Information Theory lecture 2

COMSM0075 Information Processing and Brain

`comsm0075.github.io`

September 2020

Shannon's entropy

For a finite discrete distribution with random variable X, possible outcomes $\{x_1, x_2, \ldots x_n\} \in \mathcal{X}$ and a probability mass function p_X giving probabilities $p_X(x_i)$, the entropy is

$$H(X) = -\sum_{x_i \in \mathcal{X}} p_X(x_i) \log_2 p_X(x_i)$$

Shannon's entropy

For a finite discrete distribution with random variable X, possible outcomes $\{x_1, x_2, \ldots x_n\} \in \mathcal{X}$ and a probability mass function p_X giving probabilities $p_X(x_i)$, the entropy is

$$H(X) = -\sum_{x_i \in \mathcal{X}} p_X(x_i) \log_2 p_X(x_i)$$

Shannon's entropy

For a finite discrete distribution with random variable X, possible outcomes $\{x_1, x_2, \dots x_n\} \in \mathcal{X}$ and a probability mass function p_X giving probabilities $p_X(x_i)$, the entropy is

$$H(X) = -\sum_{x_i \in \mathcal{X}} p_X(x_i) \log_2 p_X(x_i)$$

Shannon's entropy

For a finite discrete distribution with random variable X, possible outcomes $\{x_1, x_2, \ldots x_n\} \in \mathcal{X}$ and a probability mass function p_X giving probabilities $p_X(x_i)$, the entropy is

$$H(X) = -\sum_{x_i \in \mathcal{X}} p_X(x_i) \log_2 p_X(x_i)$$

In this definition $p \log_2 p = 0$ when $p = 0$; this makes sense since

$$\lim_{p \to 0} p \log_2 p = 0$$

Shannon's entropy

For a finite discrete distribution with random variable X, possible outcomes $\{x_1, x_2, \ldots x_n\} \in \mathcal{X}$ and a probability mass function p_X giving probabilities $p_X(x_i)$, the entropy is

$$H(X) = -\langle \log_2 p_X(X) \rangle$$

Shannon's entropy

For a finite discrete distribution with random variable X, possible outcomes $\{x_1, x_2, \dots x_n\} \in \mathcal{X}$ and a probability mass function p_X giving probabilities $p_X(x_i)$, the entropy is

$$H(X) = -\langle \log_2 p_X(X) \rangle$$

Shannon's entropy

Shannon's entropy has lots of nice properties, being easy to estimate isn't one.

works on any sample space

The mean of a distribution

$$\langle x \rangle = \sum_{x_i \in \mathcal{X}} p_X(x_i) x_i$$

only works if the x_i live in a vector space.

works on any sample space

Not all sample spaces are vector spaces, trying to work out the average fruit bought in a grocers doesn't make sense because

$$0.25 \times \text{apple} + 0.125 \times \text{banana} + 0.1 \times \text{orange} \ldots$$

is nonsense.

it's always positive

$$H(X) = -\sum_{x_i \in \mathcal{X}} p_X(x_i) \log_2 p_X(x_i)$$

and since $0 \leq p_X(x_i) \leq 1$

$$H(X) \geq 0$$

it's zero if the distribution isn't random

If $p_X(x_i)$ look like $\{0, 0, \ldots, 1, \ldots 0\}$ then

$$H(X) = 0$$

uniform distribution

If the distribution is uniform

$$p_X(x_i) = \frac{1}{n}$$

for all x_i where

$$n = \#\mathcal{X}$$

then, since $-\log_2(1/n) = log_2 n$

$$H(X) = \log_2 n$$

bounds

In fact, not proved here but not difficult to prove,

$$0 \leq H(X) \leq n$$

with $H(X)$ only if one probability is one and the rest zero and $H(X) = n$ only for the uniform distribution.

bounds

$$0 \leq H(X) \leq n$$

That what we want!

