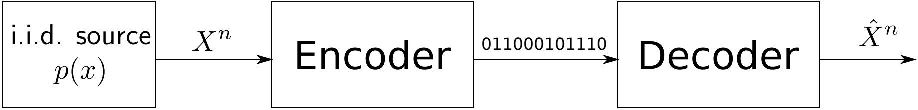
EEE 551 Information Theory (Spring 2022)

Chapter 5: Data Compression

Source Coding



- A **source code** is composed of two functions:
  1. An **encoding function** f that maps a sequence $X^n \in \mathcal{X}^n$ to an integer $M = f(X^n) \in \{1, 2, 3, \ldots, 2^{nR}\}$
  2. A **decoding function** g that maps an integer $M \in \{1, 2, \ldots, 2^{nR}\}$ to a sequence $\hat{X}^n = g(M) \in \mathcal{X}^n$
- There is a straightforward mapping between $\{1, 2, \ldots, 2^{nR}\}$ and $\{0, 1\}^{nR}$
- R is the **rate** of the code — number of bits per X symbol
- The **probability of error is**

$$\begin{aligned} P_e &= \Pr(\hat{X}^n \neq X^n) \\ &= \sum_{x^n} p(x^n) \Pr(\hat{X}^n \neq X^n | X^n = x^n) \\ &= \sum_{x^n} p(x^n) \mathbf{1}(g(f(x^n)) \neq x^n) \end{aligned}$$

- We say a rate R is **achievable** if, for any $\epsilon > 0$, there exists a length n and a code (f, g) where $P_e \leq \epsilon$
- Let $R_{\min}$ be the infimum of all achievable rates

Operational vs. Information Definitions

- An **operational** definition gives the **problem**: it describes the space of possible solutions, and defines a quantity as the minimal (or maximal) value of a performance metric over all possible solutions

  e.g. $R_{\min}$

- An **information** definition gives the **solution**: it describes a computable mathematical function of the problem parameters

  e.g. $H(X)$

Information theoretic results state that a given operational quantity is equal to a given information quantity

Source Coding Result

Theorem

$R_{\min} = H(X)$

To prove this, we need to prove two things:

- **Achievability:** If $R > H(X)$, then R is achievable
- **Converse:** If R is achievable, then $R \geq H(X)$

Achievability proof

- Assume $R > H(X)$
- Let ϵ be small enough so that $H(X) + \epsilon \leq R$
- $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)} \leq 2^{nR}$
- For each $x^n \in A_\epsilon^{(n)}$, let $\phi(x^n)$ be a unique index in $\{1, \ldots, 2^{nR}\}$
- Define the encoding and decoding functions

$$f(x^n) = \begin{cases} \phi(x^n), & x^n \in A_\epsilon^{(n)} \\ 1, & x^n \notin A_\epsilon^{(n)} \end{cases}$$

$$g(m) = \phi^{-1}(m)$$

- If $X^n \in A_\epsilon^{(n)}$, then $\hat{X}^n = X^n$; i.e. an error can only occur for an atypical sequence
- $P_e \leq \Pr(X^n \notin A_\epsilon^{(n)}) \leq \epsilon$ for sufficiently large n

Converse proof

- Assume R is achievable
- Let (f, g) be any code with rate R
- $X^n \to M \to \hat{X}^n$ is a Markov chain, so by Fano's inequality,

$$H(X^n|M) \leq 1 + P_e \log(|\mathcal{X}^n|) = 1 + nP_e \log|\mathcal{X}|$$

- Consider the chain of inequalities

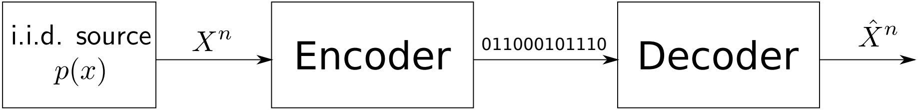
$$\begin{aligned} nR &\geq H(M) \\ &= I(X^n; M) + H(M|X^n) \\ &\geq I(X^n; M) \\ &= H(X^n) - H(X^n|M) \\ &\geq H(X^n) - (1 + nP_e \log|\mathcal{X}|) \\ &= nH(X) - 1 - nP_e \log|\mathcal{X}| \end{aligned}$$

