Lecture 20

Wednesday, April 01, 2015 11:27 AM

Entropy

Information Theory, 1948, Claude Shannon

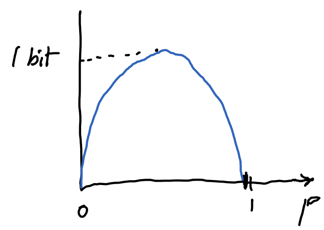
"bit" first appears in Shannon's papers.

$$H(X) = E \log_2 \frac{1}{p_X(X)} = \sum_{x \in \mathcal{X}} p_X(x) \log_2 \frac{1}{p_X(x)}$$ Units are bits.

Example 1: $X = \begin{cases} 0, & \text{w.p. } \frac{1}{2} \\ 1, & \text{w.p. } \frac{1}{2} \end{cases}$ $\quad p_X(x) = \begin{cases} \frac{1}{2}, & x \in \{0,1\} \\ 0, & \text{else} \end{cases}$

$$H(X) = \frac{1}{2} \log_2 \frac{1}{1/2} + \frac{1}{2} \log_2 \frac{1}{1/2} = \log_2 2 = 1 \text{ bit.}$$

Example 2: $p_X(x) = \begin{cases} p, & x = 1 \\ 1-p, & x = 0 \\ 0, & \text{else} \end{cases}$ $\quad H(X) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$



Suppose all $x \in \Omega$ are equally likely.

"Uniform distribution" $\quad p_X(x) = \frac{1}{|\Omega|} \quad \forall x \in \Omega$

$$H(X) = E \log \frac{1}{p(X)} = E \log |\Omega| = \log_2 |\Omega|$$

Entropy of a roll of a die:

$$H(X) = \log_2 6 = 2.58 \text{ bits}$$

Entropy of two dice: Let X_1 be the first die.

Let X_2 be the second r.v.

E.g. $(X_1, X_2) = (1, 5)$

Independence: $P(X_1 = 3 \text{ and } X_2 = 6) = P(X_1 = 3) \cdot P(X_2 = 6)$

$= \frac{1}{36}$

$p(a, b) = \text{Prob}(X_1 = a \text{ and } X_2 = b)$

$\Rightarrow p(x_1, x_2) = \left(\frac{1}{6}\right)\left(\frac{1}{6}\right) = \frac{1}{36} \quad \forall x_1 x_2$

$H(X_1, X_2) = \log_2 36 = 5.17 \text{ bits} = 2 \cdot (2.58 \text{ bits})$

If X_1 X_2 X_3 ... are independent: $H(X_1, X_2, X_3, \ldots) = \sum_i H(X_i)$

In general $H(X_1 \ X_2 \ X_3 \ldots) \leq \sum_i H(X_i)$

Example 3:

X

$P(\text{apple}) = \frac{1}{4}$

$P(\text{orange}) = \frac{1}{8}$

$P(\text{pear}) = \frac{1}{2}$

$P(\text{banana}) = \frac{1}{8}$

Okay that X is not numeric.

$H(X) = \frac{1}{4} \log_2 \frac{1}{1/4} + \frac{1}{8} \log_2 8 + \frac{1}{2} \log 2 + \frac{1}{8} \log 8$

$= 1.75 \text{ bits}$

Review: 20 Questions = Prefix code

Also, for coding

Uniquely decodable (no punctuation needed) $\Rightarrow$ Kraft inequality: $\sum_{x \in \Omega} 2^{-L(x)} \leq 1$ ← Should be tight

Kraft $\Rightarrow$ Exists a prefix code with those lengths.

Can be found by embedding in a binary tree.

Optimization Problem:

Find $L(x)$ that satisfy Kraft ineq. and minimize $EL(X)$

Integer

$\sum_{x \in \Omega} p_X(x) L(x)$

If we relax the problem, let $\hat{L}(x)$ not be integer, then easy.

Lower bound.

