# Data Compression

Limit of Information Compression

Radu Trîmbiţaş

October, 2012

**Outline**

## Contents

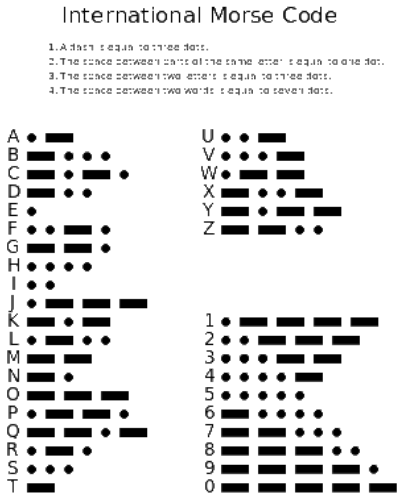


## 1 Examples of codes

**Examples of codes**

$X : S \to \mathcal{X}$ RV, $X \sim p(x)$ pmf, $\mathcal{D}^*$ the set of finite-length strings of symbols from a $D$-ary alphabet

**Definition 1.** A *source code* for $X$ is a mapping $C : \mathcal{X} \to \mathcal{D}^*$; $C(x)$ is the codeword corresponding to $x$ and $\ell(x)$ is the length of $C(X)$.




Ex: $C(red) = 00$, $C(blue) = 11$ is a source code for $\mathcal{X} = \{red, blue\}$ with alphabet $\mathcal{D} = \{0, 1\}$.

**Definition 2.** The *expected length* $L(C)$ of a source code $C(x)$ for a RV $X$ with pmf $p(x)$ is

$$L(C) = \sum_{x \in \mathcal{X}} p(x)\ell(x), \tag{1}$$

where $\ell(x)$ is the length of the codeword associated to $x$.

w.l.o.g. we can assume $\mathcal{D} = \{0, 1, \ldots, D-1\}$.

*Example* 3. $X = \begin{pmatrix} 1 & 2 & 3 & 4 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{pmatrix}$, the codewords $C(1) = 0$, $C(2) = 10$, $C(3) = 110$, $C(4) = 111$. The entropy of $X$ is $H(X) = 1.75$ bits; the expected length $L(C) = E(\ell(X)) = 1.75$. Any sequence of of bits can be uniquely decoded into a sequence of symbols of $X$. The bit string 0110111100110 is decoded as 134213.

*Example* 4. $X = \begin{pmatrix} 1 & 2 & 3 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$, the codewords $C(1) = 0$, $C(2) = 10$, $C(3) = 11$. The code is uniquely decodable. $H(X) = \log 3 = 1.58$ bits, but $L(C) = 1.66$ bits $> H(X)$.

*Example* 5 (Morse code). Code for English alphabet: dot, dash, letter space,word space. Short sequences represent frequent letters (e.g. dot is E); long sequences represent infrequent letters (dash, dash,dot, dash is Q). It is not optimal.

**Definition 6.** A code is *nonsingular* if every element of $X$ maps into a different string in $\mathcal{D}^*$, i.e.

$$x \neq x' \Longrightarrow C(x) \neq C(x'). \tag{2}$$

Nonsingularity suffices for an unambiguous description of a single value of $X$. For sequences of values we can ensure decodability by adding a special symbol (a "comma") between any two code words. This is inefficient.

**Definition 7.** The *extension* $C^*$ of a code $C$ is a mapping $C^* : \mathcal{X}^n \to \mathcal{D}^*$ defined by

$$C(x_1 x_2 \dots x_n) = C(x_1)C(x_2)\dots C(x_n). \tag{3}$$

*Example* 8. If $C(x_1) = 00$ and $C(x_2) = 11$, then $C(x_1 x_2) = 0011$.

**Definition 9.** A code is *uniquely decodable* if its extension is nonsingular.

Any encoded string in a uniquely decodable code has only one possible source string producing it. However, one must look at the entire string to determine even the first symbol in the corresponding source string.

**Definition 10.** A code is called a *prefix code* or an *instantaneous code* if no codeword is a prefix of any other codeword.

An instantaneous code can be decoded without reference to future codeword: the symbol $x_i$ can be decoded as soon as we come to the end of the codeword corresponding to it. An instantaneous code is a *self-punctuating code*: we can look at the sequence code symbols and add comas to separate codewords without looking at other symbols. Ex: 01011111010 $\to$ 0,10,111,110,10. Figure 1 shows the relationship between these codes.

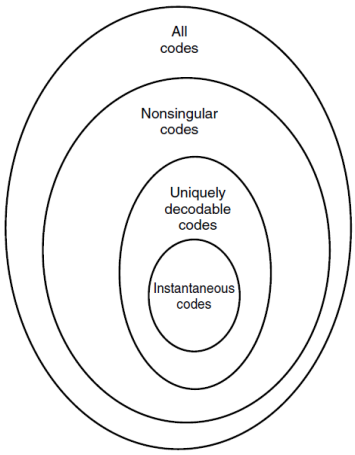



Figure 1: Classes of codes

*Examples* 11.