$n = 2$

Two outcomes, a and b with $p(a) = p$ and $p(b) = 1 - p$ then

$$H = -p \log_2 p - (1 - p) \log_2 (1 - p)$$

$n = 2$

Two outcomes, a and b with $p(a) = p$ and $p(b) = 1 - p$ then

$$H = -p \log_2 p - (1 - p) \log_2 (1 - p)$$

$n = 2$

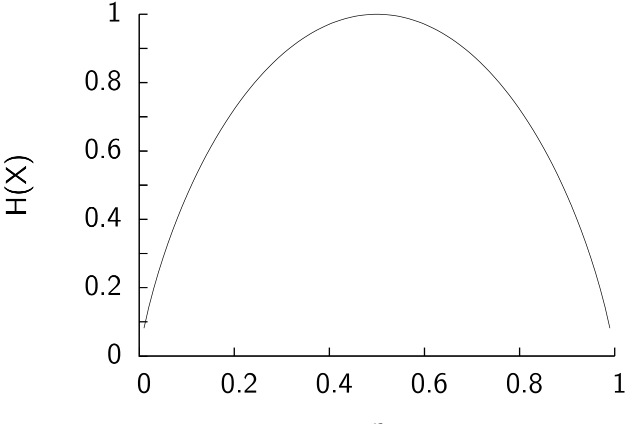
Two outcomes, a and b with $p(a) = p$ and $p(b) = 1 - p$ then

$$H = -p \log_2 p - (1-p) \log_2 (1-p)$$

$n = 2$

Two outcomes, a and b with $p(a) = p$ and $p(b) = 1 - p$ then

$$H = -p \log_2 p - (1 - p) \log_2 (1 - p)$$



source coding

The main reason to believe that Shannon's entropy is a good quantity for calculating entropy is its relationship with what is called source coding.

source coding

Consider storing a long sequence of the letters `A`, `B`, `C` and `D` as binary.

$\mathtt{AABACBDA}\ldots$

a dictionary might look like

A	B	C	D
00	01	10	11

a dictionary might look like

A	B	C	D
00	01	10	11

$AABACD\ldots$

becomes

$000001001011\ldots$

a dictionary might look like

A	B	C	D
00	01	10	11

$AABACD\ldots$

becomes

$000001001011\ldots$

a dictionary might look like

A	B	C	D
00	01	10	11

$AABACD\ldots$

becomes

$000001001011\ldots$

a dictionary might look like

A	B	C	D
00	01	10	11

$AABACD\ldots$

becomes

$000001001011\ldots$

a dictionary might look like

A	B	C	D
00	01	10	11

$AABACD\ldots$

becomes

$000001001011\ldots$

a dictionary might look like

A	B	C	D
00	01	10	11

$AABACD\ldots$

becomes

$000001001011\ldots$

a dictionary might look like

A	B	C	D
00	01	10	11

AABACD. . .

becomes

000001001011. . .

a dictionary might look like

A	B	C	D
00	01	10	11

$$AABACD\ldots$$

becomes

$$000001001011\ldots$$

average bits per letter

$$L = 2$$

say we know the letter frequencies

Now, say we also knew that

A	B	C	D
0.5	0.25	0.125	0.125

So in the message that will be encoded, A occurs half the time, B a quarter the time and C and D an eighth of the time.

say we know the letter frequencies

Can we use this information to make L smaller?

say we know the letter frequencies

Can we find an shorter encoding for the most frequent letter: `A`*?*

here is a better code

A	B	C	D
0	10	110	111

here is a better code - prefix free code

A	B	C	D
0	10	110	111

here is a better code - prefix free code

A	B	C	D
0	10	110	111

here is a better code

A	B	C	D
0	10	110	111

$AABACD\ldots$

becomes

$00100110111\ldots$

here is a better code

A	B	C	D
0	10	110	111

$AABACD\ldots$

becomes

$0010011011\ldots$

here is a better code

A	B	C	D
0	10	110	111

$AABACD\ldots$

becomes

$0010\,0110\,111\ldots$

here is a better code

A	B	C	D
0	10	110	111

$AABACD\ldots$

becomes

$0010011 0111\ldots$

here is a better code

A	B	C	D
0	10	110	111

$AABACD\ldots$

becomes

$0010011011\ldots$

here is a better code

A	B	C	D
0	10	110	111

$AABACD\ldots$

becomes

$0010011011 1\ldots$

here is a better code

A	B	C	D
0	10	110	111

$AABACD\ldots$

becomes

$0010011011 1\ldots$

this code is shorter

$$L = 0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75$$

this code is shorter

$$L = 0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75 < 2$$

this code is shorter

$$L = 0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75$$

where 0.5 is the frequency of `A` and 1 is the length of the code.

Shannon's entropy

$$H(X) = -0.5\log_2(0.5) - 0.25\log_2(0.25) - 0.250\log_2(0.125) = 1.75$$

The source coding theorem

Roughly, for the most efficient code

$$H(X) \leq L < H(X) + 1$$

The source coding theorem

$$H(X) \leq L < H(X) + 1$$

The source coding theorem shows that the entropy $H(X)$ is a lower bound on the average length of a message using the most efficient code.