Lecture 8-9 Symbol code

Corresponding to section 5.1-5.5 of the textbook

September 23th and 27th, 2022

Outline

1. Symbol codes
2. What limit is imposed by unique decodability?
3. What's the most compression that we can hope for?
4. How much can we compress?

Review

Theorem (Shannon's source coding theorem)

Let X be an random variable with entropy $H(X) = H$ bits. Given $\epsilon > 0$ and $0 < \delta < 1$, there exists a positive integer N_0 such that for $N > N_0$,

$$|\frac{1}{N} H_\delta(X^N) - H| < \epsilon.$$

In this lecture, we discuss variable-length symbol codes, which encodes one source symbol at a time, instead of encoding huge strings of N source symbols.

In this lecture, we discuss variable-length symbol codes, which encodes one source symbol at a time, instead of encoding huge strings of N source symbols.

These codes are lossless: they are guaranteed to compress and decompress without any errors; but there is a chance that the codes may sometimes produce encoded string longer than the original source string.

The idea is that we can achieve compression, on average, by assigning shorter encodings to the more probable outcomes and longer encodings to the less probable. The key issue are

- What are the implications if a symbol code is losses? If some codewords are shortened, by how much do other codewords have to be lengthened?
- Making compression practical. How can we ensure that a symbol code is easy to decode?
- Optimal symbol codes. How should we assign codelengths to achieve the best achievable compression?

Source coding theorem (symbol code)

There exists a variable-length encoding C of an random variable X such that the average length of an encoded symbol, $L(C,X)$, satisfies $L(C,X) \in [H(X), H(X)+1)$.

Outline

Some notations

- Let $\mathcal{X}^N$ denote the set of ordered N-tuples of elements from the set $\mathcal{X}$, i.e. all strings of length N.
- The symbol $\mathcal{X}^*$ will denote the set of all strings of finite length composed of elements from the set $\mathcal{X}$

Example

$\{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$.

Example

$\{0,1\}^+ = \{0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$.

Symbol code

A symbol code C for a random source $X = \{x_1, x_2, \ldots, x_n\}$ is a mapping form $\mathcal{X}$ to $\mathcal{D} = \{0, 1, \ldots, D-1\}$. $c(x)$ will denote the codeword corresponding to x, and $l(x)$ will denote its length, with $l_i = l(x_i)$.

Symbol code

A symbol code C for a random source $X = \{x_1, x_2, \ldots, x_n\}$ is a mapping form $\mathcal{X}$ to $\mathcal{D} = \{0, 1, \ldots, D-1\}$. $c(x)$ will denote the codeword corresponding to x, and $l(x)$ will denote its length, with $l_i = l(x_i)$.

Extended code

A extended code C^* is a mapping from $\mathcal{X}^*$ to $\{0,1\}^*$ obtained by concatenation, without punctuation, of the corresponding codewords:

$$c^*(x_1 x_2 \cdots x_N) = c(x_1)c(x_2)\cdots c(x_N).$$

Example

A symbol code for the random variable X defined by

$$\begin{aligned} \mathcal{X} &= \{a,b,c,d\} \\ \mathcal{P}_X &= \{1/2, 1/4, 1/8, 1/8\}, \end{aligned}$$

is C_0, shown in the following table.

a_i	$c(a_i)$	l_i
a	1000	4
b	0100	4
c	0010	4
d	0001	4

Using the extended code, we can encode acdbac as

$$c^*(acdbac) = 100000100001010010000010$$

Nonsigular code

A code $C(X)$ is said to be nonsigular if every element of $\mathcal{X}$ maps into a different string in $\mathcal{D}^*$, i.e.,

$$\forall x, y \in \mathcal{X}, x \neq y \Rightarrow c(x) \neq c(y).$$

Uniquely decodable code

Uniquely decodable code

A code $C(X)$ is uniquely decodable if, under the extended code C^*, no two distinct strings have the same encodings, i.e.,

$$\forall \mathbf{x},\mathbf{y} \in \mathcal{X}^*, \mathbf{x} \neq \mathbf{y} \Rightarrow c^*(\mathbf{x}) \neq c^*(\mathbf{y}).$$

So a code is uniquely decodable if its extension is nonsingular.

