Coding & Cryptography

January 22, 2020

0 Communication Channels

This course will be about modelling communication. In general, we have the following idea:

$$\textsc{Source} \longrightarrow \textsc{Encoder} \xrightarrow[\text{Errors, Noise}]{\textsc{Channel}} \textsc{Decoder} \longrightarrow \textsc{Receiver}$$

For example, the channel might be an optical or electrical telegraph, modems, audio CDs, satellite relays. The encoding and decoding might be something like Ascii, so that each character in the email "Call at 2pm" would be encoded into 8 bits using $a = 1100101, \ldots$, giving an 84 bit message to be transmitted via the internet, and decoded by the receiver's email client. Our general aim here will be, given some source and channel (modelled probabilistically), to design an encoder and decoder to send messages economically and reliably.

Examples

- (Noiseless coding) Morse Code. In this code, more common letters are assigned shorter codes, so that we have $A = \cdot - *, E = \cdot *, Q = - - \cdot - *, Z = - - \cdot \cdot *$. This is adapted to the *source*, in the sense that we chose the codes based off the expected distribution of letters that we will have to transmit.
- (Noisy coding) Isbn. In the Isbn encoding, every book is given a 10 digit number $a_1 a_2 \ldots a_{10}$, with $\sum_{i=1}^{10} (11 - i) a_i \equiv 0 \mod 11$. This is adapted to the *channel*, in the sense that the likely errors to occur will be 1 incorrect digit, or accidentally transposing two digits, which this code is resistant to (will return an error rather than an erroneous result).

A ***communication channel*** accepts symbols from some alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_r\}$ (e.g. $\{0, 1\}, \{a, b, \ldots, z\}$), and outputs symbols from an alphabet $\mathcal{B} = \{b_1, \ldots, b_s\}$. The channel is modelled by the probabilities:

$$\mathbb{P}(y_1, y_2, \ldots, y_n \text{ received} | x_1, x_2, \ldots, x_n \text{ sent}) = \prod_{i=1}^{n} \mathbb{P}(y_i \text{ received} | x_i \text{ sent})$$

A ***discrete memoryless channel (DMC)*** is a channel with $p_{ij} = \mathbb{P}(b_j \text{ received} | a_i \text{ sent})$ the same for each channel usage and independent of any past or future channel usages.

The ***channel matrix*** is $P = (p_{ij})$, an $r \times s$ stochastic matrix.

Examples: The ***binary symmetric channel (BSC)*** with error probability $p \in [0, 1]$ has $\mathcal{A} = \mathcal{B} = \{0, 1\}$. The channel matrix is $\begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$. A symbol is transmitted correctly with probability $1 - p$.

The ***binary erasure channel*** has input alphabet $\{0,1\}$, and output alphabet $\{0,1,*\}$, where we miss a bit is probability p, giving channel matrix $\begin{pmatrix} 1-p & 0 & p \\ 0 & 1-p & 0 \end{pmatrix}$. We can model n uses of a channel by the ***n^{th} extension*** with input alphabet $\mathcal{A}^n$, and output alphabet $\mathcal{B}^n$.

A ***code*** c of ***length*** n is a function $c : M \to \mathcal{A}^n$ where M is the set of all possible messages. Implicitly, we also have a decoding rule $\mathcal{B}^n \to M$.
The ***size*** of c is $m = |M|$.
The ***information rate*** is $\rho(c) = \frac{1}{n}\log_2(m)$.
The ***error rate*** is $\widehat{e}(c) = \max_{x \in M}\{\mathbb{P}(\text{error}|x \text{ sent})\}$.

A channel can transmit reliably at a rate R if there exists a sequence of codes $(c_n : n \geq 1)$ with c_n a code of length n, $\lim_{n\to\infty}(\rho(c_n)) = R, \lim_{n\to\infty}(\widehat{e}(c_n)) = 0$. The capacity of a channel is the supremum of all reliable transmission rates.

Theorem 0.1. *A BSC with error probability $p < \frac{1}{2}$ has a non-zero capacity (i.e. good codes exist).*

Proof. See **9.3** □

1 Noiseless Coding

1.1 Prefix-free Codes

For an alphabet $\mathcal{A}, |\mathcal{A}| < \infty$, let $\mathcal{A}^* = \bigcup_{n\geq 0}\mathcal{A}^n$, the set of all finite strings from $\mathcal{A}$. The ***concatenation*** of strings $x = x_1 \dots x_r$ and $y = y_1 \dots y_s$ is $xy = x_1 \dots x_r y_1 \dots y_s$.

