = \{x_1, \dots, x_M\}$
 - Set of values the random variable can take
 - Discrete random variable
- Probabilities: $P(X = x_i) = p(x_i)$
 - $p(x_i)$ denotes the probability of X taking value x_i

Example 1

- X : outcome of a *fair* coin toss
 - Heads and tails are equally likely
- $P(X = H) = ?$ and $P(X = T) = ?$

Example 1

- X : outcome of a *fair* coin toss
 - Heads and tails are equally likely
- $P(X = H) = 0.5$ and $P(X = T) = 0.5$

Example II

- X : outcome of a *fair* dice roll
 - All outcomes are equally likely
- Alphabet? Probabilities?

Example II

- X : outcome of a *fair* dice roll
 - All outcomes are equally likely
- Alphabet $\mathcal{X} = \{1, 2, \dots, 6\}$
- Probabilities:
 - $P(X = 1) = 1/6, P(X = 2) = 1/6, \dots, P(X = 6) = 1/6$

Axioms of Probability

- For each symbol $x \in \mathcal{X}$: $0 \leq p(x) \leq 1$

Axioms of Probability

- For each symbol $x \in \mathcal{X}$: $0 \leq p(x) \leq 1$
- Probabilities add up to 1:

$$\sum_{x \in \mathcal{X}} p(x) = 1$$

$$\text{i.e., } p(x_1) + \dots + p(x_M) = 1$$

Axioms of Probability

- For each symbol $x \in \mathcal{X}$: $0 \leq p(x) \leq 1$
- Probabilities add up to 1:

$$\sum_{x \in \mathcal{X}} p(x) = 1$$

- Let A_1, A_2, \dots, A_n be disjoint events, then

$$P(A_1 \text{ or } A_2 \text{ or } \dots \text{ or } A_n) = P(A_1) + P(A_2) + \dots + P(A_n)$$

- Examples of events (in dice roll): $\{X = 1\}$, $\{X = 2 \text{ or } X = 3\}$, $\{X \neq 1\}$
- Disjoint events: cannot occur at the same time

Example

- Let X be the outcome of a fair dice roll:
 - What is the probability that X is even?

Example

- Let X be the outcome of a fair dice toss:
 - What is the probability that X is even?

$$\begin{aligned}P(X \text{ is even}) &= P(X = 2 \text{ or } X = 4 \text{ or } X = 6) \\&= P(X = 2) + P(X = 4) + P(X = 6) \\&= \frac{1}{6} + \frac{1}{6} + \frac{1}{6} \\&= 0.5\end{aligned}$$

Independence

Multiple Random Variables: Example

- Example: flip a fair coin twice:
 - X : outcome of first flip
 - $P(X = H) = P(X = T) = 0.5$
 - Y : outcome of second flip
 - $P(Y = H) = P(Y = T) = 0.5$
- What is $P(X = H \text{ and } Y = H)$?

Multiple Random Variables: Example

- *Natural assumption:* outcome of one coin flip does not affect other coin flip
- Flipped a fair coin twice **independently**
- $X, Y \in \{HH, HT, TH, TT\}$
 - All four outcomes equally likely
- $P(X = H \text{ and } Y = H) = \frac{1}{4}$

Definition

Two random variables X and Y are independent of each other if

$$P(X = x, Y = y) = P(X = x)P(Y = y),$$

for all values x which X can take and all values y which Y can take.

- *Intuition:* Knowing the value of one r.v. does not tell anything about the value of the other

Expectation

Expectation

- Consider random variable X taking values in $\mathcal{X} = \{x_1, \dots, x_M\}$ where each x_i is a real number
- Mean/Expectation/Average of random variable X :

$$\begin{aligned}\mathbb{E}[X] &= \sum_{x \in \mathcal{X}} p(x) \times x \\ &= p(x_1)x_1 + \dots + p(x_M)x_M\end{aligned}$$

Expectation Example

- Let X be the outcome of a fair dice toss
- What is the expected value of X ?

