Lecture 7 Source code

Corresponding to section 5.1-5.5 of the textbook

November 12 and 14, 2024

Outline

1. Source 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

1. Source 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?

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

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

Source code

A source 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 $l_i = l(x_i)$.

Source code

A source code C for a random source $X = \{x_1, x_2, \dots, x_n\}$ is a mapping form $\mathcal{X}$ to $\mathcal{D} = \{0, 1, \dots, D-1\}$. $c(x)$ 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.*

Theorem

If the code-words w_i in $C(X)$ all have the same length, then $C(X)$ is uniquely decodable.

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

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

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 $\boldsymbol{x} = (acdbac)$ is encoded as $c^*(\boldsymbol{x}) = 0011111010011$.

Question

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

C_6:

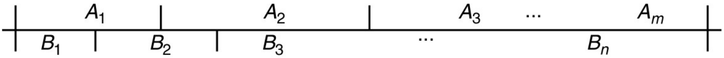
a_i	$c(a_i)$	p_i	$h(p_i)$	l_i
a	0	$1/2$	1.0	1
b	01	$1/4$	2.0	2
c	011	$1/8$	3.0	3
d	111	$1/8$	3.0	3

- We are going to state a necessary and sufficient condition for a code $\mathcal{C}$ to be uniquely decodable.
- We use induction to define a sequence $\mathcal{C}_0$, $\mathcal{C}_1$, $\cdots$ of sets of nonempty words, so $\mathcal{C}_n \subseteq \mathcal{D}^*$ for all n.
- Specifically, we define $C_0 = \mathcal{C}$, and

  $$\mathcal{C}_n = \{w \in \mathcal{D}^* | uw = v \,\text{where}\, u \in \mathcal{C},\ v \in \mathcal{C}_{n-1}\ or\ u \in \mathcal{C}_{n-1}, v \in \mathcal{C}\}$$

  for each $n \geq 1$.
- We then define

  $$\mathcal{C}_\infty = \bigcup_{n=1}^{\infty} \mathcal{C}_n.$$



Reference: A.A. Sardinas and C. W. Patterson, A necessary and sufficient condition for the unique decomposition of coded messages, IRE. Internat. Conv. Rec. 8 (1953), 104-108.

- This definition may look a little daunting at first, but it should be clearer if we take it step by step.
- we start with $\mathcal{C}_0 = \mathcal{C}$, we then construct each $\mathcal{C}_n$ $(n \geq 1)$ in terms of its predecessor $\mathcal{C}_{n-1}$, and finally we take $\mathcal{C}_\infty = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots$.
- Note that for $n = 1$ the definition of $\mathcal{C}_n$ can be simplified.
- Since $\mathcal{C}_{n-1} = \mathcal{C}_0 = \mathcal{C}$ the two conditions separated by the word "or" in the definition of $\mathcal{C}_n$ are identical, so

$$\mathcal{C}_1 = \{w \in T^+ | uw = v \text{ where } u, v \in \mathcal{C}\}.$$

- Note also that if $\mathcal{C}_{n-1} = \emptyset$ then $\mathcal{C}_n = \emptyset$, so iterating this gives $\mathcal{C}_{n+1} = \mathcal{C}_{n+1} = \cdots = \emptyset$.

Example

Let $\mathcal{C} = \{0, 01, 011\}$*. Then* $\mathcal{C}_1 = \{1, 11\}$*. At the next stage, with* $n = 2$*, inspection shows that there is no* $w \in \mathcal{D}^*$ *satisfying* $uw = v$ *where* $u \in \mathcal{C}$*,* $v \in \mathcal{C}_1$ *or vice versa. Thus* $\mathcal{C}_2 = \emptyset$*, so* $\mathcal{C}_n = \emptyset$ *for all* $n \geq 2$ *and hence* $\mathcal{C}_\infty = \mathcal{C}_1 = \{1, 11\}$*.*

From the definition of $\mathcal{C}_\infty$, it is conceivable that the construction of this set might take infinitely many steps, requiring a new set $\mathcal{C}_n$ to be constructed for each $n \geq 1$. The following theorem shows that one can always construct $\mathcal{C}_\infty$ in finitely many steps.

Theorem

If $\mathcal{C}$ has code-words of lengths $l_1, \cdots, l_q$, and $w \in \mathcal{C}_n$ for some n, then $|w| \leq l = \max(l_1, \cdots, l_q)$. Then each $\mathcal{C}_n$ is finite, and the sequence of the sets $\mathcal{C}_0$, $\mathcal{C}_1$, $\cdots$ is eventually periodic.