- $C(E, F, G, H) = (0, 1, 00, 11)$ $\overline{IU}$
- $C(E, F) = (0, 101)$ $IU$

| X | Singular | Nonsingular, but not Uniquely Decodable | Uniquely Decodable, but not Instantaneous | Instantaneous |
|---|---|---|---|---|
| 1 | 0 | 0 | 10 | 0 |
| 2 | 0 | 010 | 00 | 10 |
| 3 | 0 | 01 | 11 | 110 |
| 4 | 0 | 10 | 110 | 111 |

Table 1: Classes of codes

- $C(E,F) = (1, 101)$ $\bar{I}U$
- $C(E,F,G,H) = (00, 01, 10, 11)$ $IU$
- $C(E,F,G,H) = (0, 01, 011, 111)$ $\bar{I}U$

# 2 Kraft Inequality

## 2.1 Kraft Inequality

**Kraft Inequality**

- Aim: to construct instantaneous codes of minimum expected length
- We cannot assign short codewords to all source symbols and still be prefix-free
- The set of codeword lengths possible for instantaneous code is limited by the following inequality:

**Theorem 12** (Kraft inequality)**.** *For any instantaneous code (prefix code) over an alphabet of size D, the codeword lengths $\ell_1, \ell_2, \ldots, \ell_m$ must satisfy the inequality*

$$\sum_i D^{-\ell i} \leq 1. \tag{4}$$

*Conversely, given a set of codeword lengths that satisfy this inequality, there exists an instantaneous code with these word lengths.*

**Proof.**

**Necessity.** We construct a $D$-ary tree; the branches represent the symbols of the codeword. Each code is represented by a leaf on the tree. The path from the root traces out the symbols of the codeword. See an example for $D = 2$ in Figure 2. Prefix condition $\Longrightarrow$ no codeword is an ancestor of any other codeword $\Longrightarrow$each codeword eliminates its descendants as possible codewords. Let $\ell_{\max}$ be the length of the longest codeword and consider all the nodes at level $\ell_{\max}$; these can be codewords, descendants, and neither (unused). A codeword at level $\ell_i$ has $D^{\ell_{\max}-\ell_i}$ descendants

at level $\ell_{\max}$. Each of these descendant sets must be disjoint. The total number of nodes in sets must be $\leq D^{\ell_{\max}}$. Summing over all codewords

$$\sum D^{\ell_{\max}-\ell_i} \leq D^{\ell_{\max}} \iff \sum D^{-\ell_i} \leq 1.$$

**Sufficiency.** Conversely, given any set of codeword lengths $\ell_1, \ell_2, \ldots, \ell_m$ that satisfy the Kraft inequality, we can construct a tree like the one in Figure 2. Label the first node (lexicographically) of depth $\ell_1$ as codeword 1, and remove its descendants from the tree. Then label the first remaining node of depth $\ell_2$ as codeword 2, and so on. Proceeding in this way, we construct prefix code with the specified $\ell_1, \ell_2, \ldots, \ell_m$.

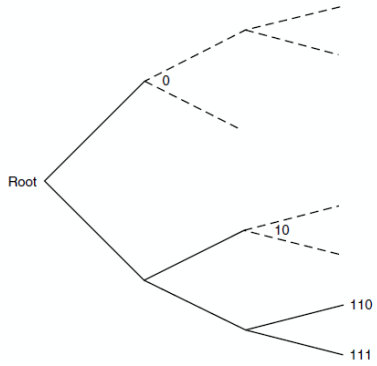
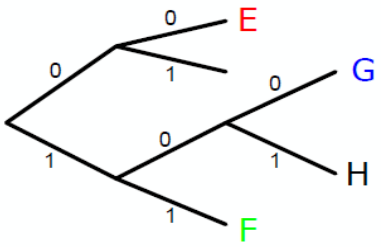



Figure 2: Code tree for the Kraft inequality ($D = 2$)

## 2.2 Kraft inequality - infinite case

### Kraft inequality - infinite case

**Theorem 13** (Extended Kraft inequality)**.** *For any countably infinite set of codewords that form a prefix code, the codeword lengths satisfy the extended Kraft inequal-*



*ity,*

$$\sum_{i=1}^{\infty} D^{-\ell_i} \leq 1. \tag{5}$$

*Conversely, given any $\ell_1, \ell_2, \ldots$ satisfying the extended Kraft inequality, we can construct a prefix code with these codeword lengths.*

*Proof.* $D = \{0, 1, \ldots, D-1\}$; the $i$th codeword $y_1 y_2 \cdots y_{\ell_i}$ and

$$0.y_1 y_2 \cdots y_{\ell_i} = \sum_{j=1}^{\ell_i} y_j D^{-j}. \tag{6}$$

This codeword corresponds to the interval

$$\left[0.y_1 y_2 \cdots y_{\ell_i}, 0.y_1 y_2 \cdots y_{\ell_i} + \frac{1}{D^{\ell_i}}\right) \subset [0,1],$$

the set of all real numbers whose $D$-ary expansion begins with $0.y_1 y_2 \cdots y_{\ell_i}$. These intervals are disjoint, due to the prefix condition, so the sum of their lengths $\leq 1$, that is

$$\sum_{i=1}^{\infty} D^{-\ell_i} \leq 1.$$

□

*Proof - continuation.* Conversely, if the lengths $\ell_1, \ell_2, \ldots$ satisfy the Kraft inequality we reorder the indexing such that $\ell_1 \leq \ell_2 \leq \ldots$. Then assign the intervals in order from the low end of the unit interval. For example if we wish to construct a binary code with $\ell_1 = 1$, $\ell_2 = 2$, ..., we assign the intervals $\left[0, \frac{1}{2}\right)$, $\left[\frac{1}{2}, \frac{1}{4}\right), \ldots$ to the symbols, with the corresponding codewords $0, 10, \ldots$ □

# 3 Optimal codes

## 3.1 Construction of optimal codes

**Construction of optimal codes**

We look for the prefix code with the minimum expected length.