Can solve using Legrange mult. : $\hat{L}^*(x) = \log_2 \frac{1}{p_X(x)}$ ← If we round up. -Integer -satisfy Kraft

$$E\hat{L}^*(X) = E \log_2 \frac{1}{p_X(X)} = H(X)$$

$$\Rightarrow EL(X) \geq H(X)$$

with integer constraints.

Optimal code: $EL(X) < H(X) + 1$ bits

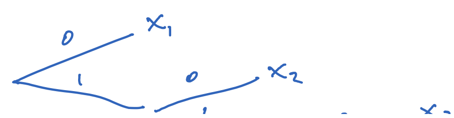
Shannon code: Let $L(x) = \lceil \log_2 \frac{1}{p_X(x)} \rceil$

Usually not optimal — Doesn't even give equality in Kraft ineq.
(unless no rounding)

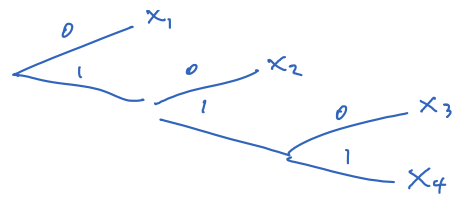
Does give $EL(X) < H(X) + 1$ bit.

Shannon-Fano code:
- Sort the outcomes by probability
- Make cuts as close to $P = \frac{1}{2}$ as possible.

	$P(x)$
x_1	0.45
x_2	0.2
x_3	0.2



x_1	[illegible]
x_2	0.2
x_3	0.2
x_4	0.15



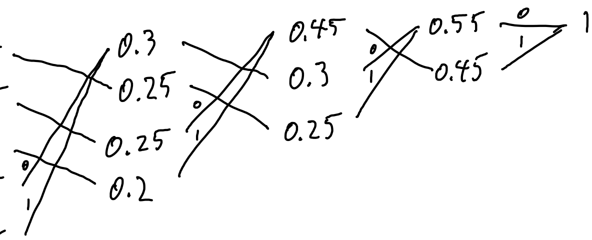
Huffman code: 1951

- Design the tree from the leaves

$$\Omega = \{x_1, x_2, x_3, x_4, x_5\}$$

$$p(x) = 0.25 \quad 0.25 \quad 0.2 \quad 0.15 \quad 0.15$$

$L(x)$	Codewords		$p(x)$
2	01	x_1	0.25
2	10	x_2	0.25
2	11	x_3	0.2
3	000	x_4	0.15
3	001	x_5	0.15



Summary: Kraft $\Rightarrow$ $E\,L(X) \geq H(X)$ Entropy

- Achieved if $L(x)$ not required to be integer by setting $L(x) = \log_2 \frac{1}{p(x)}$
- Achieved if $\log_2 \frac{1}{p(x)}$ is an integer $\forall x$

Shannon code: $L_s(x) = \left\lceil \log_2 \frac{1}{p(x)} \right\rceil$, $\quad E\,L_s(X) < H(X) + 1$ bit

Notice that if rounding occurs, Kraft is loose.
$\Rightarrow$ Shannon code is optimal if and only if $\log_2 \frac{1}{p(x)}$ are integers.

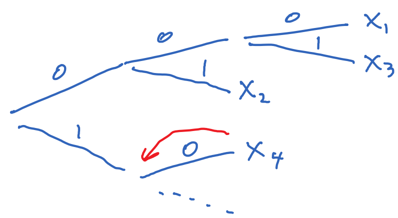
Example: $\Omega = \{x_1, x_2, x_3, x_4\}$

$p(x_1) = \frac{1}{6}$ $\qquad L(x_1) = \lceil \log_2 6 \rceil = 3$

$p(x_2) = \frac{1}{3}$ $\qquad L(x_2) = 2$

$p(x_3) = \frac{1}{6}$ $\qquad L(x_3) = 3$

$p(x_4) = \frac{1}{3}$ $\quad L(x_4) = 2$



Code:

x_1	000
x_2	01
x_3	001
x_4	10

Huffman code is optimal.