Data Compression = Modeling + Coding

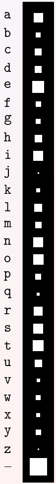
November 3, 2021

uniform codes

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</td></tr>
<tr><td>0001</td></tr>
<tr><td rowspan="2">001</td><td>0010</td></tr>
<tr><td>0011</td></tr>
<tr><td rowspan="4">01</td><td rowspan="2">010</td><td>0100</td></tr>
<tr><td>0101</td></tr>
<tr><td rowspan="2">011</td><td>0110</td></tr>
<tr><td>0111</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">10</td><td rowspan="2">100</td><td>1000</td></tr>
<tr><td>1001</td></tr>
<tr><td rowspan="2">101</td><td>1010</td></tr>
<tr><td>1011</td></tr>
<tr><td rowspan="4">11</td><td rowspan="2">110</td><td>1100</td></tr>
<tr><td>1101</td></tr>
<tr><td rowspan="2">111</td><td>1110</td></tr>
<tr><td>1111</td></tr>
</table>

Symbol codes

i	a_i	p_i
1	a	0.0575
2	b	0.0128
3	c	0.0263
4	d	0.0285
5	e	0.0913
6	f	0.0173
7	g	0.0133
8	h	0.0313
9	i	0.0599
10	j	0.0006
11	k	0.0084
12	l	0.0335
13	m	0.0235
14	n	0.0596
15	o	0.0689
16	p	0.0192
17	q	0.0008
18	r	0.0508
19	s	0.0567
20	t	0.0706
21	u	0.0334
22	v	0.0069
23	w	0.0119
24	x	0.0073
25	y	0.0164
26	z	0.0007
27	–	0.1928



some definitions

Definition

A *code (source code)* C for a random variable X is a mapping from $\{x_1, x_2, \ldots, x_n\}$ to $\mathcal{D}*$the set of finite-length strings of symbols from an alphabet of length D.

Note:
- $C(x)$ the **codeword** of x
- $l(x)$ the length of codeword $C(x)$

Definition

The expectation length $L(C)$ of a source code $C(X)$ for a random variable X with probability mass function $p(x)$is given by:

$$L(C) = \sum p(x_i)l(x_i) = \sum p_i l_i$$

source code examples

Example

ASCII codes
- 96 printable keyboard characters
- log(96) ??

Example

Morse code
- special stop symbol

Definition

A code is called a *prefix code* or *instantaneous code (prefix-free code)* if no codeword is a prefix of any other codeword.

Kraft-McMillan Inequality.

Theorem

For any ***uniquely decodable code*** C

$$\sum_{x \in C} 2^{-l(x)} \leq 1,$$

where $l(x)$ *is the length of the codeword. Also, for any set of lengths* L *such that*

$$\sum_{l \in L} 2^{-l} \leq 1,$$

there is a prefix code C *of the same size such that* $l(x) = l$

Source Coding Theorem

Theorem

There exists a variable-length encoding C of an ensemble X such that the average length of an encoded symbol $L(C, X)$ satify:

$$H(X) \leq \quad L(C, X) \quad < H(X) + 1.$$

are optimal prefix codes

- Generated from a set of probabilities
- David Huffman developed the algorithm as a student in a class on information theory at MIT in 1950
- The algorithm is the most used component of compression algorithms
- used as the back end of GZIP, JPEG

The Huffman algorithm

How it generates the prefix-code tree.

- Start with a forest of trees, one for each message. Each tree contains a single vertex with weight $w_i = p_i$
- Repeat until only a single tree remains
  - Select two trees with the lowest weight roots (w_i and w_2)
  - Combine them into a single tree by adding a new root with weight $w_1 + w_2$, and making the two trees its children.

Arithmetic coding

- information from the messages is combined to share the same bits.
- asymptotically approach the sum of the self information of the individual messages
- The Shannon information content of an outcome

$$h(x \quad = \quad a_i) = log_2(\frac{1}{P(x = a_i)}) = \log_2 \frac{1}{p_i}$$

- Using a Huffman code, each message has to take at least 1 bit

Arithmetic coding

Example

Arithmetic coding

Let $X = \begin{pmatrix} x_1 & \dots & x_n \\ p_1 & \dots & p_n \end{pmatrix}$

The **Arithmetic coding algorithm**

- define the accumulated probability

$$f_i = \sum_{j=1}^{i-1} p(j) \text{ with } i = 1, \dots, n$$

- compute the interval length and size:

$$l_i = \begin{cases} f_i & i = 1 \\ l_{i-1} + f_i * s_{i-1} & 1 < i \leq n \end{cases}$$

$$s_i = \begin{cases} p_i & i = 1 \\ s_{i-1} * p_i & 1 < i \leq n \end{cases}$$