- Divide by n:

$$R \geq H(X) - \frac{1}{n} - P_e \log|\mathcal{X}|$$

- If R is achievable, then P_e can be made arbitrarily small, so we must have $R \geq H(X)$

Fixed-to-Variable Source Codes



- A **fixed-to-variable source code** C is a function mapping $\mathcal{X}^n$ to the set of finite-length bit strings

$$\{0, 1, 00, 01, 11, 10, 000, 001, 010, 011, \ldots\}$$

  Given sequence x^n, $C(x^n)$ is the **codeword** for x^n with length $\ell(x^n)$

- Unlike fixed-to-fixed source codes, with fixed-to-variable codes there are no errors — just longer codewords
- The **expected length** of a code C is given by

$$L(C) = \mathbb{E}[\ell(X^n)] = \sum_{x^n \in \mathcal{X}^n} p(x^n)\ell(x^n)$$

- A code is **uniquely decodable** if no two sequences map to the same codeword, i.e. if $x^n \neq x'^n$, then $C(x^n) \neq C(x'^n)$

Single-letter source codes: Example 1

Let $n = 1$, so the code acts on the single letter X

$\mathcal{X} = \{1, 2, 3, 4\}$, where

$$
\begin{aligned}
p(1) &= \frac{1}{2}, \quad \text{codeword } C(1) = 0 \\
p(2) &= \frac{1}{4}, \quad \text{codeword } C(2) = 10 \\
p(3) &= \frac{1}{8}, \quad \text{codeword } C(3) = 110 \\
p(4) &= \frac{1}{8}, \quad \text{codeword } C(4) = 111
\end{aligned}
$$

Expected length is $L(C) = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1.75$

Moreover, $H(X) = 1.75$

Single-letter source codes: Example 2

$\mathcal{X} = \{1, 2, 3\}$, where

$$p(1) = \frac{1}{3}, \quad \text{codeword } C(1) = 0$$
$$p(2) = \frac{1}{3}, \quad \text{codeword } C(2) = 10$$
$$p(3) = \frac{1}{3}, \quad \text{codeword } C(3) = 11$$

Expected length is $L(C) = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 2 = \frac{5}{3} \approx 1.67$

$H(X) = \log 3 \approx 1.58$

Codes can get arbitrarily close to $H(X)$ (but no lower) by compressing **sequences**

Code Extensions

- Given a code $C(x)$, an extension $C^*(x^n)$ is given by

$$C^*(x^n) = C(x_1)C(x_2)\cdots C(x_n)$$

- Can also extend codes on sequences, e.g. given $C(x_1x_2)$,

$$C^*(x^n) = C(x_1x_2)C(x_3x_4)\cdots C(x_{n-1}x_n)$$

- Decodability for code extensions is a problem if the boundaries between codewords are ambiguous
- **Example:**

$$\begin{aligned} C(1) &= 0 \\ C(2) &= 010 \\ C(3) &= 01 \\ C(4) &= 10 \end{aligned}$$

The string 0010 can be parsed as $0, 010$ or $0, 0, 10$ or $0, 01, 0$

In this case, C is uniquely decodable but its extension is not

Prefix Codes

- A **prefix code** is one for which no codeword is a prefix of another codeword
- The extension of any prefix code is always uniquely decodable, because there is no ambiguity in string parsing
- **Example:**

$$C(1) = 0$$
$$C(2) = 10$$
$$C(3) = 110$$
$$C(4) = 111$$

0110100 can only be parsed as $0, 110, 10, 0$

Kraft Inequality

Theorem

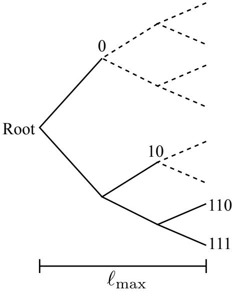
For any prefix code, the codeword lengths $\ell_1, \ell_2, \ldots, \ell_m$ *must satisfy*

$$\sum_{i=1}^{m} 2^{-\ell_i} \leq 1.$$

Conversely, given a set of codeword lengths that satisfy this inequality, there exists a prefix code with these lengths

Proof:

- Any prefix code can be represented by a binary tree where the leaves are the codewords, and no codeword is an ancestor of another codeword
- Let $\ell_{\max}$ be the length of the longest codeword (i.e. maximum depth of the tree)



