Introduction to Information Sciences

Information Theory
Shannon’s Entropy

mcuturi@i.kyoto-u.ac.jp
Summary of Today’s Lecture

- Shannon’s framework for information
- Shannon’s entropy
Starting point...

Not everything that can be counted counts,
and not everything that counts can be counted.
(Einstein)

- For things which **can be counted**, the science which provides a framework to
  - **store** *(efficiently)* that information,
  - **communicate** *(efficiently)* that information between individuals/computers,

is a branch of

- mathematics,
- statistics,
- electrical engineering, *etc.*

called **information theory**
Some History

- Unlike most disciplines, the exact birth-date of information theory is known.


Claude Shannon in 1948 (32 years old)

- Shannon proposed both a new problem and a few answers.
Claude Shannon, April 30, 1916 February 24, 2001


• After graduate studies at MIT, Work at
  ○ Princeton, (Von Neumann, Einstein)
  ○ Bell labs, (Turing during war) mainly work on cryptography
  ○ back to MIT from 50’s

• Closer to us, first recipient of the Kyoto prize in 1985,
Shannon’s framework

- this diagram, from the original paper, defines the usual problems of communication

![Diagram of a general communication system.](image)

- how to convert efficiently a message into a signal (transmitter)
- how to decipher efficiently the signal back into a message (receiver)
- how to cope with noisy environments which alter the signal.

- before Shannon, different approaches for each type of signal
  - telegraph,
  - texts,
  - codes,

- after Shannon, a unifying theory on all information.
A short movie by Charles and Ray Eames

- The Eames couple are most known for their industrial design

- this documentary was shot in 1953...

- Merely 5 years after Shannon’s breakthrough!
Shannon’s framework through examples

• Example: you do $N$ coin flips,

and record the results in a long word

$$b_1 \cdots b_N$$

where each $b_i$ is either Tails or Heads, that is $b_i \in \{T, H\}$.

• To keep things a bit more simple we use $\{0, 1\}$ instead of $\{T, H\}$.

• You want to send the outcome of this experiment of $N$ coin tosses to someone.
Information and Entropy

• if \( N = 1000 \) you can write 0111011100 \( \cdots \) 101 on a piece of paper and send it

• More handy approach: punch holes in cards,
  ...with the convention “hole=1”, “no hole=0”.

• to each hole corresponds one coin toss, ordered by time.

• The information given by each location (hole/no hole) on the card is a **bit**.

**bit** = **binary digit**, coined down in 1948 by Shannon (originally Tukey in ’37)
• if $N = 1,000,000$, is there an efficient way to transmit this information?

• intuitively, this is hopeless:
  ○ each coin toss is independent,
  ○ each coin toss has two equally likely outcomes, 1 or 0.
  ○ you must provide the information for each coin toss.

• If your coin tosses is very atypical... *e.g.*

  "I made 1,000,000 coin tosses and only had Heads"

  ...you may get away with a very short message...

• unfortunately, you will more likely need 1,000,000 bits of information.

• we will show this later.
Information and Entropy

Suppose the coin is actually **biased**

- Suppose that the probability of tails (0) is $p_0 = 1/3$ & heads (1) is $p_1 = 2/3$.
- Yet we know that, on average, we will have to punch more holes than not, as

\[ p_1 = 2/3 > p_0. \]

... twice more 1’s than 0’s... we might consider punching 0’s instead!!

- yet, if we send the exact result, $b_1 b_2 \cdots b_N$, we still need 1,000,000 bits.

What about **taking advantage** of the **differences** in probabilities $p_0 \neq p_1$ to **design a shorter message**?
• Simple approach: since the events are independent...
  ...for all $i \leq N - 1$, the probability that

$$\begin{align*}
  p(b_ib_{i+1} = 00) &= 1/9 \\
  p(b_ib_{i+1} = 01) &= 2/9 \\
  p(b_ib_{i+1} = 10) &= 2/9 \\
  p(b_ib_{i+1} = 11) &= 4/9
\end{align*}$$

• We could also consider 8 triplets, 16 quadruplets, etc.

• Let’s rewrite our tosses $b_1 \cdots b_N$ two by two, using some notations:
  ○ $a = 00$, $b = 01$, $c = 10$, $d = 11$.
  ○ We could send sequences of one of four letters, $abdcabb \cdots$.

• no gain so far... each letter needs $500.000 \times 2$ bits.
Information and Entropy

- Remember that

\[
\begin{align*}
    p(b_i b_{i+1} = a) &= 1/9 \\
    p(b_i b_{i+1} = b) &= 2/9 \\
    p(b_i b_{i+1} = c) &= 2/9 \\
    p(b_i b_{i+1} = d) &= 4/9
\end{align*}
\]

- Following the same idea, we will have, on average, a lot more d’s than b or c’s and few a’s.

- Let’s translate back \(a, b, c, d\) back into binary codes. Setting \textbf{for instance}\(^1\)
  \begin{itemize}
    \item \(d = 0\),
    \item \(c = 10\),
    \item \(b = 110\),
    \item \(a = 111\).
  \end{itemize}

- Intuition: \textbf{LONG} codewords for unlikely tokens.

---

\(^{1}\text{This is called a Huffman code}\)
In our example,

\[1011101001110111\] (16 bits)
\[cdccbdbd\] (2 \times \text{bits})
\[100101011001100\] (15 bits)
• **On average**, as $N$ goes to infinity and given $N$ tosses,

  - the naive technique needs $N$ bits,
  - our trick requires $\frac{N}{2} \times (1p_d + 2p_c + 3p_b + 3p_d) = \frac{N}{2}(\frac{4}{9} + \frac{4}{9} + \frac{6}{9} + \frac{3}{9}) = \frac{17}{18}N$

• not so bad for such a simple trick.

• We could actually take advantage further of this trick by considering triplets, quadruplets etc...

• Shannon’s theorem tells us something far more powerful
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; but conversely, if they are compressed into fewer than $NH(X)$ bits it is virtually certain that information will be lost.

- In the previous example,

$$H(b) = -p_1 \log_2 p_1 - p_0 \log_2 p_0 = -\frac{2}{3} \log_2 \left( \frac{2}{3} \right) - \frac{1}{3} \log_2 \left( \frac{1}{3} \right) \approx 0.918$$

- We had $17/18 = 0.944...$ getting closer.
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...
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 $NH(X)$.

- 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 $p$?