Information Theory

Information theory concerns the measure of information contained in data. The security of a cryptosystem is determined by the relative content of the key, plaintext, and ciphertext.

In this course we consider a *language* $\mathcal{L}$ to be a finite set which will represent the space of keys, of plaintext, or of ciphertext, together with a probability function on $\mathcal{L}$. Although we initially assume that $\mathcal{L}$ is finite, this is not strictly necessary, as we will see later. We refer to an element X of $\mathcal{L}$ as a *message*. As part of the language, we assume the existence of a probability function $P : \mathcal{L} \to \mathbb{R}$, defined to be a non-negative real-valued function on $\mathcal{L}$ such that

$$\sum_{X \in \mathcal{L}} P(X) = 1.$$

For a naturally occuring language, we reduce to a finite model of it by considering finite sets consisting of strings of length N in that language. If $\mathcal{L}$ models the English language, then the function P assigns to each string the probability of its appearance, among all strings of length N, in the English language.

Entropy

The *entropy* of a given language with probability measure P is a measure of the information content of the language. The formal definition of entropy is

$$H(\mathcal{L}) = \sum_{X \in \mathcal{L}} P(X) \log_2(P(X)^{-1}).$$

For $0 < P(X) < 1$, the value $\log_2(P(X)^{-1})$ is a positive real number, and we define $P(X) \log_2(P(X)^{-1}) = 0$ when $P(X) = 0$. The following exercise justifies this definition.

Exercise. Show that the limit

$$\lim_{x \to 0^+} x \log_2(x^{-1}) = 0.$$

What is the maximum value of $x \log_2(x)$ and at what value of x does it occur?

An *optimal encoding* for a language $\mathcal{L}$ is an injective map from the language to strings over some alphabet, such that the bit-length of encoded messages is minimized. The term $P(X) \log_2(P(X)^{-1})$ is the expected bit-length for the encoding of the message X in an optimal encoding, if one exists, and the entropy is the expected number of bits on a random message in the language.

As an example, English text files written in 8-bit ASCII can typically be compressed to 40% of the original size without loss of information, since the structure of the language itself encodes the remaining information. The human genome encodes data for producing sequences of 20 different amino acid, each with a triple of letters in the alphabet $\{\mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{G}\}$. The 64 possibile "words" (codons in genetics) includes more than 3-fold redundancy, in

specifying one of these 20 amino acides. Moreover, huge sections of the genome are repeats such as AAAAAA..., whose information can be captured by an expression like $\mathtt{A}^n$. More accurate models for the languages specified by English or by human DNA sequences would permit greater compression rates for messages in these languages.

Example. Let $\mathcal{L}$ be the language $\{\mathtt{A}, \mathtt{B}, \mathtt{C}\}$ of three elements, and assume that $P(\mathtt{A}) = 1/2$, $P(\mathtt{B}) = 1/4$, and $P(\mathtt{C}) = 1/4$. The entropy of the space is then

$$P(\mathtt{A}) \log_2(2) + P(\mathtt{B}) \log_2(4) + P(\mathtt{C}) \log_2(4) = 1.5.$$

An optimal encoding is attained by the encoding of A with 0, B with 10, and C with 11. With this encoding one expects to use an average of 1.5 bits to transmit a message in this encoding.

The following example gives methods by which we might construct models for the English language.

Example. Empirical models for the English language. First, choose a standard encoding — this might be an encoding as strings in the set $\{\mathtt{A}, \ldots, \mathtt{Z}\}$ or as strings in the ASCII alphabet. Next, choose a sample text. The text might be the complete works of Shakespeare, the short story *Black cat* of Edgar Allan Poe, or the U.S. East Coast version of the *New York Times* from 1 January 2000 to 31 January 2000. The following are finite languages based on these choices:

1. Let $\mathcal{L}$ be the set of characters of the encoding and set $P(c)$ to be the probability that the character c occurs in the sample text.
2. Let $\mathcal{L}$ be the set of character pairs over the encoding alphabet and set $P(X)$ to be the probability that the pair $X = c_1c_2$ occurs in the sample text.
3. Let $\mathcal{L}$ be the set of words in the sample text, and set $P(X)$ to be the probability that the word X occurs in the sample text.