- Consider all tree nodes at depth $\ell_{\max}$
- A codeword at level ℓ_i has $2^{\ell_{\max}-\ell_i}$ descendants at depth $\ell_{\max}$
- The descendants of each codeword are disjoint, so

$$\sum_{i=1}^{m} 2^{\ell_{\max}-\ell_i} \leq 2^{\ell_{\max}}$$

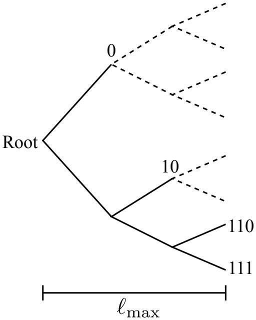
$$\implies \quad \sum_{i=1}^{m} 2^{-\ell_i} \leq 1$$

Conversely, given any codeword lengths $\ell_1, \ldots, \ell_m$ satisfying the Kraft inequality, we can construct a tree (i.e. a prefix code) as follows:

- Label the first node of depth ℓ_1 as codeword 1, and remove its descendants
- Label the first remaining node of depth ℓ_2 as codeword 2, and remove its descendants
- Continue for all lengths

Relation to yes/no questions

Every prefix code corresponds to a strategy of asking yes/no questions to determine X (yes=1, no=0):



Thus, finding the minimum expected number of yes/no questions is equivalent to finding the prefix code with minimum expected length

Converse bound on average codeword length

For any prefix code C, the expected length satisfies

$$L(C) \geq H(X)$$

with equality iff $2^{-\ell_i} = p_i$ for all i

Proof: Let $c = \sum_i 2^{-\ell_i}$. By the Kraft inequality, $c \leq 1$.

$$\begin{aligned} L(C) - H(X) &= \sum_i p_i \ell_i + \sum_i p_i \log p_i \\ &= -\sum_i p_i \log 2^{-\ell_i} + \sum_i p_i \log p_i \\ &= \sum_i p_i \log \frac{p_i}{2^{-\ell_i}} \\ &= \sum_i p_i \log \frac{p_i}{2^{-\ell_i}/c} - \log c \\ &= D(\mathbf{p} \| 2^{-\ell_i}/c) - \log c \\ &\geq -\log c \geq 0 \end{aligned}$$

More generally, a code $C(x^n)$ must satisfy $\frac{1}{n} L(C) \geq \frac{1}{n} H(X^n) = H(X)$

Achievability proof for fixed-to-variable codes

For any X, there exists a prefix code $C(x)$ where

$$H(X) \le L(C) < H(X) + 1$$

Proof:

- Recall that for any set of lengths ℓ_i that satisfy the Kraft inequality, there exists a prefix code with these lengths
- Let $\ell_i = \left\lceil \log \frac{1}{p_i} \right\rceil$ for all i
- Check Kraft: $\sum_i 2^{-\ell_i} = \sum_i 2^{-\lceil \log \frac{1}{p_i} \rceil} \le \sum_i 2^{-\log \frac{1}{p_i}} = \sum_i p_i = 1$
- The code lengths satisfy

$$\begin{aligned} & \log \frac{1}{p_i} \le \ell_i < \log \frac{1}{p_i} + 1 \\ \implies \quad & H(X) \le L(C) < H(X) + 1 \end{aligned}$$

Note: This result implies there is a code $C(x^n)$ such that

$$nH(X) \le L(C) < nH(X) + 1 \quad \implies \quad H(X) \le \frac{1}{n} L(C) < H(X) + \frac{1}{n}$$

Length per symbol is arbitrarily close to the entropy

Huffman Codes

The **Huffman code** has the optimal expected length over all prefix codes

Steps to construct Huffman code:

1. Merge the two lowest-probability symbols into one symbol
2. Repeat step (1) until all symbols are merged
3. Choose code based on resulting binary tree

Proof of optimality in Cover-Thomas

Example:

X	1	2	3	4	5
Probability	0.25	0.25	0.2	0.15	0.15
Huffman codeword length	2	2	2	3	3
Huffman codeword	00	10	11	010	011
Shannon codeword length[1]	2	2	3	3	3
Shannon codeword	00	01	100	110	111

Expected Huffman length: 2.3
Expected Shannon length: 2.5
Entropy: 2.286

[1] Where $\ell_i = \lceil \log 1/p_i \rceil$