Proof.

- Use induction on n.
- If $n = 0$ then $\mathcal{C}_n = \mathcal{C}$, so $|w| \leq l$.
- If $n > 0$ then for $w \in \mathcal{C}_n$, $uw = v$ with $uw = v$ with $v \in \mathcal{C}_{n-1}$ or $\mathcal{C}$, so $|w| \leq |v| \leq l$ by induction or by definition of l respectively.
- There are only $N = r + r^2 + \cdots + r^l = r(r^l - 1)/(r - 1)$ nonempty r-ary words w with $|w| \leq l$, so $|\mathcal{C}_n| \leq N$ for each n.
- There are only 2^N different sets of such words w, so within the sets $\mathcal{C}_0, \cdots, \mathcal{C}_{2^N}$ there must be a repetition, $\mathcal{C}_j = \mathcal{C}_i$ with $i < j \leq 2^N$.

Proof.

- Note that

$$\mathcal{C}_n = \{w \in \mathcal{D}^* | uw = v \,\text{where}\, u \in \mathcal{C},\ v \in \mathcal{C}_{n-1}\ or\ u \in \mathcal{C}_{n-1}, v \in \mathcal{C}\}$$

- So each $\mathcal{C}_n$ depends only on $\mathcal{C}$ and $\mathcal{C}_{n-1}$.
- So $\mathcal{C}_{j+k} = \mathcal{C}_{i+k}$ for all $k \geq 0$.
- Hence each $\mathcal{C}_n = \mathcal{C}_0$ or $\mathcal{C}_1$ or $\cdots$ or $\mathcal{C}_{j-1}$, so $\mathcal{C}_\infty = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{j-1}$.
- Thus we have constructed all of $\mathcal{C}_\infty$ as soon as we find a repetition among the successive sets $\mathcal{C}_0$, $\mathcal{C}_1$, $\cdots$.

□

Example

Consider the ternary code $\mathcal{C} = \{02, 12, 120, 20, 21\}$*. Then* $\mathcal{C}_1 = \{0\}$, $\mathcal{C}_2 = \{2\}$, $\mathcal{C}_3 = \{0, 1\}$, $\mathcal{C}_4 = \{2, 20\}$, $\mathcal{C}_5 = \{0, 1\}$*; the repetition* $\mathcal{C}_5 = \mathcal{C}_3$ *implies that* $\mathcal{C}_n = \{0, 1\}$ *or* $\{2, 20\}$ *for odd or even* $n \geq 3$*, so* $\mathcal{C}_\infty = \mathcal{C}_1 \cup \cdots \mathcal{C}_4 = \{0, 1, 2, 20\}$.

Example

Consider the ternary code $\mathcal{C} = \{02, 12, 120, 21\}$. *Then* $\mathcal{C}_1 = \{0\}$, $\mathcal{C}_2 = \{2\}$, $\mathcal{C}_3 = \{1\}$, $\mathcal{C}_4 = \{2, 20\}$, $\mathcal{C}_5 = \{1\}$; *again* $\mathcal{C}_5 = \mathcal{C}_3$ *implies that* $\mathcal{C}_n = \{1\}$ *or* $\{2, 20\}$ *for odd or even* $n \geq 3$, *so* $\mathcal{C}_\infty = \mathcal{C}_1 \cup \cdots \mathcal{C}_4 = \{0, 1, 2, 20\}$.

Sardinas-Patterson Theorem

Theorem

A code $\mathcal{C}$ is uniquely decodable if and only if the sets $\mathcal{C}$ and $\mathcal{C}_\infty$ are disjoint.

- Since the proof of the Sardinas-Patterson Theorem is rather long, we will not give the full proof.
- Instead, we will give two typical arguments to illustrate the ideals involved.