Expectation Example

- Let X be the outcome of a fair dice toss
- What is the expected value of X ?

$$\begin{aligned}\mathbb{E}[X] &= 1 \cdot P(X = 1) + \dots + 6 \cdot P(X = 6) \\ &= 1 \cdot \frac{1}{6} + \dots + 6 \cdot \frac{1}{6} \\ &= 3.5\end{aligned}$$

Expectation of a function of random variable

- Let X be the outcome of a fair dice toss
- Let $Z = g(X)$ for some function g
 - Z is another random variable
 - Examples: X^2 , $5X + 2$, etc.
- What is the expected value of Z ?

$$\mathbb{E}[Z] = \mathbb{E}[g(X)] = \sum_{x \in \mathcal{X}} p(x) \times g(x)$$

Back to Source Coding

Source Coding

- Consider the problem of coding a source that generates symbols randomly from alphabet \mathcal{X}
 - symbol $x \in \mathcal{X}$ is generated with probability $p(x)$
- A code maps each symbol x in \mathcal{X} to codeword $c(x)$ with length $\ell(x)$
- What is the average number of bits per source symbol?

Expected Code Length

- Average code length or Expected code length
 - Average number of bits per symbol for given source distribution and given code

$$\begin{aligned}\bar{\ell} &= \mathbb{E}[\ell(X)] \\ &= \sum_{x \in \mathcal{X}} p(x) \times \ell(x)\end{aligned}$$

Example

Symbol	Probability	Codeword
A	$\frac{1}{2}$	0
B	$\frac{1}{4}$	10
C	$\frac{1}{8}$	110
D	$\frac{1}{8}$	111

$$\bar{\ell} = \sum_{x \in \mathcal{X}} p(x) \times \ell(x) = ?$$

Example

Symbol	Probability	Codeword
A	$\frac{1}{2}$	0
B	$\frac{1}{4}$	10
C	$\frac{1}{8}$	110
D	$\frac{1}{8}$	111

$$\begin{aligned}\bar{\ell} &= \sum_{x \in \mathcal{X}} p(x) \times \ell(x) \\ &= p(\text{A})\ell(\text{A}) + p(\text{B})\ell(\text{B}) + p(\text{C})\ell(\text{C}) + p(\text{D})\ell(\text{D}) \\ &= \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 \\ &= 1.75\end{aligned}$$

Some Intuition behind $\bar{\ell}$

- Suppose a source generates a large sequence of symbols X_1, X_2, \dots, X_N
 - The symbols are identically distributed
- Total number of bits used is $\ell(X_1) + \dots + \ell(X_N) = \sum_{x \in \mathcal{X}} \ell(x)n(x)$
 - $n(x)$ denotes the number of times symbol x appears
- Average number of bits per symbol for this sequence is $\sum_{x \in \mathcal{X}} \ell(x)f(x)$
 - $f(x) = n(x)/N$ denotes the frequency of symbol x in this sequence

Some Intuition behind $\bar{\ell}$

- Suppose a source generates a large sequence of symbols X_1, X_2, \dots, X_N

- Identically distributed

- **For large N** (under suitable assumptions):

- Empirical frequencies:

$$f(x) \approx p(x)$$

- Average number of bits per symbol:

$$\sum_{x \in \mathcal{X}} \ell(x) f(x) \approx \sum_{x \in \mathcal{X}} \ell(x) p(x) = \bar{\ell}$$

Huffman Coding

- For a given source distribution, want to find the prefix-free code which minimizes the average code length $\bar{\ell}$
 - **Huffman coding algorithm**

Huffman Codes

Huffman Coding Algorithm

1. **Initialize Nodes:** Start with a list of leaf nodes
{(symbol, probability)}
2. While more than one node left:
 - 2.1 Pick **2 nodes with the lowest probabilities**
 - 2.2 **Merge** them into a new node
 - the two nodes become its children
 - the new node's probability is the **sum** of their probabilities
3. **Finish:** Last remaining node is the root of the Huffman tree
4. **Assign Codes:**
 - 4.1 Label edges with 0/1
 - 4.2 The codeword for a symbol is the bit sequence along the path from root to leaf