Minimize

$$L = \sum_{i=1}^{|\mathcal{X}|} p_i \ell_i \tag{7}$$

over all integers $\ell_1, \ell_2, \ldots, \ell_m$ satisfying

$$\sum D^{-\ell_i} \leq 1. \tag{8}$$

We ignore the condition "$\ell_i$ integer", assume equality in (8) and use the Lagrange multiplication method. Find

$$\min J = \sum p_i \ell_i + \lambda \left( \sum D^{-\ell_i} \right) \tag{9}$$

Differentiating w.r.t. $\ell_i$, we obtain

$$\frac{\partial J}{\partial \ell_i} = p_i - \lambda D^{-\ell i} \ln D.$$

Equating to 0

$$D^{-\ell_i} = \frac{p_i}{\lambda \ln D}.$$

Substituting this in (8) we find $\lambda = 1/\ln D$ and $p_i = D^{-\ell_i}$, yielding optimal code lengths,

$$\ell_i^* = -\log_D p_i. \tag{10}$$

This *noninteger codeword lengths* yield expected codeword length

$$L^* = \sum p_i \ell_i^* = -\sum p_i \log_D p_i = H_D(X). \tag{11}$$

Rather than demonstrate that $\ell_i^* = -\log_D \ell_i$ is a global minimum we verify optimality directly in the proof of the following theorem.

**Theorem 14.** *The expected length L of any instantaneous D-ary code for a random variable X is greater than or equal to the entropy $H_D(X)$; that is,*

$$L \geq H_D(X), \tag{12}$$

*with equality iff $D^{-\ell_i} = p_i$.*

*Proof.*

$$\begin{aligned} L - H_D(X) &= \sum p_i \ell_i - \sum p_i \log_D \frac{1}{p_i} \\ &= -\sum p_i \log_D D^{-\ell_i} + \sum p_i \log p_i. \end{aligned}$$

Letting $r_i = D^{-\ell_i} / \sum_j D^{-\ell_j}$ and $c = \sum D^{-\ell_i}$, we obtain

$$\begin{aligned} L - H_D(X) &= \sum p_i \log_D \frac{p_i}{r_i} - \log_D c \\ &= D(p \parallel r) + \log_D \frac{1}{c} \geq 0 \end{aligned}$$

by the nonnegativity of relative entropy and Kraft inequality ($c \leq 1$). Hence $L \geq H_D(X)$, with equality iff $p_i = D^{-\ell_i}$ (i.e. iff $-\log_D p_i \in \mathbb{N}, \forall i$). □

**Definition 15.** A probability distribution is called $D$-adic if each of the probabilities is equal to $D^{-n}$ for some $n$.

Thus, we have equality in the theorem iff the distribution of $X$ is $D$-adic.

The preceding proof also indicates a procedure for finding an optimal code: Find the $D$-adic distribution that is closest (in the relative entropy sense) to the distribution of $X$. This distribution provides the set of codeword lengths. Construct the code by choosing the first available node as in the proof of the Kraft inequality. We then have an optimal code for $X$. However, this procedure is not easy, since the search for the closest $D$-adic distribution is not obvious.

## 3.2 Bounds on the optimal code length

**Bounds on the optimal code length**

Since $\log_D \frac{1}{p_i}$ may not equal an integer; we round up

$$\ell_i = \left\lceil \log_D \frac{1}{p_i} \right\rceil$$

These lengths satisfy the Kraft inequality

$$\sum D^{-\left\lceil \log_D \frac{1}{p_i} \right\rceil} \leq \sum D^{-\log_D \frac{1}{p_i}} = \sum p_i = 1$$

But

$$\log_D \frac{1}{p_i} \leq \ell_i < \log_D \frac{1}{p_i} + 1$$

Multiplying by $p_i$ and summing, we obtain

$$H_D(X) \leq L < H_D(X) + 1 \tag{13}$$

An optimal code can do better than this code!

**Theorem 16.** *Let $\ell_1^*, \ell_2^*, \ldots, \ell_m^*$ be optimal codeword lengths for a source distribution $p$ and a $D$-ary alphabet, and let $L^*$ be the associated expected length of an optimal code ($L^* = \sum p_i \ell_i^*$). Then*

$$H_D(X) \leq L^* < H_D(X) + 1. \tag{14}$$

*Proof.* Let $\ell_i = \left\lceil \log_D \frac{1}{p_i} \right\rceil$. Then $\ell_i$ satisfy the Kraft inequality and from (13)

$$H_D(X) \leq L = \sum p_i \ell_i < H_D(X) + 1$$

$L^* \leq L$, since the code is optimal and $L^* \geq H_D(X)$ from Theorem 14, we have the conclusion. □

- *Assume $D = 2$!*
- Overhead maximum 1bit.
- We can do better by encoding blocks of $n$ symbols – a block is a "super-symbol" from the alphabet $\mathcal{X}^n$

- Define $L_n$, the expected codeword length per input symbol

$$L_n = \frac{1}{n} \sum p(x_1, \ldots, x_n) \ell(x_1, \ldots, x_n) = \frac{1}{n} E\left(\ell(X_1, \ldots, X_n)\right) \tag{15}$$