Prefix code

A symbol code is called a prefix code if no codeword is a prefix of any other codeword.

Example

$C_1 = \{0, 101\}$ *is a prefix code.*

Example

$C_2 = \{1, 101\}$ *is not a prefix code.*

Question

Is C_2 uniquely decodable?

Example

$C_3 = \{0, 10, 110, 111\}$ *is a prefix code.*

Example

$C_4 = \{00, 01, 10, 11\}$ *is a prefix code.*

The expected length

The expected length $L(C, X)$ of a symbol code C for a random variable X is

$$L(C, X) = \sum_{x \in \mathcal{X}} P(x) l(x)$$

We may also write this quantity as

$$L(C, X) = \sum_{i=1}^{l} P_i l_i$$

where $I = |\mathcal{X}|$.

Example

Let

$$\mathcal{X} = \{a,b,c,d\}$$
$$\mathcal{P}_X = \{1/2, 1/4, 1/8, 1/8\},$$

and consider code C_3. *The entropy of* X *is* 1.75 *bits, and the expected length* $L(C_3, X)$ *of this code is also* 1.75 *bits. The sequence of symbols* $\mathbf{x} = (acdbac)$ *is encoded as* $c^*(\mathbf{x}) = 0110111100110$. C_3 *is a prefix code and is therefore uniquely decodeable.*

C_3:

a_i	$c(a_i)$	p_i	$h(p_i)$	l_i
a	0	$1/2$	1.0	1
b	10	$1/4$	2.0	2
c	110	$1/8$	3.0	3
d	111	$1/8$	3.0	3

Example

Consider the fixed length code for the same random variable X, C_4. The expected length $L(C_4, X)$ is 2 bits.

Example

Consider C_5. The expected length $L(C_5, X)$ is 1.25 bits, which is less than X. Bit the code is not uniquely decodeable. The sequence $\mathbf{x} = (acdbac)$ encodes as 000111000, which can also be decoded as $(cabdca)$.

	C_4	C_5
a	00	0
b	01	1
c	10	00
d	11	11

Example

Consider the code C_6. The expected length $L(C_6, X)$ of this code is 1.75 bits. The sequence $\mathbf{x} = (acdbac)$ is encoded as $c^*(\mathbf{x}) = 0011111010011$.

Question

Is C_6 a prefix code? If not, is C_6 uniquely decodeable?

C_6:

a_i	$c(a_i)$	p_i	$h(p_i)$	l_i
a	0	$1/2$	1.0	1
b	01	$1/4$	2.0	2
c	011	$1/8$	3.0	3
d	111	$1/8$	3.0	3

Outline

1. Symbol codes
2. What limit is imposed by unique decodability?
3. What's the most compression that we can hope for?
4. How much can we compress?

Theorem (Kraft inequality)

For any uniquely decodable code $C(X)$ over the binary alphabet $\{0, 1\}$, the codeword lengths must satisfy:

$$\sum_{i=1}^{I} 2^{-l_i} \leq 1,$$

where $I = |\mathcal{X}|$.

Proof.

Define $S = \sum_{i=1}^{I} 2^{-l_i}$. Consider the quantity

$$s^N = [\sum_{i=1}^{I} 2^{-l_i}]^N = \sum_{i_1=1}^{I} \sum_{i_2=1}^{I} cdots \sum_{i_N=1}^{I} 2^{-(l_{i_1}+l_{i_2}+\cdots+l_{i_N})}.$$

