Introduction to Information Sciences

Information Theory
Shannon's Source Code Theorem

mcuturi@i.kyoto-u.ac.jp

Summary of Today's Lecture

- Reminders on last lecture
- Codes and uniquely decodable codes
- Shannon source code theorem

Information and Entropy

- For a random variable X taking values in a finite set $\mathcal{X}$ with probability p, we call the entropy of X,

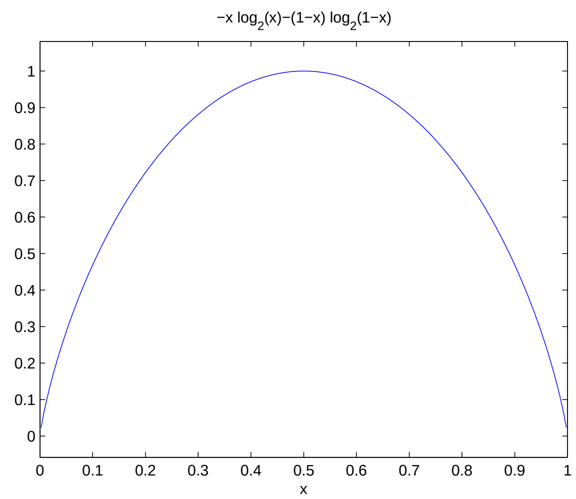
$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

N i.i.d. **random** variables *each* with entropy $H(X)$
can be compressed into more than $NH(X)$ bits with negligible risk of information loss, as N tends to infinity

Conversely, if they are **compressed into fewer** than $NH(X)$ bits it is virtually certain that information **will be lost**.

Entropy for binary random variables

- Two outcomes for a random variable X, 0 or 1.
- Two probabilities, $p_0 = p(X = 0)$ and $p_1 = p(X = 1)$.
- Moreover, $p_0 = p_1 - 1$, hence $H(X) = -p_1 \log p_1 - (1 - p_1) \log(1 - p_1)$.
- This is the curve represented below. $H(X) = 1$



- When $p_1 = \frac{1}{2}$, the entropy is at its **maximum**...

  ...which is why we cannot do better, on average, than ***actually*** *send* $1,000,000$ *bits if we want to* ***communicate*** $1,000,000$ *bits...*

Information and Entropy

Whatever the method used to design the **signal**,
if the word is made up of N **observations**
of **i.i.d random variables** distributed like X,
the **signal** **cannot be shorter on average than** $NH(X)$.

Information and Entropy

- Shannon's source code theorem gives a **lower bound**.
- The **reference length** becomes $NH(X)$,
- The main question of **coding and compression theory**:

how to define **compression mechanisms (codes)**
to **transform** messages into **shorter** signals
so as to get **as close as possible to Shannon's bound**
without necessarily knowing p?

Codes

Code: Definition

Code: rule to **convert** a piece of **information**
($e.g.$, a letter, word, phrase, gesture)
into **another form**, not necessarily of the same type.

- For these lectures: Σ_1, Σ_2, two finite alphabets.
- A code: a partial function from Σ_1^* to Σ_2^*

$$C : U \subset \Sigma_1^* \rightarrow V \subset \Sigma_2^*$$

Types of Code

- **Error Correcting Code**: code strings in $\Sigma_1^\star$ as strings in $\Sigma_2^\star$.
  - Of which Block Codes: $\Sigma_1^k \rightarrow \Sigma_2^n$

- **Variable Length Code**: **only source symbols** of Σ_1 are mapped to $\Sigma_2^\star$.
- **Extension** of a variable length code: code words in $\Sigma_1^\star$ by concatenating codewords in $\Sigma_2^\star$.

Types of Code

- Variable Length Code: source symbols of Σ_1 mapped to $\Sigma_2^\star$.
  - **Non-singular codes**: coding mechanism $C : \Sigma_1 \rightarrow \Sigma_2^\star$ is **injective**.
  - **Uniquely decodable codes**: extension of C to $\Sigma_1^\star$ is **non-singular**.
  - **Prefix Codes**: $C(x) = m$ and $C(x') = m' \rightarrow m$ **cannot** be a prefix of m'.