- Bounds

$$H(X_1, \ldots, X_n) \leq E\left(\ell(X_1, \ldots, X_n)\right) < H(X_1, \ldots, X_n) + 1 \tag{16}$$

- $X_1, \ldots, X_n$ i.i.d $\Rightarrow H(X_1, \ldots, X_n) = \sum H(X_i) = nH(X) \Rightarrow$

$$H(X) \leq L_n < H(X) + \frac{1}{n}$$

- Using large block lengths we can achieve an expected codelength per symbol arbitrarily close to the entropy.
- For a stochastic process with $(X_i)$ not necessarily i.i.d:

**Theorem 17.** *The minimum expected codeword length per symbol satisfies*

$$\frac{H(X_1, \ldots, X_n)}{n} \leq L_n^* < \frac{H(X_1, \ldots, X_n)}{n} + \frac{1}{n}. \tag{17}$$

*Moreover, if $X_1, \ldots, X_n$ is a stationary stochastic process,*

$$L_n^* \to H\left(\mathcal{X}\right), \tag{18}$$

*where $H\left(\mathcal{X}\right)$ is the entropy rate of the process.*

*Proof.* In this case we still have the bound (16). Dividing by $n$ and defining $L_n$ to be the expected description length per symbol, we obtain

$$\frac{H(X_1, \ldots, X_n)}{n} \leq L_n < \frac{H(X_1, \ldots, X_n)}{n} + \frac{1}{n}.$$

If the stochastic process is stationary, then $\frac{H(X_1, \ldots, X_n)}{n} \to H\left(\mathcal{X}\right)$, as $n \to \infty$. □

If the code is designed for the wrong distribution $q(x)$ (for example the wrong distribution may be the best estimate of the unknown distribution that we can make):

**Theorem 18** (Wrong code)**.** *The expected length under $p(x)$ of the code assignment $\ell(x) = \left\lceil \log \frac{1}{q(x)} \right\rceil$ satisfies*

$$H(p) + D(p \parallel q) \leq E_p(\ell(X)) < H(p) + D(p \parallel q) + 1. \tag{19}$$

*Proof.* The expected codelength is

$$\begin{aligned} E(\ell(X) &= \sum_x p(x) \left\lceil \log \frac{1}{q(x)} \right\rceil < \sum_x p(x) \left( \log \frac{1}{q(x)} + 1 \right) \\ &= \sum_x p(x) \left( \log \frac{p(x)}{q(x)} \frac{1}{p(x)} + 1 \right) \\ &= \sum_x p(x) \log \frac{p(x)}{q(x)} + \sum_x p(x) \log \frac{1}{p(x)} + 1 \\ &= D(p \parallel q) + H(p) + 1. \end{aligned}$$

Lower bound similarly. □

Thus, believing that the distribution is $q(x)$ when the true distribution is $p(x)$ incurs a penalty of $D(p||q)$ in the average description length.

# 4 Kraft Inequality for Uniquely Decodable Codes

**Kraft Inequality for Uniquely Decodable Codes**

The class of uniquely decodable code is larger than the class of instantaneous code. Can we achieve a lower expected codeword length for this class? *No!*

**Theorem 19** (McMillan)**.** *The codeword lengths of any uniquely decodable D-ary code must satisfy the Kraft inequality*

$$\sum D^{-\ell_i} \leq 1. \tag{20}$$

*Conversely, given a set of codeword lengths that satisfy this inequality, it is possible to construct a uniquely decodable code with these codeword lengths.*

*Proof.*

$$\ell(x_1, \ldots, x_k) = \sum_{i=1}^{k} \ell(x_i). \tag{21}$$

$$\sum_{x \in \mathcal{X}} D^{-\ell(x)} \leq 1? \tag{22}$$

Trick: consider the $k$th power of lhs of (22)

$$\begin{aligned} \left( \sum_{x \in \mathcal{X}} D^{-\ell(x)} \right)^k &= \sum_{x_1 \in \mathcal{X}} \cdots \sum_{x_1 \in \mathcal{X}} D^{-\ell(x_1)} \cdots D^{-\ell(x_k)} \\ &= \sum_{x_1, \ldots, x_k \in \mathcal{X}^k} D^{-\ell(x_1)} \cdots D^{-\ell(x_k)} = \sum_{x^k \in \mathcal{X}^k} D^{-\ell(x^k)} \end{aligned}$$

□

*Proof - continuation.* Last relation by (21). Gathering terms by word lengths

$$\sum_{x^k \in \mathcal{X}^k} D^{-\ell(x^k)} = \sum_{m=1}^{k\ell_{\max}} a(m) D^{-m},$$

where $\ell_{\max}$ is the maximum codeword length and $a(m)$ is the number of source sequences $x^k$ mapping into codewords of length $m$, $a(m) \leq D^m$, and we have

$$\left( \sum_{x \in \mathcal{X}} D^{-\ell(x)} \right)^k \leq \sum_{m=1}^{k\ell_{\max}} D^m D^{-m} = k\ell_{\max}. \tag{23}$$

□

*Proof - continuation.* (23) is equivalent to

$$\sum_j D^{-\ell_j} \leq (k\ell_{\max})^{1/k}. \tag{24}$$

Since $\lim_{k\to\infty} (k\ell_{\max})^{1/k} = 1$, we have

$$\sum_j D^{-\ell_j} \leq 1.$$

Conversely, given any set of $\ell_1, \ell_2, \ldots, \ell_m$ satisfying the Kraft inequality, we can construct an instantaneous code as proved in Section on Kraft inequality. Since every instantaneous code is uniquely decodable, we have also constructed a uniquely decodable code. □