- ($\Rightarrow$) Suppose that $\mathcal{C} \cap \mathcal{C}_\infty \neq \emptyset$, say $w \in \mathcal{C} \cap \mathcal{C}_2$; thus $uw = v$ with $u \in \mathcal{C}$ and $v \in \mathcal{C}_1$ or vice versa.
- Assume that the first case holds, then $u'v = v'$ when $u', v' \in \mathcal{C}$, so the sequence $\mathbf{t} = u'vw \in T^*$ could represent a sequence $\mathbf{s}$ of three source-symbols (since $u', u, w \in \mathcal{C}$) or one source-symbol (since $u; uw = u'v = v' \in \mathcal{C}$). Thus decoding is not unique.
- If the second case holds, where $uw = v$ with $u \in \mathcal{C}_1$ and $v \in \mathcal{C}$. Since $u \in \mathcal{C}_1$, $u'u = v'$ for some $u', v' \in \mathcal{C}$, so $\mathbf{t} = u'uw$ decodes as $u'v$ or $v'w$.

- ($\Leftarrow$) Suppose that we have an instance of non-unique decoding of the form $\mathbf{t} = u_1 u_2 = v - 1 v_2$, where $u_1, u_2, v_1, v_2 \in \mathcal{C}$.
- We cannot have $|u_1| = |v_1|$, for this we would have $u_1 = v_1$ and so $u_2 = v_2$.
- Remembering if necessary, we may therefore assume that $|u_1| > |v_1|$, so $u_1 = v_1 w$ where $|w| > 0$.
- Then $w \in \mathcal{C}_1$, so $u_2 \in \mathcal{C}_2$ since $w u_2 = v_2$. Thus $u_2 \in \mathcal{C} \cap \mathcal{C}_\infty$, so $\mathcal{C}$ and $\mathcal{C}_\infty$ are not disjoint.

Example

Let $\mathcal{C}$ *be the ternary code* $\{01, 1, 2, 210\}$. *We find that* $\mathcal{C}_1 = \{10\}$, $C_2 = \{0\}$ *and* $\mathcal{C}_3 = \{1\}$, *so* $1 \in \mathcal{C} \cap \mathcal{C}_\infty$ *and thus* $\mathcal{C}$ *is not uniquely decodable. An example of non-unique decodability is that the code-sequence* $\mathbf{t} = 2101$ *can be decoded as* 210.1 *or as* $2.1.01$.

Outline

1. Source 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

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 **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 **x** have encoded length l. Then, defining $l_{\min} = \min_i l_i$ and $l_{\max} = \max_i l_i$:

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

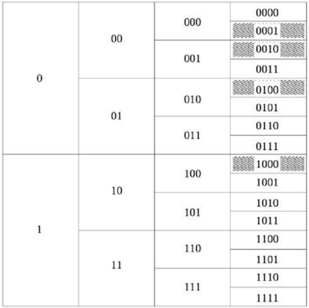
Proof.

Now assume C is uniquely decodable, so that for all **x** $\neq$ **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

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

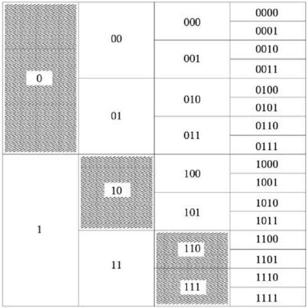
Thus $S^N \leq l_{\max}N$ for all N. Now if S were greater than 1, then as N increases, S^N would be an exponentially growing function, and for large enough N, an exponential always exceeds a polynomial such as $l_{\max}N$. But our result ($S^N \leq l_{\max}N$) is true for any N. Therefore $S \leq 1$.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

C_0



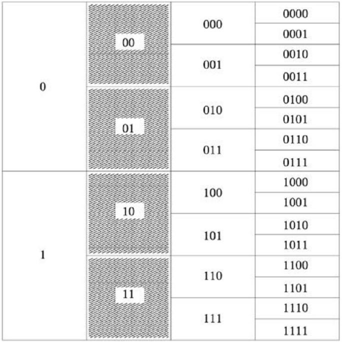
0	00	000	0000
			▨ 0001
		001	▨ 0010
			0011
	01	010	▨ 0100
			0101
		011	0110
			0111
1	10	100	▨ 1000
			1001
		101	1010
			1011
	11	110	1100
			1101
		111	1110
			1111

C_3

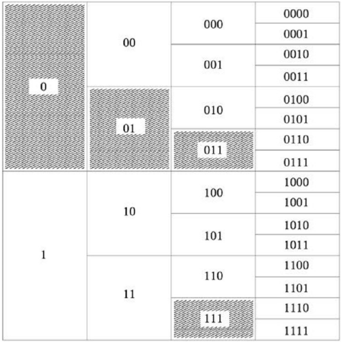


▨ 0	00	000	0000
			0001
		001	0010
			0011
	01	010	0100
			0101
		011	0110
			0111
1	▨ 10	100	1000
			1001
		101	1010
			1011
	11	▨ 110	1100
			1101
		▨ 111	1110
			1111

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

The same proof will give similar results for D-ary codes.