The quantity in the exponent, $(l_{i_1} + l_{i_2} + \cdots + l_{i_N})$, is the length of the encoding of the string $\mathbf{x} = a_{i_1} a_{i_2} \cdots a_{i_N}$. For every string $\mathbf{x}$ of length N, there is one term in the above sum. Introduce an array A_l that counts how many strings $\mathbf{x}$ have encoded length l. Then, defining $l_{\min} = \min_i l_i$ and $l_{\max} = \max_i l_i$:

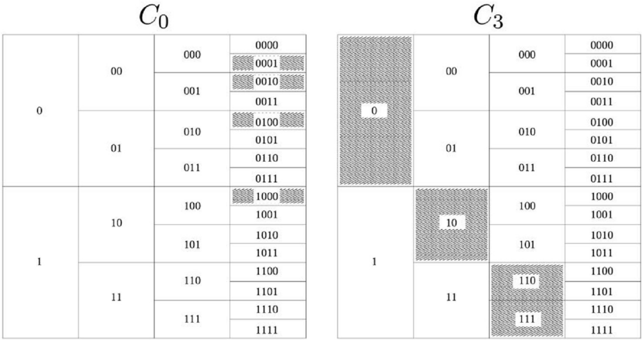
$$S^N = \sum_{l=Nl_{\min}}^{l=Nl_{\max}} 2^{-l} |A_l|$$

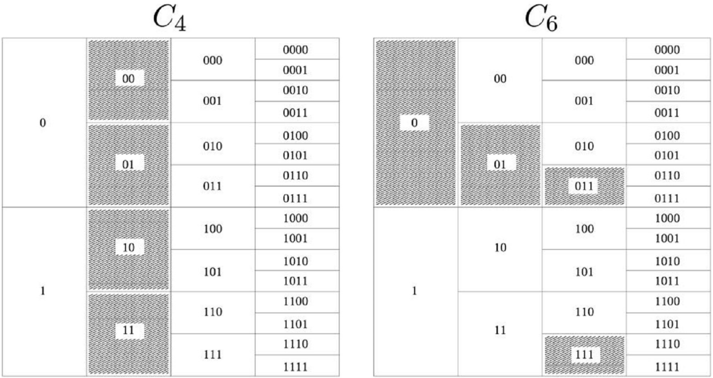
Proof.

Now assume C is uniquely decodable, so that for all $\mathbf{x} \neq \mathbf{y}$, $c(\mathbf{x}) \neq c(\mathbf{y})$. Focus on the set of codes of length l. There are a total of 2^l distinct bit strings of length l, so it must be the case that $A_l \leq 2^l$. So

$$S^N = \sum_{l=Nl_{\min}}^{l=Nl_{\max}} 2^{-l}|A_l| \leq \sum_{l=Nl_{\min}}^{l=Nl_{\max}} 1 \leq Nl_{max}.$$

Thus $S^N \leq l_{\max}N$ for all N. Now if S were greater than 1, then as N increases, S^N would be an exponentially growing function, and for large enough N, an exponential always exceeds a polynomial such as $l_{\max}N$. But our result ($S^N \leq l_{\max}N$) is true for any N. Therefore $S \leq 1$. □


C_0
C_3
0
1
00
01
10
11
000
001
010
011
100
101
110
111
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111


C_4
0
1
00
01
10
11
000
001
010
011
100
101
110
111
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
C_6
0
1
00
01
10
11
000
001
010
011
100
101
110
111
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

Theorem

For any set of codeword lengths $\{l_i\}$ satisfying the Kraft inequality, there is a prefix code having those lengths.

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</td><td rowspan="16">The total symbol code budget</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>

Proof.

We think the codewords as being in a 'codeword supermarket'. with size indicating cost. We imagine purchasing codewords one at a time, starting from the shortest codeword (i.e., the biggest purchases), using the budget shown at the right of the figure in last page.