**Corollary 20.** *A uniquely decodable code for an infinite source alphabet $\mathcal{X}$ also satisfies the Kraft inequality.*

*Proof.* For infinite $|\mathcal{X}|$ the preceding proof crashes at (24). Fixing: any subset of uniquely decodable code is uniquely decodable; any finite subset satisfies Kraft inequality, hence

$$\sum_{i=1}^{\infty} D^{-\ell_i} = \lim_{N\to\infty} \sum_{i=1}^{N} D^{-\ell_i} \leq 1.$$

Conversely, given any set of $\ell_1, \ell_2, \ldots$ satisfying the Kraft inequality, we can construct an instantaneous code as proved in Section on Kraft inequality. Since every instantaneous code is uniquely decodable, we have also constructed a uniquely decodable code with an infinite number of codewords. Hence the McMillan inequality also applies for infinite alphabets. □

# 5 Huffman Codes

## 5.1 Huffman codes

**Huffman codes**

Huffman gave in [1] an algorithm for the construction of an optimal code

- An optimal binary instantaneous code must satisfy:
  1. $p(x_i) > p(x_j) \Rightarrow \ell(x_i) \leq \ell(x_j)$ (else swap codewords)
  2. The two longest codewords have the same length (else chop a bit off the longer codeword)
  3. $\exists$ two longest codewords differing only in the last bit (else chop a bit off all of them)
- Huffman Code construction
  1. Take the two smallest $p(x_i)$ and assign each a different last bit. Then merge into a single symbol.
  2. Repeat step 1 until only one symbol remains
- Used in JPEG, MP3, ...

**Examples**

*Example* 21.

$$X \sim \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0.25 & 0.25 & 0.2 & 0.15 & 0.15 \end{pmatrix}$$

We can combine the symbols 4 and 5 into a single source symbol, with a probability assignment 0.30. Proceeding this way, combining the two least likely symbols into one symbol until we are finally left with only one symbol, and then assigning codewords to the symbols, we obtain the following table:

| Codeword Length | Codeword | $X$ | Probability |
|---|---|---|---|
| 2 | 01 | 1 | 0.25 0.3 0.45 0.55 1 |
| 2 | 10 | 2 | 0.25 0.25 0.3 0.45 |
| 2 | 11 | 3 | 0.2 0.25 0.25 |
| 3 | 000 | 4 | 0.15 0.2 |
| 3 | 001 | 5 | 0.15 |

This code has average length $L = 2.3$ bits and $H(X) = 2.286$.

For $D$-ary code, first add extra zero-probability (dummy) symbols until $|X| - 1$ is a multiple of $D - 1$ and then group $D$ symbols at a time.

## 5.2 Optimality of Huffman codes

### Optimality of Huffman codes

- We prove by induction that the binary Huffman code is optimal.
- There are many optimal codes: inverting all the bits or exchanging two codewords of the same length will give another optimal code.
- The Huffman procedure constructs one such optimal code.
- Assume w.l.o.g. that $p_1 \geq p_2 \geq \cdots \geq p_m$.
- A code is optimal if $\sum p_i \ell_i$ is minimal.

**Lemma 22.** *For any distribution, there exists an optimal instantaneous code (with minimum expected length) that satisfies the following properties:*

1. *The lengths are ordered inversely with the probabilities (i.e., if $p_j > p_k$, then $\ell_j \leq \ell_k$).*
2. *The two longest codewords have the same length.*
3. *Two of the longest codewords differ only in the last bit and correspond to the two least likely symbols.*

**Proof**. The proof amounts to swapping, trimming, and rearranging, as shown in Figure 3.

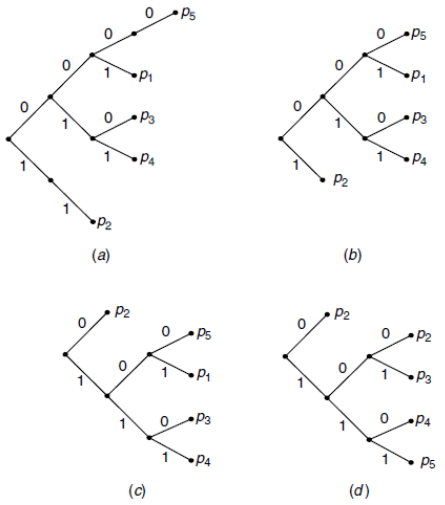



Figure 3: Properties of optimal codes.

**Optimality of Huffman codes - proof**

- We assume that $p_1 \geq p_2 \geq \cdots \geq p_m$.
- A possible instantaneous code is given in (a).
- By trimming branches without siblings, we improve the code to (b).
- We now rearrange the tree as shown in (c), so that the word lengths are ordered by increasing length from top to bottom.
- Finally, we swap probability assignments to improve the expected depth of the tree, as shown in (d).
- Every optimal code can be rearranged and swapped into canonical form as in (d), where $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_m$ and $\ell_{m-1} = \ell_m$, and the last two codewords differ only in the last bit.