Let $\mathcal{A}, \mathcal{B}$, be alphabets. A ***code*** is a function $c : \mathcal{A} \to \mathcal{B}^*$. The strings $c(a)$ for $a \in \mathcal{A}$ are called ***codewords*** (cws). If $x, y \in \mathcal{B}^*$ then x is a ***prefix*** of y if $y = xz$ for some $z \in \mathcal{B}^*$.

For example, we have the Greek fire code, found in the writings of Polybius around 280 BC. $\mathcal{A} = \{\alpha, \beta, \dots, \omega\}, \mathcal{B} = \{1,2,3,4,5\}$, with code $\alpha \mapsto 11, \beta \mapsto 12, \dots, \psi \mapsto 53, \omega \mapsto, 54$, where xy means "x torches held up, and another y torches nearby".

The English language is even a code: we can let $\mathcal{A}$ be words in a given dictionary, and $\mathcal{B} = \{a, b, \dots, z, \sqcup\}$, where the coding function is to spell the word and follow it with a space.

We send a message $x_1 \dots x_n \in \mathcal{A}^*$ as $c(x_1) \dots c(x_n) \in \mathcal{B}^*$. So c extends to a function $c^* : \mathcal{A}^* \to \mathcal{B}^*$.

c is ***decipherable/decidable*** if c^* is injective, so that each string in $\mathcal{B}^*$ could have come from at most one message. Note that it isn't sufficient to just have c injective, although clearly this is necessary:

$\mathcal{A} = \{1,2,3,4\}, \mathcal{B} = \{0,1\}, c : 1 \mapsto 0, 2 \mapsto 1, 3 \mapsto 00, 4 \mapsto 01$. Then $c^*(114) = 0001 = c^*(312)$.

If $|\mathcal{A}| = m, |\mathcal{B}| = a$, then we say c is an ***a-ary code of size m***. 2-ary = ***binary***, 3-ary=***ternary***.

We aim to construct decipherable codes with short word lengths. Assuming c is injective, the following are always decipherable:

- Block codes, where every codeword has the same length (e.g. Greek fire, ASCII)

- Comma codes, where we have an "end of word" character (e.g. English language)
- Prefix-free codes, where no codeword is a prefix of any other distinct words.

Note that both of the first two are special cases of prefix-free codes. Prefix-free codes are often called ***instantaneous*** or ***self-punctuating*** codes. Note that not all decipherable codes are prefix-free: $0 \mapsto 01, 1 \mapsto 011$ is decipherable but not prefix free.

Theorem 1.1 (Kraft's Inequality)**.** *Let $|\mathcal{A}| = m, |\mathcal{B}| = a$. A prefix-free code $c : \mathcal{A} \to \mathcal{B}^*$ with word lengths $\ell_1, \ldots, \ell_m$ exists if and only if:*

$$\sum_{i=1}^{m} a^{-\ell_i} \leq 1 \tag{$*$}$$

Proof. Rewrite $(*)$ as $\sum_{\ell=1}^{s} n_\ell a^{-\ell} \leq 1 (\star)$, where n_ℓ is the number of codewords of length ℓ and $s = \max_{1 \leq i \leq m} \ell_i$.

$\implies$ If $c : \mathcal{A} \to \mathcal{B}^*$ is prefix-free, then $n_1 a^{s-1} + n_2 a^{s-2} + \ldots n_s \leq a^s$, since the LHS is the number of strings of length s in $\mathcal{B}$ with some codeword of c as a prefix, and RHS is the number of strings of length s. Dividing by a^s gives $(\star)$.

$\impliedby$ Given $n_1, \ldots, n_s$ satisfying $(\star)$, we need to construct a prefix-free code c with n_ℓ codewords of length ℓ for all $\ell \leq s$. We use induction on s. The case $s = 1$ is clear: we have $(\star)$ gives $n_1 \leq a$, so we can choose a code.

By the induction hypothesis there is a prefix-free code $\widehat{c}$ with n_ℓ codewords of length ℓ for all $\ell \leq s - 1$. Then $(\star)$ gives:

$$n_1 a^{s-1} + n_2 a^{s-2} + \ldots + n_{s-1} a + n_s \leq a^s$$

where the first $s - 1$ terms on LHS sum to the number of strings of length s with some codeword of $\widehat{c}$ as a prefix, and the RHS is the number of strings of length s. Hence we can add at least n_s new codewords of length s to $\widehat{c}$ and maintain the prefix-free property, giving our code.