We start at one side of the codeword supermarket, say the top, and purchase the first codeword of the required length. We advance down the first supermarket a distance 2^{-l}, and purchase the next codeword of the next required length, and so forth. Because the codeword lengths are getting longer, and the corresponding intervals are getting shorter, we can always buy an adjacent codeword to the lastest purchase, so there is no wasting of the budget. Thus at the Ith codeword we have advanced a distance $\sum_{i=1}^{I} 2^{-l_i}$ down the supermarket; if $\sum_i 2^{-l_i} \leq 1$, we will have purchased all the codewords without running out of budget. □

Outline

We wish to minimize the expected length of a code,

$$L(C, X) = \sum_i p_i l_i.$$

Theorem (Lower bound on expected length)

The expected length $L(C, X)$ of a uniquely decodeable code is bounded below by $H(X)$.

Proof.

We define the implicit probabilities $q_i := 2^{-l_i}/z$, where $z = \sum_{i'} 2^{-l_{i'}}$, so that $l_i = \log 1/q_i - \log z$. Then using information inequality, we have

$$\sum_i p_i \log 1/q_i \geq \sum_i p_i \log 1/p_i,$$

with equality if $q_i = p_i$ and the Kraft inequality $z \leq 1$:

$$\begin{aligned} L(C, X) &= \sum_i p_i l_i - \sum_i p_i \log 1/q_i - \log z \\ &\geq \sum_i p_i \log 1/p_i - \log z \\ &\geq H(X). \end{aligned}$$

The equality $L(C, X) = H(X)$ is achieved only if the Kraft equality $z = 1$ is satisfied, and if the codelengths satisfy $l_i = \log(1/p_i)$. □

We can use the same argument for a sequence of symbols from a stochastic process that is not necessarily i.i.d.. In this case, we still have the bound

$$H(X_1, X_2, \cdots, X_n) \leq El(X_1, X_2, \cdots, X_n) < H(X_1, X_2, \cdots, H_n) + 1.$$

Dividing by n again and defining L_n be the expected description length per symbol, we obtain

$$\frac{H(X_1, X_2, \cdots, X_n)}{n} \leq L_n < \frac{H(X_1, X_2, \cdots, H_n)}{n} + \frac{1}{n}.$$

Theorem

The minimum expected codewoed length per symbol satisfies

$$\frac{H(X_1, X_2, \cdot, X_n)}{n} \leq L_n^* < \frac{H(X_1, X_2, \cdots, H_n)}{n} + \frac{1}{n}.$$

Moreover, if $X_1, X_2, \ldots$ is a stationary stochastic process,

$$L_n^* \to H(\mathcal{X}),$$

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

Outline

Theorem (Source coding theorem for symbol codes)

For a random variable X there exists a prefix code C with expected length satisfying

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

Proof.

We set the codelengths to integers slight large than the optimum lengths:

$$l_i = \lceil \log_2(1/p_i) \rceil$$

where $\lceil l^* \rceil$ denotes the smallest integer greater than or equal to l^*. [We are not asserting that the optimal code necessarily uses these lengths, we are simply choosing these lengths because we can use them to prove the theorem.] We check that there is a prefix code with these lengths by confirming that Kraft inequality is satisfied.

$$\sum_i 2^{-l_i} = \sum_i 2^{-\lceil \log(1/p_i) \rceil} \leq \sum_i 2^{-\log(1/p_i)} = \sum_i p_i = 1.$$

Then we confirm

$$L(C, X) = \sum_i p_i \lceil \log(1/p_i) \rceil < \sum_i p_i (\log(1/p_i) + 1) = H(X) + 1. \qquad \square$$

The cost of using the wrong codelengths

If we use a code whose lengths are not equal to the optimal codelengths, the average message length will be larger than the entropy.

The cost of using the wrong codelengths

If we use a code whose lengths are not equal to the optimal codelengths, the average message length will be larger than the entropy.

If the true probabilities are $\{p_i\}$ and we use a complete code with lengths l_i, we can view those lengths as defining implicit probabilities $q_i = 2^{-l_i}$. The average length is

$$L(C, X) = H(X) + \sum_i p_i \log p_i / q_i,$$

i.e., it exceeds the entropy by the relative entropy $D(p \| q)$.