Consider an optimal code $C_m$:

- *If $p_j > p_k$, then $\ell_j \leq \ell_k$.* (Swap) $C'_m$ is $C_m$ with $j \leftrightarrow k$

$$\begin{aligned} L(C'_m) - L(C_m) &= \sum p_i \ell'_i - \sum p_i \ell_i \\ &= p_j \ell_k + p_k \ell_j - p_j \ell_j - p_k \ell_k \\ &= (p_j - p_k)(\ell_k - \ell_j) \end{aligned}$$

  But $p_j - p_k > 0$, and since $C_m$ is optimal, $L(C'_m) - L(C_m) \geq 0$. Hence, we must have $\ell_k \geq \ell_j$ . Thus, $C_m$ itself satisfies property 1.

- *The two longest codewords are of the same length.* (Trim) Otherwise, one can delete the last bit of the longer one, preserving the prefix property and achieving lower expected codeword length. By property 1, the longest codewords must belong to the least probable source symbols.
- *The two longest codewords differ only in the last bit and correspond to the two least likely symbols.* Not all optimal codes satisfy this property, but by rearranging, we can find an optimal code that does. If there is a maximal-length codeword without a sibling, we can delete the last bit of the codeword and still satisfy the prefix property. This reduces the average codeword length and contradicts the optimality of the code. Hence, every maximal-length codeword in any optimal code has a sibling. Now we can exchange the longest codewords so that the two lowest-probability source symbols are associated with two siblings on the tree. This does not change the expected length, $\sum p_i \ell_i$ . Thus, the codewords for the two lowest-probability source symbols have maximal length and agree in all but the last bit. □

Thus, we have shown that there exists an optimal code satisfying the properties of the lemma. We call such codes *canonical codes*.

**Theorem 23.** *Huffman coding is optimal; that is, if $C^*$ is a Huffman code and $C$ is any other uniquely decodable code, $L(C^*) \leq L(C)$.*

**Proof**.

- For any probability mass function for an alphabet of size $m$, $\mathbf{p} = (p_1, p_2, ..., p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \ldots, p_{m-2}, p_{m-1} + p_m)$ over an alphabet of size $m - 1$(Figure 4). Let $C^*_{m-1}(\mathbf{p}')$ be an optimal code for $\mathbf{p}'$ , and let $C^*_m(\mathbf{p})$ be the canonical optimal code for $\mathbf{p}$.
- Induction - two steps
  1. expand an optimal code for $\mathbf{p}'$ to construct a code for $\mathbf{p}$;
  2. condense an optimal canonical code for $\mathbf{p}$ to construct a code for the Huffman reduction $\mathbf{p}'$.
- Comparing the average codeword lengths for the two codes establishes that the optimal code for $\mathbf{p}$ can be obtained by extending the optimal code for $\mathbf{p}'$.
- From $\mathbf{p}'$ we construct an extension code for $m$ elements: take the codeword in $C^*_{m-1}$ to weight $p_{m-1} + p_m$ and extend it by adding a 0 to form a codeword for symbol $m - 1$ and adding 1 to form the codeword for symbol $m$:

$$\begin{array}{cccccc} & C^*_{m-1}(\mathbf{p}') & & C^*_m(\mathbf{p}) & & \\ p_1 & w'_1 & \ell'_1 & w_1 = w'_1 & \ell_1 = \ell'_1 & \\ p_2 & w'_2 & \ell'_2 & w_2 = w'_2 & \ell_2 = \ell'_2 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ p_{m-2} & w'_{m-2} & \ell'_{m-2} & w_{m-2} = w'_{m-2} & \ell_{m-2} = \ell'_{m-2} & \\ p_{m-1} + p_m & w'_{m-1} & \ell'_{m-1} & w_{m-1} = w'_{m-1} & \ell'_{m-1} = \ell'_{m-1} & \\ & & & w_m = w'_{m-1} & \ell_m \ = \ell'_{m-1} & \end{array} \tag{25}$$

- Calculation of average length $\sum_i p'_i \ell'_i$

$$L(\mathbf{p}) = L^*(\mathbf{p}') + p_{m-1} + p_m. \tag{26}$$

- Similarly, from $\mathbf{p}$, we construct a code for $\mathbf{p}'$ by merging the codewords for the two lowest-probability symbols $m - 1$ and $m$ with the probabilities $p_{m-1}$ şi $p_m$, which are siblings by the properties of canonical code. The new code for $\mathbf{p}'$ has average length

$$\begin{aligned} L(\mathbf{p}') &= \sum_{i=1}^{m-2} p_i \ell_i + p_{m-1} (\ell_{m-1} - 1) + p_m (\ell_m - 1) \\ &= \sum_{i=1}^{m} p_i \ell_i - p_{m-1} - p_m \\ &= L^*(\mathbf{p}) - p_{m-1} - p_m. \end{aligned} \tag{27}$$

- (26)+(27)⇒
$$L(\mathbf{p}') + L(\mathbf{p}) = L^*(\mathbf{p}') + L^*(\mathbf{p}) \tag{28}$$
or
$$\big(L(\mathbf{p}') - L^*(\mathbf{p}')\big) + (L(\mathbf{p}) - L^*(\mathbf{p})) = 0. \tag{29}$$
- The expressions in paranthesis are nonnegative, so $L(\mathbf{p}) = L^*(\mathbf{p})$.□