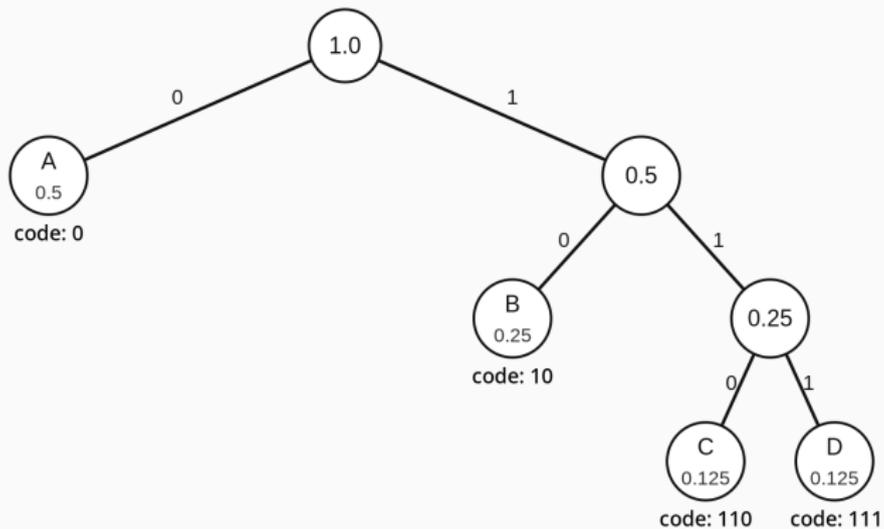
Example I

Probabilities:

- A: 0.5
- B: 0.25
- C: 0.125
- D: 0.125

Example 1

Huffman tree for A:0.5, B:0.25, C:0.125, D:0.125



Example I

Symbol	Probability	Codeword
A	$\frac{1}{2}$	0
B	$\frac{1}{4}$	10
C	$\frac{1}{8}$	110
D	$\frac{1}{8}$	111

$$\bar{\ell} = 1.75$$

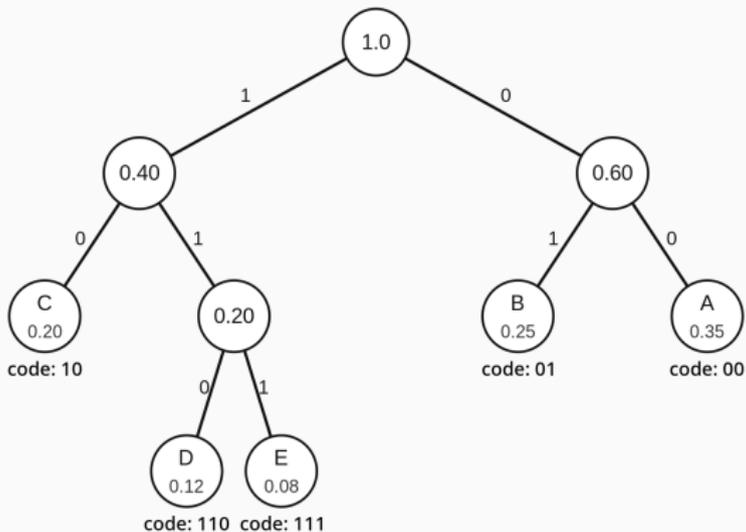
Example II

Probabilities:

- A: 0.35
- B: 0.25
- C: 0.2
- D: 0.12
- E: 0.08

Example II

Huffman tree for A:0.35, B:0.25, C:0.20, D:0.12, E:0.08



Example II

Symbol	Probability	Codeword
A	0.35	00
B	0.25	01
C	0.2	10
D	0.12	110
E	0.08	111

$$\bar{\ell} = 2.2$$

Theorem

Huffman algorithm always outputs a prefix-free code.

