

Lecture 3

Perfect Secrecy, Information Theory, and Spurious Keys and Unicity Distance

Douglas Wikström
KTH Stockholm
dog@csc.kth.se

January 24

- Perfect Secrecy
- Information Theory
- Spurious Keys and Unicity Distance

Quote of the Day

You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, nobody knows what entropy really is, so in a debate you will always have the advantage.

– von Neumann to Shannon, 1949

Perfect Secrecy (1/3)

When is a cipher perfectly secure?

Perfect Secrecy (1/3)

When is a cipher perfectly secure?

How should we formalize this?

Perfect Secrecy (2/3)

Definition. A cryptosystem has perfect secrecy if guessing the plaintext is as hard to do given the ciphertext as it is without it.

Perfect Secrecy (2/3)

Definition. A cryptosystem has perfect secrecy if guessing the plaintext is as hard to do given the ciphertext as it is without it.

Definition. A cryptosystem has perfect secrecy if

$$\Pr[M = m | C = c] = \Pr[M = m]$$

for every $m \in \mathcal{M}$ and $c \in \mathcal{C}$, where M and C are random variables taking values over \mathcal{M} and \mathcal{C} .

Perfect Secrecy (3/3)

Game Based Definition. Exp_A^b , where A is a strategy:

1. $k \leftarrow_R \mathcal{K}$
2. $(m_0, m_1) \leftarrow A$
3. $c = E_k(m_b)$
4. $d \leftarrow A(c)$, with $d \in \{0, 1\}$
5. Output d .

Definition. A cryptosystem has perfect secrecy if for every **computationally unbounded** strategy A ,

$$\Pr [\text{Exp}_A^0 = 1] = \Pr [\text{Exp}_A^1 = 1] \ .$$

One-Time Pad

One-Time Pad (OTP).

- ▶ **Key.** Random tuple $k = (b_0, \dots, b_{n-1}) \in \mathbb{Z}_2^n$.
- ▶ **Encrypt.** Plaintext $m = (m_0, \dots, m_{n-1}) \in \mathbb{Z}_2^n$ gives ciphertext $c = (c_0, \dots, c_{n-1})$, where $c_i = m_i \oplus b_i$.
- ▶ **Decrypt.** Ciphertext $c = (c_0, \dots, c_{n-1}) \in \mathbb{Z}_2^n$ gives plaintext $m = (m_0, \dots, m_{n-1})$, where $m_i = c_i \oplus b_i$.

Bayes' Theorem

Theorem. If A and B are events and $\Pr[B] > 0$, then

$$\Pr[A|B] = \frac{\Pr[A] \Pr[B|A]}{\Pr[B]}$$

Terminology:

$\Pr[A]$ – prior probability of A

$\Pr[B]$ – prior probability of B

$\Pr[A|B]$ – posterior probability of A given B

$\Pr[B|A]$ – posterior probability of B given A

One-Time Pad Has Perfect Secrecy

- ▶ **Probabilistic Argument.** Bayes implies that:

$$\begin{aligned}\Pr[M = m | C = c] &= \frac{\Pr[M = m] \Pr[C = c | M = m]}{\Pr[C = c]} \\ &= \Pr[M = m] \frac{2^{-n}}{2^{-n}} \\ &= \Pr[M = m] .\end{aligned}$$

- ▶ **Simulation Argument.** The ciphertext is uniformly and independently distributed from the plaintext. We can **simulate** it on our own!

Bad News

- ▶ **“The key must be as long as the plaintext”:**
 - ▶ $|\mathcal{K}| \geq |\mathcal{C}|$, since every ciphertext must have been created using some key.
 - ▶ $|\mathcal{C}| \geq |\mathcal{M}|$, since every plaintext must be encrypted in an invertible way.

Theorem. “For every cipher with perfect secrecy, the key requires at least as much space to represent as the plaintext.”

- ▶ **Dangerous in practice to rely on no reuse.**

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 $\Sigma = \{a_1, \dots, a_n\}$. Each generated symbol is identically and 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, \dots, 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

$$\mathbb{E}[l(X)] = \sum_{x \in \Sigma} P_X(x) l(x) .$$

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 $I(x) = \log |\Sigma|$. 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 $P_X(a) = \frac{1}{2}$, $P_X(b) = \frac{1}{4}$, and $P_X(c) = \frac{1}{4}$. 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 $P_X(a) = \frac{1}{2}$, $P_X(b) = \frac{1}{4}$, and $P_X(c) = \frac{1}{4}$. How would you encode this?

It seems we need $I(x) = \log \frac{1}{P_X(x)}$ 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

$$H(X) = - \sum_{x \in \mathcal{X}} P_X(x) \log P_X(x) .$$

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.

Jensen's Inequality

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

$$\lambda \cdot f(x) + (1 - \lambda)f(y) \leq f(\lambda \cdot x + (1 - \lambda)y) \text{ ,}$$

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

Jensen's Inequality

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

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

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

$$\mathbb{E}[f(X)] \leq f(\mathbb{E}[X]) \ ,$$

with equality iff X is constant.

Jensen's Inequality

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

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

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

$$\mathbb{E}[f(X)] \leq f(\mathbb{E}[X]) \ ,$$

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 neighboring leaves $E(x)$ and $E(y)$ by (xy) with length $l_x - 1$. Repeat.