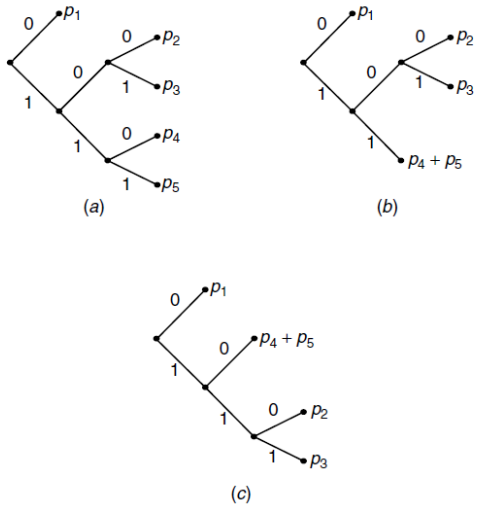



Figure 4: Induction step for Huffman coding. A canonical optimal code is illustrated in (a). Combining the two lowest probabilities → (b). Rearranging the probabilities in decreasing order → (c) for $m - 1$ symbols.

# 6 Shannon-Fano-Elias Coding

## 6.1 The code

**Shannon-Fano-Elias Coding**

- $\mathcal{X} = \{1, 2, \ldots, m\}$; $p(x) > 0, \forall x$
- cdf - *cumulative distribution function*
$$F(x) = \sum_{a \le x} p(a). \tag{30}$$

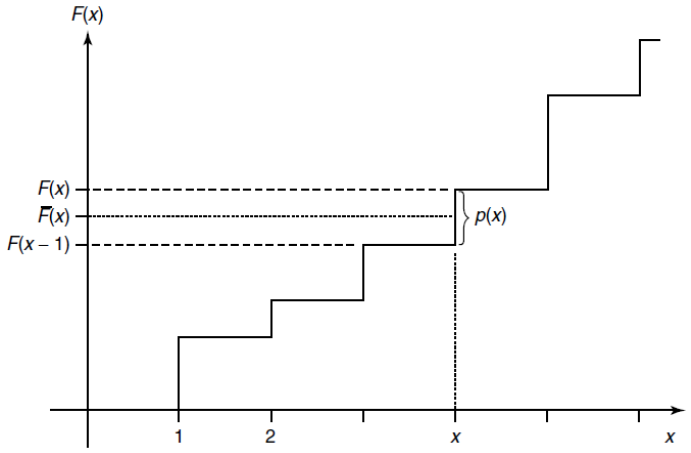



Figure 5: cdf and Shannon-Fano-Elias coding

- the *modified cumulative distribution function*

$$\overline{F}(x) = \sum_{a \leq x} p(a) + \frac{1}{2} p(x). \tag{31}$$

(see Figure 5)

- $a \neq b \Rightarrow \overline{F}(a) \neq \overline{F}(b)$; we can determine $x$ if we know $\overline{F}(x)$, thus $\overline{F}(x)$ *can be used as a code for* $x$.
- $\overline{F}(x) \in \mathbb{R} \Rightarrow \overline{F}(x)$ has an infinite number of bits
- use $\left\lfloor \overline{F}(x) \right\rfloor_{\ell(x)}$ is $\overline{F}(x)$ truncated at $\ell(x)$ bits

$$\overline{F}(x) - \left\lfloor \overline{F}(x) \right\rfloor_{\ell(x)} < \frac{1}{2^{\ell(x)}}. \tag{32}$$

$$\ell(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1 \Longrightarrow \frac{1}{2^{\ell(x)}} < \frac{p(x)}{2} = \overline{F}(x) - F(x-1) \tag{33}$$

- $\left\lfloor \overline{F}(x) \right\rfloor_{\ell(x)}$ lies within the step corresponding to $x$
- The code is prefix-free!
- codeword $z_1 z_2 \ldots z_l \mapsto \left[0.z_1 z_2 \ldots z_l, 0.z_1 z_2 \ldots z_l + \frac{1}{2^\ell}\right)$; the code is prefix-free iff the intervals are disjoint
- interval length = $2^{-\ell(x)} <$ half of the step in (33) $\Longrightarrow$ interval are disjoint

- Since $\ell(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$, the expected length of the code is

$$L = \sum_x p(x)\ell(x) = \sum_x p(x) \left( \left\lceil \log \frac{1}{p(x)} \right\rceil + 1 \right) < H(X) + 2. \tag{34}$$

- The procedure does not require ordered probabilities

*Example* 24. We consider an example where all the probabilities are dyadic.

| $x$ | $p(x)$ | $F(x)$ | $\overline{F}(x)$ | $\overline{F}(x)_{[2]}$ | $\left\lceil \log \frac{1}{p(x)} \right\rceil + 1$ | 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 |

$L = 2.75$ bits, $H(X) = 1.75$

*Example* 25. The distribution is not dyadic, the binary representation is infinite.

| $x$ | $p(x)$ | $F(x)$ | $\overline{F}(x)$ | $\overline{F}(x)_{[2]}$ | $\left\lceil \log \frac{1}{p(x)} \right\rceil + 1$ | Codeword |
|---|---|---|---|---|---|---|
| 1 | 0.25 | 0.25 | 0.125 | 0.001 | 3 | 001 |
| 2 | 0.25 | 0.5 | 0.375 | 0.011 | 2 | 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 | $0.110\overline{0011}$ | 4 | 1110 |

## 6.2 Competitive optimality of the Shannon code

**Competitive optimality of the Shannon code**

- Huffman coding is optimal - it has minimum expected length.
- For particular sequencess Huffman code can be worse than other codes.
- Formalization - zero-sum game: two people are given a probability distribution and are asked to design an instantaneous code for the distribution. Then a source symbol is drawn the payoff to player A is 1, -1 or 0 (code shorter, longer or tie)
- Since analysis for Huffman code is difficult, we consider Shannon code with codeword lengths $\ell(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil$.

