I232 Information Theory
Chapter 4: Source Coding for Single Sources

Brian Kurkoski

Japan Advanced Institute of Science and Technology

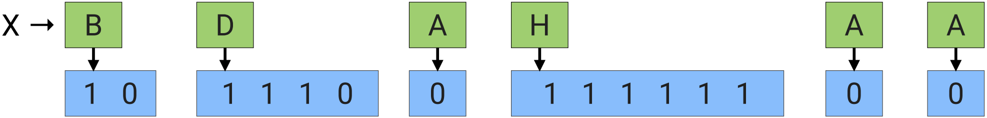
2023 April

Overview of Chapters 4–6 Source Coding

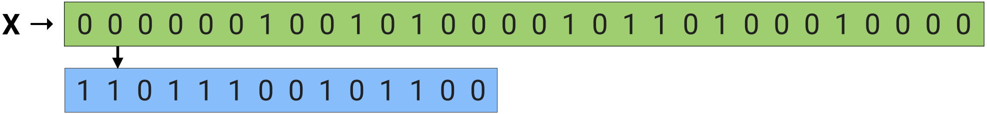
"Source coding" means data compression.

- **Lecture 4: Source Coding for Single Source** Source is a single letter. Example: horse race
- **Lecture 5: Source Coding for Vector Sources** Source is a vector of symbols. Example: binary vector with more zeros than ones
- **Lecture 6: Markov Chains and Entropy Rate** Source is a vector from a Markov source. Example: a zero will be followed by a zero, with high probability.

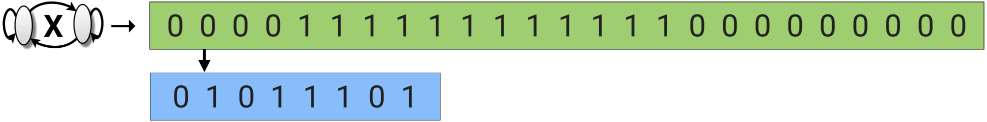
Chapter 4 — Source coding for a single source X



Chapter 5 — Source coding for a vector source **X**



Chapter 6 — Source coding for a Markov source



Outline

4.1 Source Code Strings

Non-Singular Codes and Uniquely Decodable Codes

Prefix Codes

Codes on Trees

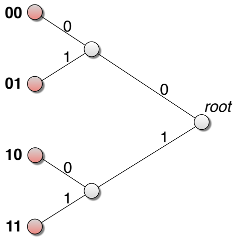
Codes on Trees

Prefix codes can be represented by a code tree

Definition

A *code tree* is a D-ary tree representing a D-ary prefix code. The children of the root node correspond to the first codeword symbol. Successive children correspond to successive codeword symbols. The tree's leaves represent the codewords.

Example The tree for the code 00, 01, 10, 11 is shown at the right.
★1



4.2 Kraft Inequality

Source X produces one symbol from the alphabet $\mathcal{X} = \{1, 2, \ldots, m\}$.

Each symbol x is mapped to a codeword $C(x)$, from a prefix code.

The length of codeword $C(x)$ is $\ell(x)$.

Proposition

Kraft Inequality. For any prefix code over an alphabet of size D, the codeword lengths $\ell(1), \ell(2), \ldots, \ell(m)$ must satisfy the inequality:

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

Conversely, given $\ell(x)$ that satisfy this inequality, there exists a prefix code with these words lengths.

For binary codes $D = 2$.

Visual Proof of Kraft Inequality

(1) Assume m codewords all of length $\ell_{\max}$:

$$\ell(x) = \ell_{\max}$$

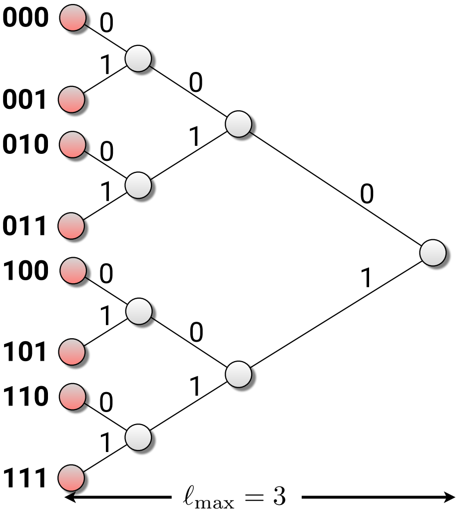
There cannot be more than $2^{\ell_{\max}}$ codewords:

$$m \leq 2^{\ell_{\max}}$$

Then we can show:

