DD2448 Foundations of Cryptography
Lecture 7

Douglas Wikström
KTH Royal Institute of Technology
dog@kth.se

March 31, 2020

Information Theory

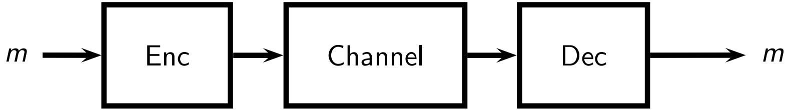
Information Theory

- Information theory is a mathematical theory of communication.
- Typical questions studied are how to compress, transmit, and store information.
- Information theory is also useful to argue about some cryptographic schemes and protocols.

Classical Information Theory

- **Memoryless Source Over Finite Alphabet.** A source produces symbols from an alphabet $\Sigma = \{a_1, \ldots, a_n\}$. Each generated symbol is independently distributed.
- **Binary Channel.** A binary channel can (only) send bits.
- **Coder/Decoder.** Our goal is to come up with a scheme to:
  1. convert a symbol a from the alphabet Σ into a sequence $(b_1, \ldots, b_l)$ of bits,
  2. send the bits over the channel, and
  3. decode the sequence into a again at the receiving end.

Classical Information Theory



Alice

Bob

Optimization Goal

We want to minimize the **expected** number of bits/symbol we send over the binary channel, i.e., if X is a random variable over Σ and $l(x)$ is the length of the codeword of x then we wish to minimize

$$\mathrm{E}\left[l(X)\right] = \sum_{x \in \Sigma} \mathsf{P}_X\left(x\right) l(x) \ .$$

Examples:

- X takes values in $\Sigma = \{a, b, c, d\}$ with uniform distribution. How would you encode this?

Examples:

- X takes values in $\Sigma = \{a, b, c, d\}$ with uniform distribution. How would you encode this?

It seems we need $l(x) = \log|\Sigma|$. This gives the Hartley measure.

Examples:

- X takes values in $\Sigma = \{a, b, c, d\}$ with uniform distribution. How would you encode this?
- X takes values in $\Sigma = \{a, b, c\}$, with $\mathrm{P}_X(a) = \frac{1}{2}$, $\mathrm{P}_X(b) = \frac{1}{4}$, and $\mathrm{P}_X(c) = \frac{1}{4}$. How would you encode this?

It seems we need $I(x) = \log |\Sigma|$. This gives the Hartley measure.
hmmm...

Examples:

- X takes values in $\Sigma = \{a, b, c, d\}$ with uniform distribution. How would you encode this?
- X takes values in $\Sigma = \{a, b, c\}$, with $\mathrm{P}_X(a) = \frac{1}{2}$, $\mathrm{P}_X(b) = \frac{1}{4}$, and $\mathrm{P}_X(c) = \frac{1}{4}$. How would you encode this?

It seems we need $I(x) = \log \frac{1}{\mathrm{P}_X(x)}$ bits to encode x.

Entropy

Let us turn this expression into a definition.

Definition. Let X be a random variable taking values in $\mathcal{X}$. Then the **entropy** of X is

$$H(X) = -\sum_{x \in \mathcal{X}} \mathrm{P}_X(x) \log \mathrm{P}_X(x) \quad .$$

Examples and intuition are nice, but what we need is a theorem that states that this is **exactly** the right expected length of an optimal code.

Definition. A function $f : \mathcal{X} \to (a, b)$ is **concave** if

$$\lambda \cdot f(x) + (1-\lambda) f(y) \leq f\left(\lambda \cdot x + (1-\lambda) y\right) \ ,$$

for every $x, y \in (a, b)$ and $0 \leq \lambda \leq 1$.

Jensen's Inequality

Definition. A function $f : \mathcal{X} \to (a, b)$ is **concave** if

$$\lambda \cdot f(x) + (1 - \lambda) f(y) \leq f\left(\lambda \cdot x + (1 - \lambda) y\right) \ ,$$

for every $x, y \in (a, b)$ and $0 \leq \lambda \leq 1$.

Theorem. Suppose f is continuous and strictly concave on (a, b), and X is a discrete random variable. Then

$$\mathrm{E}\left[f(X)\right] \leq f(\mathrm{E}\left[X\right]) \ ,$$

with equality iff X is constant.

Jensen's Inequality

Definition. A function $f : \mathcal{X} \rightarrow (a, b)$ is **concave** if

$$\lambda \cdot f(x) + (1 - \lambda) f(y) \leq f\left(\lambda \cdot x + (1 - \lambda) y\right) \ ,$$

for every $x, y \in (a, b)$ and $0 \leq \lambda \leq 1$.

Theorem. Suppose f is continuous and strictly concave on (a, b), and X is a discrete random variable. Then

$$\mathrm{E}\left[f(X)\right] \leq f(\mathrm{E}\left[X\right]) \ ,$$

with equality iff X is constant.

Proof idea. Consider two points + induction over number of points.