For each of these we can define an infinite model for the English language based on this constrution.

How well do you think each of these model the English language?

Rate and Redundancy.

Let $\mathcal{L}$ be a finite language. We define the *rate* of $\mathcal{L}$ to be

$$r(\mathcal{L}) = \frac{H(\mathcal{L})}{\log_2(\#\mathcal{L})},$$

and the *redundancy* to be $1 - r_{\mathcal{L}}$. Now suppose we have any encoding ϕ of $\mathcal{L}$ in the random language $\mathcal{A}^*$, i.e. an injective function $\phi : \mathcal{L} \to \mathcal{A}^*$. Then we define the rate of ϕ to be

$$r(\phi) = \lim_{N \to \infty} \frac{H(\mathcal{L}^*_{N,\phi})}{N \log_2(m)},$$

where $\mathcal{L}^*_{N,\phi}$ is the sublanguage of elements of $\mathcal{L}^*$ of length at most N under the encoding ϕ in $\mathcal{A}^*$. We can now formally define an *optimal encoding* ϕ of $\mathcal{L}$ to be any encoding such that $r(\phi) = 1$.

Exercise. Show that the rate of a random language is 1 and that this is the maximal value for any language.

The redundancy in a language derives from the structure of a language such as character frequency distributions, digram frequency distributions (the probabilities of ordered, adjacent character pairs), and more generally N-gram frequency distributions. Global structures of a natural language such as vocabulary and grammar rules determine yet more structure, adding to the redundancy of the language.

Conditional Probability

We would now like to have a concept of conditional probability for cryptosystems. Let E be a cryptosystem, $\mathcal{M}$ a plaintext language, $\mathcal{K}$ a keyspace, and $\mathcal{C}$ a ciphertext language. For a symmetric key system the space of plaintext and ciphertext coincide, but the probability distributions on them may differ in the context of the cryptosystem.

We use P for both the probability function on the plaintext language $\mathcal{M}$ and $\mathcal{K}$. We can now define a probability function on $\mathcal{C}$ relative to the cryptosystem E:

$$P(Y) = \sum_{K \in \mathcal{K}} P(K) \sum_{\substack{X \in \mathcal{M} \\ E_K(X) = Y}} P(X).$$

Exercise. Verify that the above definition gives a probability function on the ciphertext language $\mathcal{C}$.

We can now define $P(X, Y)$, for $X \in \mathcal{M}$ and $Y \in \mathcal{C}$ to be the probability that the pair (X, Y) appears as a plaintext–ciphertext pair. Assuming the independence of plaintext and key spaces, we can define this probability as:

$$P(X, Y) = \sum_{\substack{K \in \mathcal{K} \\ E_K(X) = Y}} P(K)P(X).$$

X and Y are said to be *independent* if $P(X, Y) = P(X)P(Y)$.

Exercise. Verify the equalities:

$$P(Y) = \sum_{X \in \mathcal{M}} P(X, Y), \quad \text{and} \quad P(X) = \sum_{Y \in \mathcal{C}} P(X, Y).$$

For ciphertext Y and plaintext X, we can now define the conditional probability $P(Y|X)$ by

$$P(Y|X) = \begin{cases} \dfrac{P(X, Y)}{P(X)} & \text{if } P(X) \neq 0 \\ \\ 0 & \text{if } P(X) = 0 \end{cases}$$

Conditional Entropy

We can now define the conditional entropy $H(\mathcal{M}|Y)$ of the plaintext space with respect to a given ciphertext $Y \in \mathcal{C}$.

$$H(\mathcal{M}|Y) = \sum_{X \in \mathcal{M}} P(X|Y) \log_2(P(X|Y)^{-1})$$

The conditional entropy $H(\mathcal{M}|\mathcal{C})$ of a cryptosystem (more precisely, of the plaintext with respect to the ciphertext) as an expectation of the individual conditional entropies:

$$H(\mathcal{M}|\mathcal{C}) = \sum_{Y \in \mathcal{C}} P(Y) H(\mathcal{M}|Y)$$

This is sometimes referred to as the *equivocation* of the plaintext language $\mathcal{M}$ with respect to the ciphertext language $\mathcal{C}$.