$$\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$$

the Kraft inequality for this special case. ★2



Visual Proof of Kraft Inequality

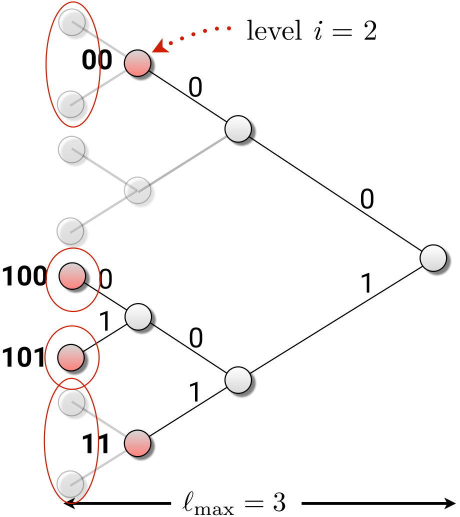
(2) Consider non-equal lengths. A codeword at level i has

$$2^{\ell_{\max}} - 2^{\ell(x)}$$

descendants at level $\ell_{\max}$ (Descendants are not necessarily codewords). Ex.: a level $i = 2$ codeword has 2^{3-2} descendants at level $\ell_{\max}$. The sum of all of these is $\sum_i 2^{\ell_{\max}} - \ell(x)$. We can use this to prove:

$$\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$$

which is the Kraft inequality. ★3 ★Poll



4.3 Huffman Codes

4.3.1 Expected Length of Codes

- Previously we did not consider the distribution of the source X.
- Now, the probability distribution $p_{\mathsf{X}}(x)$ for the source X is known.

Definition

The *expected length* $L(C)$ of a source code C for a random variable X with probability mass function $p_{\mathsf{X}}(x), x \in \mathcal{X}$ is given by:

$$L(C) = \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x)\ell(x)$$

For binary codes, the units of $L(C)$ are bits/source symbol.

4.3.2 Huffman Codes

- Huffman codes are optimal prefix codes for source coding.
- Huffman codes are an important type of entropy coding
- Are constructed using tree, then by labeling the tree to obtain the codewords.
- Construction begins by combining the *two least likely symbols*, to obtain a set of with $m-1$ source symbols.
- Then proceed recursively, until only one symbol with probability 1 remains.

Construction of binary Huffman codes:

1. Input is

$$\mathcal{X} = \{1, 2, \ldots, m\}$$
$$p_1 \geq p_2 \geq \cdots \geq p_{m-1} \geq p_m$$

2. Combine the two least likely symbols:

$$(m-1, m) \rightarrow m'$$
$$p_{m-1} + p_m \rightarrow p'$$

3. Repeat step 2 on:

$$\mathcal{X}' = 1, 2, 3, ..., m-2, m-1 \qquad (|\mathcal{X}'| = |\mathcal{X}| - 1)$$
$$p_1 \geq p_2 \geq \cdots \geq p_{m-2} \geq p_{m-2} \geq p_{m-1}$$

   where p' replaces one of the $p_1, \ldots, p_{m-1}$ values. Repeat until only one node with probability 1.0 remains.

Codebook construction:

4. Label the branch of each node with (0,1)
5. Each codeword is a sequence of labels for that leaf node.

Example: Binary Huffman Codes

- Construct a Huffman code with $p_{\mathsf{X}}(x) = (0.5, 0.3, 0.1, 0.1)$, Example 4.12
- Construct a Huffman code with $p_{\mathsf{X}}(x) = (0.25, 0.25, 0.2, 0.15, 01.15)$, Example 4.13

Properties of Huffman Codes

Some properties of Huffman codes:

- Binary Huffman codes are optimal in the sense of minimizing $L(C)$.
- A Huffman code is not unique. If all the bits in C are inverted, then another optimal code with the same $L(C)$ is obtained. Or, two codewords of the same length can be exchanged without changing $L(C)$.
- The codewords for symbols m and $m-1$ are the same length, and are the longest codewords; other symbols may also have this same longest length.

4.3.3 Non-binary Huffman Codes

D-ary Huffman codes: combine D symbols at each step.

May need to add dummy symbols with 0 probability. Number of dummy symbols to add:

$$(1 - |\mathcal{X}|) \bmod (D - 1).$$

After adding dummy symbols, proceed as in the binary case, but combine D symbols at each step. The last combining step should combine D symbols.