Kraft's Inequality

Theorem. There exists a prefix-free code E with codeword lengths l_x, for $x \in \Sigma$ if and only if

$$\sum_{x \in \Sigma} 2^{-l_x} \leq 1 \ .$$

Proof Sketch. $\Rightarrow$ Given a prefix-free code, we consider the corresponding binary tree with codewords at the leaves. We may "fold" it by replacing two sibling leaves $\mathrm{E}(x)$ and $\mathrm{E}(y)$ by (xy) with length $l_x - 1$. Repeat.

$\Leftarrow$ Given lengths $l_{x_1} \leq l_{x_2} \leq \ldots \leq l_{x_n}$ we start with the complete binary tree of depth l_{x_n} and prune it.

Binary Source Coding Theorem (1/2)

Theorem. Let E be an optimal code and let $l(x)$ be the length of the codeword of x. Then

$$H(X) \leq \mathrm{E}\left[l(X)\right] < H(X) + 1 \ .$$

Binary Source Coding Theorem (1/2)

Theorem. Let E be an optimal code and let $l(x)$ be the length of the codeword of x. Then

$$H(X) \leq \mathrm{E}\left[l(X)\right] < H(X) + 1 \ .$$

Proof of Upper Bound.
Define $l_x = \lceil -\log \mathrm{P}_X(x) \rceil$. Then we have

$$\sum_{x \in \Sigma} 2^{-l_x} \leq \sum_{x \in \Sigma} 2^{\log \mathrm{P}_X(x)} = \sum_{x \in \Sigma} \mathrm{P}_X(x) = 1$$

Kraft's inequality implies that there is a code with codeword lengths l_x. Then note that
$\sum_{x \in \Sigma} \mathrm{P}_X(x) \lceil -\log \mathrm{P}_X(x) \rceil < H(X) + 1$.

Binary Source Coding Theorem (2/2)

Proof of Lower Bound.

$$
\begin{aligned}
\mathrm{E}\left[l(X)\right] &= \sum_x \mathsf{P}_X(x)\, l_x \\
&= -\sum_x \mathsf{P}_X(x) \log 2^{-l_x} \\
&\geq -\sum_x \mathsf{P}_X(x) \log \mathsf{P}_X(x) \\
&= H(X)
\end{aligned}
$$

Input: $\{(a_1, p_1), \ldots, (a_n, p_n)\}$.
Output: 0/1-labeled rooted tree.

$\textsc{Huffman}(\{(a_1, p_1), \ldots, (a_n, p_n)\})$

(1) $S \leftarrow \{(a_1, p_1, a_1), \ldots, (a_n, p_n, a_n)\}$
(2) **while** $|S| \geq 2$
(3) Find $(b_i, p_i, t_i), (b_j, p_j, t_j) \in S$ with minimal p_i and p_j.
(4) $S \leftarrow S \setminus \{(b_i, p_i, t_i), (b_j, p_j, t_j)\}$
(5) $S \leftarrow S \cup \{\big(b_i \| b_j, p_i + p_j, \textsc{Node}(t_i, t_j)\big)\}$
(6) **return** S

Huffman's Code (2/2)

Theorem. Huffman's code is optimal.

Proof idea.
There exists an optimal code where the two least likely symbols are neighbors.

Let us turn this expression into a definition.

Definition. Let X be a random variable taking values in $\mathcal{X}$. Then the **entropy** of X is

$$H(X) = -\sum_{x \in \mathcal{X}} \mathrm{P}_X(x) \log \mathrm{P}_X(x) \quad .$$

Conditional Entropy

Definition. Let (X, Y) be a random variable taking values in $\mathcal{X} \times \mathcal{Y}$. We define **conditional entropy**

$$H(X|y) = -\sum_x \mathsf{P}_{X|Y}(x|y) \log \mathsf{P}_{X|Y}(x|y) \quad \text{and}$$

$$H(X|Y) = \sum_y \mathsf{P}_Y(y) H(X|y)$$

Note that $H(X|y)$ is simply the ordinary entropy function of a random variable with probability function $\mathsf{P}_{X|Y}(\cdot|y)$.

Properties of Entropy

Let X be a random variable taking values in $\mathcal{X}$.

Upper Bound. $H(X) = \mathrm{E}\left[-\log \mathrm{P}_X(X)\right] \leq \log |\mathcal{X}|$.

Chain Rule and Conditioning.

$$
\begin{aligned}
H(X, Y) &= -\sum_{x,y} \mathrm{P}_{X,Y}(x, y) \log \mathrm{P}_{X,Y}(x, y) \\
&= -\sum_{x,y} \mathrm{P}_{X,Y}(x, y) \left(\log \mathrm{P}_Y(y) + \log \mathrm{P}_{X|Y}(x|y) \right) \\
&= -\sum_{y} \mathrm{P}_Y(y) \log \mathrm{P}_Y(y) - \sum_{x,y} \mathrm{P}_{X,Y}(x, y) \log \mathrm{P}_{X|Y}(x|y) \\
&= H(Y) + H(X|Y) \leq H(Y) + H(X)
\end{aligned}
$$