Theorem (McMillan)

The codeword lengths of any uniquely decodable D-ary code must satisfy the Kraft inequality

$$\sum D^{-l_i} \leq 1.$$

Corollary

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

Proof.

The point at which the preceding proof breaks down for infinite $|\mathcal{X}|$ is at the inequality $\sum_j D^{-l_j} \leq (kl_{\max})^{1/k}$, since for an infinite code $l_{\max}$ is infinite. But there is a simple fix to the proof. Any subset of a uniquely decodable code is also uniquely decodable; thus, any infinite subset of the infinite set of codewords satisfies the Kraft inequality. Hence,

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

Theorem

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

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

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 D^{-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 $\sum_{i=1}^{I} D^{-l_i}$ down the supermarket; if $\sum_i D^{-l_i} \leq 1$, we will have purchased all the codewords without running out of budget.

Proof.

- We only need to prove the existence of a D-ary prefix code with code-word lengths l_1, L_2, $\cdots$, l_m if these length satisfy the Kraft inequality.
- Without loss of generality, assume that $l_1 \leq l_2 \leq \cdots \leq l_m$ is satisfied.
- Consider all the D-ary sequences of lengths less than or equal to l_m and regard them as the nodes of the full D-ary tree of depth l_m.
- We will refer to a sequence of length l as a node of **order** l. Our strategy is to choose nodes as codewords in non-decreasing order of the codeword lengths.

Proof.

- Specifically, we choose a node of order l_1 as the first codeword, then a node of order l_2 as the second codeword, so on and so forth, such that each newly chosen codeword is not prefixed by any of the previously chosen codewords.
- If we can successfully choose all the m codewords, then the resultant set of codewords forms a prefix code with the desired set of lengths.
- There are $D^{l_1} > 1$ (since $l_1 \geq 1$) nodes of order l_1 which can be chosen as the first codeword.
- Thus choosing the first codeword is always possible.

Proof.

- Assume that the first i codewords have been chosen successfully, where $1 \leq i \leq m-1$, and we want to choose a node of L_{i+1} as the $(i+1)$st codewords.
- In other words, the $(i+1)$st node to be chosen cannot be a descent of any of the previously chosen codewords.
- Observe that for $1 \leq j \leq i$, the codeword with length l_j has $D^{l_{i+1}-l_j}$ descendants of order l_{i+1}.
- Since all the previously chosen codewords are not prefixes of each other, their descendants of order l_{i+1} do not overlap.
- Therefore, upon noting that the total number of nodes of order l_{i+1} is D^{l_i+1}, the number of nodes which can be chosen as the $(i+1)$st codeword is

$$D^{l_{i+1}} - D^{l_{i+1}-l_1} - \cdots - D^{l_{i+1}-l_i}.$$

Proof.

- If l_1, l_2, $\cdots$, l_m satisfy the Kraft inequality, we have

$$D^{-l_1} + \cdots + D^{-l_i} + D^{-l_{i+1}} \leq 1.$$

- Multiplying by $D^{l_{i+1}}$ and rearranging the terms, we have

$$D^{l_{i+1}} - D^{l_{i+1}-l_1} - \cdots - D^{l_{i+1}-l_i} \geq 1.$$

- The left-hand side is the number of nodes which can be chosen.
- Thus we have shown the existence of a prefix code with code word lengths l_1, l_2, $\cdots$, l_m, completing the proof.

□

Outline

1. Source 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?

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 $l_i = \log(1/p_i)$. □

Definition

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 if and only if 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 code-word 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.
- We will give a good suboptimal procedure (Shannon–Fano coding) later.
- In the next lecture we describe a simple procedure (Huffman coding) for actually finding the optimal code.

Theorem

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

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

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

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

Theorem (Source coding theorem for symbol codes)

Let $l_1^*, l_2^*, \cdots, l_m^*$ *be optimal codeword lengths for a source distribution* $\mathbf{p}$ *and a* D*-ary alphabet, and let* L^* *be the associated expected length of an optimal code* $L^* = \sum p_i l_i^*$. *Then*

$$H_D(X) \leq L^* < H_D(X) + 1.$$

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 codeword length per symbol L_n^* 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

1. Source 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?

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 $D(p\|q)$.