- Construct $D = 3$ Huffman code for $p_{\mathsf{X}}(x) = (0.25, 0.25, 0.15, 0.15, 0.1, 0.1)$
- Construct $D = 3$ Huffman code for $p_{\mathsf{X}}(x) = (0.35, 0.2, 0.15, 0.1, 0.1, 0.1)$

4.4 Bounds on length of optimal source codes

4.4.1 Entropy bound on single-variable compression

Definition

Given a source X distributed as $p_{\mathsf{X}}(x)$, the *base-D entropy* $H_D(\mathsf{X})$ is:

$$H_D(\mathsf{X}) = -\sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) \log_D p_{\mathsf{X}}(x)$$

Definition

A probability distribution is called *D-adic* if each probability is equal to D^{-n} for some non-negative integer n. ★4

Definition

A code C^* with lengths $\ell^*(1), \ell^*(2), \ldots$ and probabilities $p_{\mathsf{X}}(1), p_{\mathsf{X}}(2), \ldots$ is an *optimal code* if:

$$L(C^*) \quad = \quad \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) \ell^*(x) \text{ is minimal.}$$

Entropy bound on single-variable compression

Proposition

Entropy bound on single-variable compression Let $\ell^*(1), \ell^*(2), \dots, \ell^*(m)$ be optimal codewords lengths for source X distributed as $p_{\mathsf{X}}(x)$ and a D-ary alphabet, and let $L(C^*)$ be the expected length. Then $L(C^*)$ satisfies:

$$H_D(\mathsf{X}) \leq L(C^*) \leq H_D(\mathsf{X}) + 1.$$

The lower bounds holds with equality if and only if $p_{\mathsf{X}}(x)$ is D-adic.

- The best possible compression is $H_D(\mathsf{X})$.
- For single source, $H_D(\mathsf{X})$ is achieved only when $p_{\mathsf{X}}(x)$ is D-adic.
- Upper bound $H_D(\mathsf{X}) + 1$: penalty of one bit, for the worst-case $p_{\mathsf{X}}(x)$.

Proof in Two Steps

The proof of the entropy bound on single-variable compression is given in two steps:

Step 1, Section 4.4.2: $H(\mathsf{X}) \leq L(C^*)$

Step 2, Section 4.4.3: $L(C^*) \leq H(\mathsf{X}) + 1$

4.4.2 Proof of lower bound

As preparation for the lower bound proof, make the following definitions:

$$p_i = p_{\mathsf{X}}(i)$$
$$c = \sum_{j \in \mathcal{X}} D^{-\ell_j}$$
$$r_i = \frac{D^{-\ell_i}}{\sum_{j \in \mathcal{X}} D^{-\ell_j}} = \frac{D^{-\ell_i}}{c}$$

Note $c \leq 1$ by the Kraft inequality.

★5

4.4.3 Proof of upper bound

To prove the upper bound:

- Show the existence of a good code C^{good} satisfying $\ell(x) = \lceil -\log p(x) \rceil$
- Uses non-negativity of KL divergence to prove $L(C^{\text{good}}) \leq H_D(\mathsf{X}) + 1$

★6

4.4.4 KL Divergence is the Cost of Miscoding

The KL divergence is a measure of the information lost when q is used to approximate p, as the following cost of miscoding example shows.

- To construct an optimal Huffman source code, we need to know the source distribution p.
- Suppose p is not known, and instead we construct a source code C using another source q.
- Since C is not optimized for the source p, the expected length may increase.
- Interestingly, the KL divergence can be used to describe this increase. $H(p) + D(p||q)$ bits on average are required to describe the random variable following p, when using a code for q.

The KL divergence is the expected number of extra bits required for a source code sampled from p, when using a code designed for q (rather than using a code designed for p).

4.4.5 Lagrange Multipliers Example

See other slide set.

What You Should Have Learned — Source Coding for a Single Source

- How to compress a single random variable, like horse race
- Prefix codes are "instantaneously decodable". Easier to prove results using prefix codes
- Kraft inequality is an upper bound involving codeword lengths
- Huffman coding is both practical and optimal
- Entropy of a source X is the lower bound the expected code length
- In particular, the expected length of an optimal code C^* satisfies:

$$H(X) < L(C^*) < H(X) + 1$$

Class Info

- Tutorial Hours: Monday, April 24 at 13:30. Ask questions about Homework.
- Homework 1 and 2 on LMS. Deadline: Monday, April 24 at 18:00
- Next lecture: Wednesday, April 26. Source Coding for Memoryless Sources. There will be a pop quiz.
- Homework 3 and 4 on LMS. Deadline: Monday, May 1 at 18:00.

And just for fun...

Just for Fun: Eve's Birthday

Eve has just become friends with Michael and David, and they want to know when her birthday is. Eve gives them a list of 10 possible dates:

4 Mar, 5 Mar, 8 Mar

4 Jun, 7 Jun

1 Sep, 5 Sep

1 Dec, 2 Dec, 8 Dec

Eve tells the month to Michael, and the day to David.

Michael said, "I don't know Eve's birthday, but I know that David does not know either."

David said: "At first I did not know Eve's birthday. But now I know"

Michael said: "Now I know Eve's birthday, too"

What is Eve's birthday? How is this possible?