- Only the leaves of the tree are used for the codewords (rather than using internal nodes), no codeword is a prefix of another codeword.

Fact

For any source distribution, the Huffman code is optimal, i.e., has the smallest $\bar{\ell}$ among all prefix-free codes.

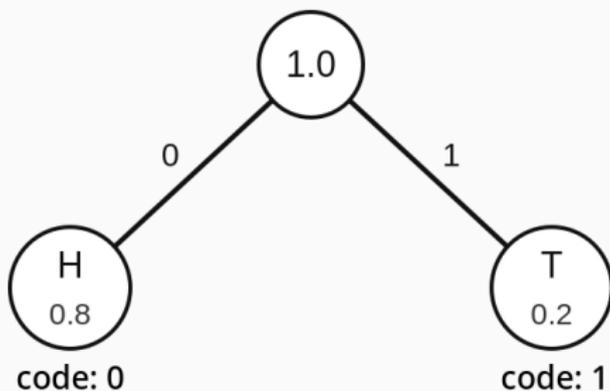
Another Example

Probabilities (biased coin flip):

- H: 0.8
- T: 0.2

Another Example

Huffman tree for H:0.8, T:0.2



- Average number of bits per symbol is 1
- Can we do anything?
 - Recall that we are encoding *sequences*

Block Coding

- Encode blocks of symbols together
- **Assumption:** X_1, X_2, X_3, \dots are independent and identically distributed
 - Example: repeated coin flips

Previous Example

- Probabilities (biased coin flip):
 - H: 0.8
 - T: 0.2

X_1X_2	Probability	Codeword (Huffman code)
HH	?	?
HT	?	?
TH	?	?
TT	?	?

Previous Example

- Probabilities (biased coin flip):
 - H: 0.8
 - T: 0.2

X_1X_2	Probability	Codeword (Huffman code)
HH	$0.8 \times 0.8 = 0.64$	0
HT	$0.8 \times 0.2 = 0.16$	11
TH	$0.2 \times 0.8 = 0.16$	100
TT	$0.2 \times 0.2 = 0.04$	101

Previous Example

X_1X_2	Probability	Codeword (Huffman code)
HH	0.64	0
HT	0.16	11
TH	0.16	100
TT	0.04	101

- Average number of bits per block =

$$\begin{aligned}\bar{\ell} &= 0.64 \times 1 + 0.16 \times 2 + 0.16 \times 3 + 0.04 \times 3 \\ &= 1.56\end{aligned}$$

Average number of bits per symbol

- Average number of bits per block =

$$\begin{aligned}\bar{\ell} &= 0.64 \times 1 + 0.16 \times 2 + 0.16 \times 3 + 0.04 \times 3 \\ &= 1.56\end{aligned}$$