Perfect Secrecy. A cryptosystem is said to have *perfect secrecy* if the entropy $H(\mathcal{M})$ equals the conditional entropy $H(\mathcal{M}|\mathcal{C})$.

One-time pads

Let $K = k_1 k_2 \dots$ be a key stream of random bits, and let $M = m_1 m_2 \dots$ be the plaintext bits. We define a ciphertext $C = c_1 c_2 \dots$ by

$$c_i = m_i \oplus k_i,$$

where $\oplus$ is the addition operation on bits in $\mathbb{Z}/2\mathbb{Z}$. In the language of computer science, this is the `xor` operator:

$$\begin{aligned} 0 \oplus 0 &= 0, \qquad 1 \oplus 0 = 1, \\ 0 \oplus 1 &= 1, \qquad 1 \oplus 1 = 0. \end{aligned}$$

In general such a cryptosystem is called the *Vernam cipher*. If the keystream bits are generated independently and randomly, then this cipher is called a *one-time pad*.

Perfect secrecy of one-time pads. Recall that $P(X|Y)$ is defined to be $P(X|Y) = P(X,Y)/P(Y)$ if $P(Y) \neq 0$ and is zero otherwise. If $\mathcal{M}$ is the plaintext language and $\mathcal{C}$ the ciphertext language (with probability function defined in terms of the cryptosystem), then the condidtional entropy $H(\mathcal{M}|\mathcal{C})$ is defined to be:

$$\begin{aligned} H(\mathcal{M}|\mathcal{C}) &= \sum_{Y \in \mathcal{C}} P(Y) H(\mathcal{M}|Y) \\ &= \sum_{Y \in \mathcal{C}} P(Y) \sum_{X \in \mathcal{M}} P(X|Y) \log_2(P(X|Y)^{-1}) \\ &= \sum_{Y \in \mathcal{C}} \sum_{X \in \mathcal{M}} P(X,Y) \log_2(P(X|Y)^{-1}). \end{aligned}$$

If for each $X \in \mathcal{M}$ and $Y \in \mathcal{C}$ the joint probability $P(X, Y)$ is equal to $P(X)P(Y)$ (i.e. the plaintext and ciphertext languages are independent) and thus $P(X|Y) = P(X)$, then the above expression simplifies to:

$$\begin{aligned} H(\mathcal{M}|\mathcal{C}) &= \sum_{X \in \mathcal{M}} \sum_{Y \in \mathcal{C}} P(X)P(Y) \log_2(P(X)^{-1}) \\ &= \Big(\sum_{Y \in \mathcal{C}} P(Y) \Big) \sum_{X \in \mathcal{M}} P(X) \log_2(P(X)^{-1}) \\ &= \sum_{X \in \mathcal{M}} P(X) \log_2(P(X)^{-1}) = H(\mathcal{M}). \end{aligned}$$

Therefore the cryptosystem has perfect secrecy.

Exercise. Show that the one-time pad has perfect secrecy.

Note neither the Vernam cipher nor the one-time pad has to be defined with respect to a binary alphabet. The `xor` can be replaced by addition in $\mathbb{Z}/n\mathbb{Z}$, where n is the alphabet size, using any bijection of the alphabet with the set $\{0, \ldots, n-1\}$.

Entropy of Key space. It can be shown that perfect secrecy (or unconditional security) requires the entropy $H(\mathcal{K})$ of the key language $\mathcal{K}$ to be at least as large as the entropy $H(\mathcal{M})$ of the plaintext language $\mathcal{M}$. If the key language is defined to be the set of N-bit strings chosen uniformly at random, then the entropy of $\mathcal{K}$ is N, and this is the maximum entropy for a language of N-bit strings (see exercise). This implies that in order to achieve perfect secrecy, the number of bits of strings in the keyspace should be at least equal the entropy $H(\mathcal{M})$ of the plaintext language.

Exercise. Show that N is the maximum entropy for a language of bit strings of length N (i.e. the maximum for any probability function).