Prefix Codes $\subset$ **Uniquely decodable codes**
Uniquely decodable codes $\subset$ **Non-singular codes**
Non-singular codes $\subset$**Variable Length**

Variable Length Codes - Quizz

For each code below,

$$M_0 = \{\, ab \mapsto 1, bb \mapsto 0, ba \mapsto 111, aa \mapsto 001 \,\}$$

$$M_1 = \{\, a \mapsto 0, b \mapsto 0, c \mapsto 1 \,\}$$

$$M_2 = \{\, a \mapsto 0, b \mapsto 10, c \mapsto 110, d \mapsto 111 \,\}$$

$$M_3 = \{\, a \mapsto 1, b \mapsto 011, c \mapsto 01110, d \mapsto 1110, e \mapsto 10011 \,\}$$

$$M_4 = \{\, a \mapsto 0, b \mapsto 01, c \mapsto 011 \,\}$$

specify if the code is

1. Variable Length 2. Non-singular 3. Uniquely decodable 4. Prefix

Source Code Theorem

Shannon's Source Code Theorem

- Suppose that X is a r.v. taking values in Σ_1.
- Let f be a **uniquely decodable** code from Σ_1 to Σ_2^* where $|\Sigma_2| = a$.
- Let S denote the random variable given by the wordlength $f(X)$.

If f is optimal (with minimal expected wordlength) for X, then

$$\frac{H(X)}{\log_2 a} \leq \mathbb{E}S < \frac{H(X)}{\log_2 a} + 1$$

(Shannon 1948)

ref: Wikipedia article

Proof of Shannon's Source Code Theorem

- Let s_i be the wordlength of each possible wordcode

$$y_i \in \Sigma_2^\star$$

  coding for the i^{th} symbol of Σ_1, i.e. $y_i = f(x_i)$.

- Define

$$q_i = a^{-s_i}/C,$$

  where C is chosen so that $\sum q_i = 1$.

Two tools to prove it : Gibbs (KL)

$\boxed{\text{Gibb's Inequality}}$

- Kullback-Leibler divergence between $p = (p_1, \cdots, p_n)$ and $q = (q_1, \cdots, q_n)$

$$D_{\mathrm{KL}}(P\|Q) = \sum_{i=1}^{n} p_i \log_2 \frac{p_i}{q_i} \geq 0.$$

- equivalently,

$$-\sum_{i=1}^{n} p_i \log_2 p_i \leq -\sum_{i=1}^{n} p_i \log_2 q_i$$

Two tools to prove it: Kraft

Kraft's Inequality

- Let each source symbol from the alphabet

$$S = \{ s_1, s_2, \ldots, s_n \}$$

be encoded into a **uniquely decodable code** over an alphabet of size r with codeword lengths

$$\ell_1, \ell_2, \ldots, \ell_n.$$

- Then $\sum_{i=1}^{n} \left(\frac{1}{r}\right)^{\ell_i} \leq 1$.
- Conversely,

$$\forall \ell_1, \ell_2, \ldots, \ell_n \in \mathbf{N}$$

satisfying the inequality, $\exists$ a uniquely decodable code over an alphabet of size r with those codeword lengths.

Proof of Shannon's Source Code Theorem

- Using the chain of inequalities,

$$
\begin{aligned}
H(X) &= -\sum_{i=1}^{n} p_i \log_2 p_i \leq -\sum_{i=1}^{n} p_i \log_2 q_i \\
&= -\sum_{i=1}^{n} p_i \log_2 a^{-s_i} + \sum_{i=1}^{n} p_i \log_2 C \\
&= -\sum_{i=1}^{n} p_i \log_2 a^{-s_i} + \log_2 C \leq -\sum_{i=1}^{n} -s_i p_i \log_2 a \leq \mathbb{E} S \log_2 a
\end{aligned}
$$

- the second line follows from *Gibbs' inequality*.
- the fifth line follows from *Kraft's inequality*.

Proof of Shannon's Source Code Theorem

- For the second inequality we set

$$s_i = \lceil -\log_a p_i \rceil$$

so that

$$-\log_a p_i \leq s_i < -\log_a p_i + 1$$

and so

$$a^{-s_i} \leq p_i$$

and

$$\sum a^{-s_i} \leq \sum p_i = 1.$$

Proof of Shannon's Source Code Theorem

- By Kraft's inequality there exists a prefix-free code having those wordlengths.
- Thus the minimal S satisfies

$$
\begin{aligned}
\mathbb{E}S &= \sum p_i s_i \\
&< \sum p_i \left(-\log_a p_i + 1\right) \\
&= \sum -p_i \frac{\log_2 p_i}{\log_2 a} + 1 \\
&= \frac{H(X)}{\log_2 a} + 1.
\end{aligned}
$$

Assignment

- Write a report on either:
  - Lempel-Ziv compression
  - Arithmetic coding
  - Context-tree weighting (harder but more interesting)
  - Any lossless compression algorithm of your choice
  - JPEG compression