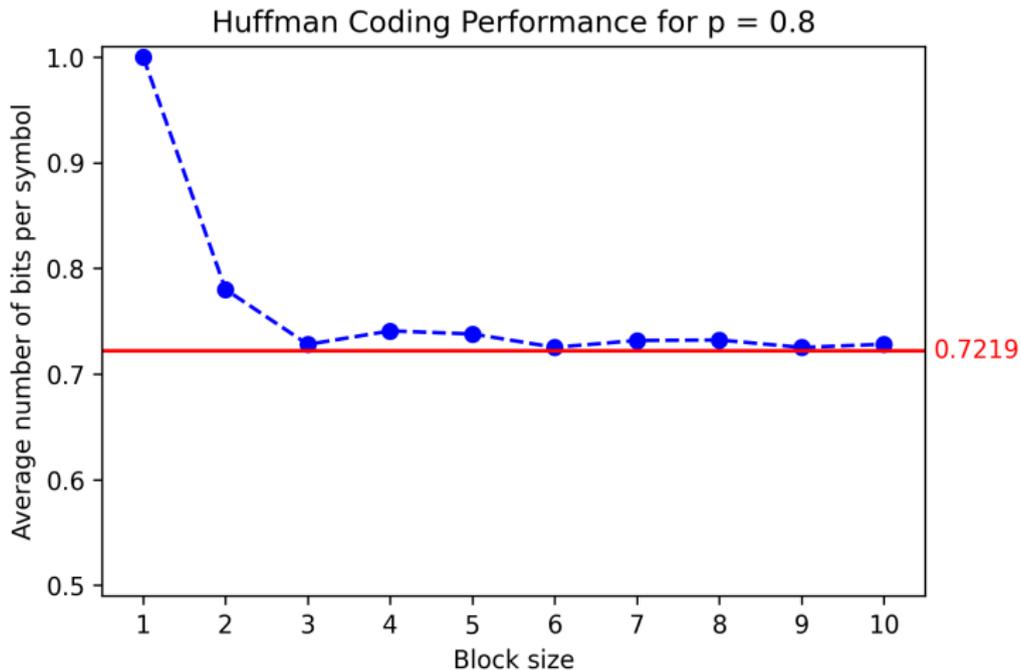
- Average number of bits per symbol = $\frac{\text{Average number of bits per block}}{\text{Block size}}$

$$\text{Average number of bits per symbol} = \frac{\bar{\ell}}{2} = 0.78$$

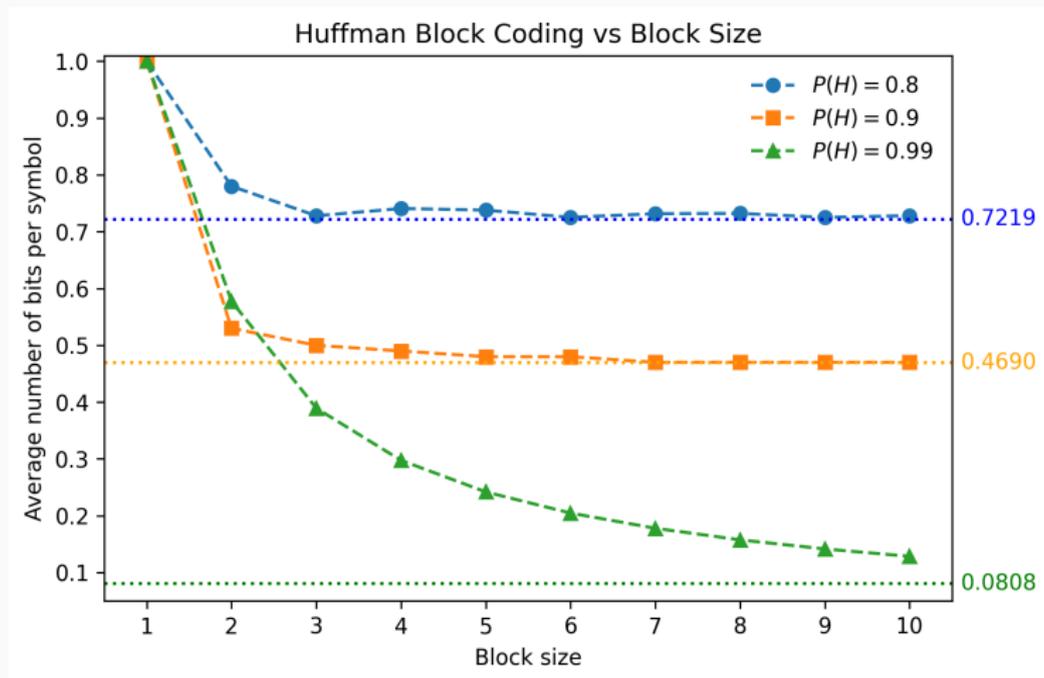
- Using blocks of size 2 we achieve 0.78 bits per symbol
 - Improvement over 1 bit per symbol achieved without block coding

Increasing block lengths

$H : 0.8$ and $T : 0.2$



Increasing block lengths



What are these lower limits?

Thank You!