**Theorem 26.** *Let $\ell(x)$ be the codeword lengths of Shannon code and $\ell'(x)$ be the codeword lengths associated with any other uniquely decodable code. Then*

$$P(\ell(X) \geq \ell'(X) + c) \leq \frac{1}{2^{c-1}}. \tag{35}$$

*Proof.*

$$\begin{aligned}
P\left(\ell(X) \geq \ell'(X) + c\right) &= P\left(\left\lceil \log \frac{1}{p(X)} \right\rceil \geq \ell'(X) + c\right) \\
&\leq P\left(\log \frac{1}{p(X)} \geq \ell'(X) + c - 1\right) \\
&= P\left(p(X) \leq 2^{-\ell'(X)-c+1}\right) \\
&= \sum_{x:p(x)\leq 2^{-\ell'(X)-c+1}} p(x) \leq \sum_{x:p(x)\leq 2^{-\ell'(X)-c+1}} 2^{-\ell'(X)-(c-1)} \\
&\leq \sum_x 2^{-\ell'(X)} 2^{-(c-1)} \qquad \text{Kraft inequality} \\
&\leq 2^{-(c-1)}
\end{aligned}$$

□

Stronger result if $p(x)$ is dyadic:

**Theorem 27.** *For a dyadic pmf $p(x)$, if $\ell(x) = \log \frac{1}{p(x)}$ is the length of the binary Shannon code and $\ell'(x)$ is the length of any other uniquely decodable binary code, then*

$$P\left(\ell(X) < \ell'(X)\right) \geq P\left(\ell(X) > \ell'(X)\right), \tag{36}$$

*with equality iff $\ell'(x) = \ell(x)$ for all $x$. Thus, the code length assignment $\ell(x) = \log \frac{1}{p(x)}$ is uniquely competitively optimal.*

**Proof.** Consider

$$sgn(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \\ -1 & \text{if } t < 0 \end{cases}$$

It can be shown graphically that

$$sgn(t) \leq 2^t - 1, \forall t \in \mathbb{Z}. \tag{37}$$

$$\begin{aligned}
&P\left(\ell'(X) < \ell(X)\right) - P\left(\ell'(X) > \ell(X)\right) = \sum_{x:\ell'(x)<\ell(x)} p(x) - \sum_{x:\ell'(x)>\ell(x)} p(x) \\
&= \sum_x p(x) sgn\left(\ell(x) - \ell'(x)\right) = E\left(sgn\left(\ell(X) - \ell'(X)\right)\right) \\
&\leq \sum_x p(x)\left(2^{\ell(x)-\ell'(x)} - 1\right) \qquad \text{bound on } sgn \\
&= \sum_x 2^{-\ell(x)}\left(2^{\ell(x)-\ell'(x)} - 1\right) = \sum_x 2^{-\ell'(x)} - \sum_x 2^{-\ell(x)} \\
&\leq \sum_x 2^{-\ell'(x)} - 1 \qquad \text{Kraft inequality} \\
&\leq 1 - 1 = 0.
\end{aligned}$$

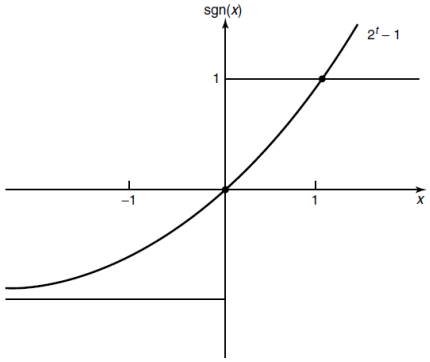



Figure 6: *sgn* function and bound

We have equality in the above chain if $t = 0$ or $t = 1$ in $sgn(t)$ (i.e. $\ell(x) = \ell'(x)$ or $\ell(x) = \ell'(x) + 1$) and $\ell'(x)$ satisfies Kraft inequality with equality, that is $\ell'(x)$ is length for an optimal code (i.e. $\ell'(x) = \ell(x)$ for all $x$).□

**Corollary 28.** *For nondyadic probability mass functions,*

$$E\left(sgn\left(\ell(X) - \ell'(X) - 1\right)\right) \leq 0, \tag{38}$$

*where* $\ell(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil$ *and* $\ell'(x)$ *is any other code for the source.*

*Proof.* Along the same lines as the preceding proof. □

Hence Shannon coding is optimal under a large variety of criteria; it is robust with respect to the payoff function. In particular, for dyadic $p$, $E(\ell - \ell') \leq 0$, $E\left(sgn(\ell - \ell')\right) \leq 0$, and by use of inequality (37), $E\left(f(\ell - \ell')\right)$ for any function $f$ satisfying $f(t) \leq 2^t - 1$, $t = 0, \pm 1, \pm 2, \ldots$.

**References**

# References

[1] Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, 2nd edition, Wiley, 2006.

[2] David J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[3] Robert M. Gray, *Entropy and Information Theory*, Springer, 2009

Figure 7: David A. Huffman (1925–1999)

Figure 8: Robert Fano (1917–)

# References

[1] D. A. Huffman, A method for the construction of minimum redundancy codes, Proc. IRE, 40: 1098–1101,1952