\Leftarrow Given lengths $l_{x_1} \leq l_{x_2} \leq \dots \leq l_{x_n}$ 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

$$H(X) \leq E[l(X)] < H(X) + 1 .$$

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

$$H(X) \leq E[l(X)] < H(X) + 1 .$$

Proof of Upper Bound.

Define $l_x = \lceil -\log P_X(x) \rceil$. Then we have

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

Kraft's inequality implies that there is a code with codeword lengths l_x . Then note that

$$\sum_{x \in \Sigma} P_X(x) \lceil -\log P_X(x) \rceil < H(X) + 1.$$

Binary Source Coding Theorem (2/2)

Proof of Lower Bound.

$$\begin{aligned} \mathbb{E}[l(X)] &= \sum_x P_X(x) l(x) \\ &= - \sum_x P_X(x) \log 2^{-l(x)} \\ &\geq - \sum_x P_X(x) \log P_X(x) \\ &= H(X) \end{aligned}$$

Huffman's Code (1/2)

Input: $\{(a_1, p_1), \dots, (a_n, p_n)\}$.

Output: 0/1-labeled rooted tree.

HUFFMAN($\{(a_1, p_1), \dots, (a_n, p_n)\}$)

- (1) $S \leftarrow \{(a_1, p_1, a_1), \dots, (a_n, p_n, a_n)\}$
- (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) $S \leftarrow S \cup \{(b_i \| b_j, p_i + p_j, \text{NODE}(t_i, t_j))\}$
- (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.

Conditional Entropy

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

$$H(X|y) = - \sum_x P_{X|Y}(x|y) \log P_{X|Y}(x|y) \quad \text{and}$$
$$H(X|Y) = \sum_y P_Y(y) H(X|y)$$

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

Properties of Entropy

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

Upper Bound. $H(X) = \mathbb{E}[-\log P_X(X)] \leq \log |\mathcal{X}|$.

Chain Rule and Conditioning.

$$\begin{aligned} H(X, Y) &= - \sum_{x,y} P_{X,Y}(x,y) \log P_{X,Y}(x,y) \\ &= - \sum_{x,y} P_{X,Y}(x,y) (\log P_Y(y) + \log P_{X|Y}(x|y)) \\ &= - \sum_y P_Y(y) \log P_Y(y) - \sum_{x,y} P_{X,Y}(x,y) \log P_{X|Y}(x|y) \\ &= H(Y) + H(X|Y) \leq H(Y) + H(X) \end{aligned}$$

Spurious Keys

We consider how many ciphertexts a computationally unbounded ciphertext-only attacker needs to see to determine the secret key.

Definition.

- ▶ $\mathcal{K}(c) = \{k \in \mathcal{K} \mid \exists m \in \mathcal{M} : E_k(m) = c\}$ is the set of keys consistent with the ciphertext c .
- ▶ $S(c, k) = \mathcal{K}(c) - k$ is the set of **spurious keys**, where k is the correct key.
- ▶ $s_n = \mathbb{E} [|S(C, k)|] = \mathbb{E}[\mathcal{K}(C)] - 1$ is the expected number of spurious keys, over a random choice of ciphertext with n symbols.

What Does a Ciphertext Tell Us About the Key?

Theorem.

$$\begin{aligned}H(C) + H(K|C) &= H(C) + H(K|C) + H(P|C, K) \\ &= H(C, K, P) \\ &= H(K) + H(P|K) + H(C|K, P) \\ &= H(K) + H(P)\end{aligned}$$

Entropy of Natural Language

Definition.

- ▶ P_n is a random $n \times |\mathcal{M}|$ -letter substring of English text.
- ▶ The **entropy of English** is $H_L = \lim_{n \rightarrow \infty} \frac{1}{n} H(P_n) \approx 1.25$.
- ▶ The **relative redundancy of English** is $R_L = 1 - \frac{H_L}{\log |\mathcal{M}|}$.

Number of Spurious Keys

Lemma. $H(K|C_n) \approx H(K) - nR_L \log |\mathcal{M}|$

Proof.

$$\begin{aligned} H(K|C_n) &= H(K) + H(P_n) - H(C_n) \\ &\gtrsim H(K) + nH_L - n \log |\mathcal{C}| \\ &= H(K) + n(1 - R_L) \log |\mathcal{M}| - n \log |\mathcal{C}| . \end{aligned}$$

Number of Spurious Keys

Lemma. $H(K|C_n) \approx H(K) - nR_L \log |\mathcal{M}|$

Lemma. $H(K|C_n) \leq \log(s_n + 1)$

Proof.

$$\begin{aligned} H(K|C_n) &= \sum_{c_n} P_{C_n}(c_n) H(K|c_n) \\ &\leq \mathbb{E} [\log |\mathcal{K}(C_n)|] \\ &\leq \log \mathbb{E} [|\mathcal{K}(C_n)|] \\ &= \log(s_n + 1) \end{aligned}$$

Number of Spurious Keys

Lemma. $H(K|C_n) \approx H(K) - nR_L \log |\mathcal{M}|$

Lemma. $H(K|C_n) \leq \log(s_n + 1)$

Corollary. $\log(s_n + 1) \geq H(K) - nR_L \log |\mathcal{M}|$ or restated:

$$s_n \gtrsim \frac{2^{H(K)}}{|\mathcal{M}|^{nR_L}}$$

Unicity Distance

Definition. The **unicity distance** is defined as $\min\{n \in \mathbb{N} \mid s_n \leq 0\}$, i.e. the smallest number of ciphertext blocks needed until we expect the key to be uniquely defined.