□

Theorem 1.2 (McMillan)**.** *Any decipherable code satisfies Kraft's inequality*

Karush. Let $c : \mathcal{A} \to \mathcal{B}^*$ be a decipherable code with codewords of lengths $\ell_1, \ldots, \ell_m$. Let $s = \max_{1 \leq i \leq m} \ell_i$. Then for $R \in \mathbb{N}$:

$$\left(\sum_{i=1}^{m} a^{-\ell_i} \right)^R = \sum_{l=1}^{Rs} b_\ell a^{-\ell}$$

where $b_\ell = |\{x \in \mathcal{A}^R : c^*(x) \text{ has length } \ell\}| \leq |\mathcal{B}^\ell| = a^\ell$, using the fact that c^* is injective. Then:

$$\left(\sum_{i=1}^{m} a^{-\ell_i} \right)^R \leq \sum_{l=1}^{R} a^\ell a^{-\ell} = Rs$$

$$\sum_{i=1}^{m} a^{-\ell_i} \leq (Rs)^{\frac{1}{R}} \to 1 \text{ as } R \to \infty$$

□

Corollary 1.3. *A decipherable code with prescribed word lengths exists iff a prefix-free code with same word lengths exists.*

Proof.

$\Longrightarrow$ Use **1.2** to generate a prefix-free code by **1.1**

$\Longleftarrow$ Prefix-free codes are decipherable.

□

2 Shannon's Noiseless Coding Theorem

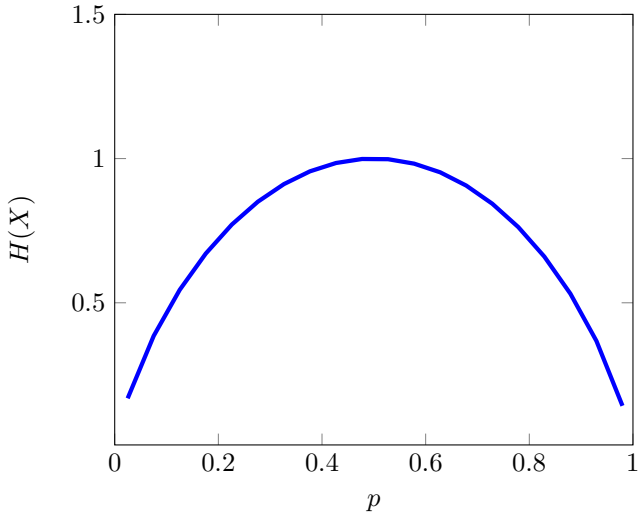
Entropy is a measure of 'randomness' or 'uncertainty'. Suppose we have a random variable X that takes values $x_1, \ldots, x_n$ with probabilities $p_1, \ldots, p_n$. Then the ***entropy*** (roughly speaking) is the expected number of fair coin tosses needed to simulate X.

Examples:

- $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$. We can identify $\{x_1, x_2, x_3, x_4\}$ with $\{HH, HT, TH, TT\}$, and so the entropy of this random variable is 2.
- $(p_1, p_2, p_3, p_4) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$. Here, the entropy is $1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = \frac{7}{4}$. We might say then, since the entropy is greater, that the first example is "more random" than the second.

More concretely, the ***Shannon (or information) entropy*** of X is $H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$. Note that $H(X) \geq 0$ with equality if and only if $\mathbb{P}(X = x_i) = 1$. This is measured in ***bits***. We take the convention that $0 \log 0 = 0$.

Example: Consider a biased coin, where $\mathbb{P}(X = H) = p, \mathbb{P}(X = T) = 1 - p$. Then $H(X) = -p \log p - (1-p) \log(1-p) = p \log(\frac{1-p}{p} - \log(1-p)$.



Proposition 2.1 (Gibb's Inequality)**.** *Let* $(p_1, \ldots, p_n)$ *and* $(q_1, \ldots, q_n)$ *be probability distributions. Then:*

$$-\sum_{i=1}^{n} p_i \log p_i \leq -\sum_{i=1}^{n} p_i \log q_i$$

with equality if and only if $p_i = q_i$ *for all* i*.*

Proof. Since $\log x = \frac{\ln x}{\ln 2}$, we may replace log by ln in the proof. Put $I = \{1 \leq r \leq n : p_i \neq 0\}$. Now $\ln x \leq x - 1$ with equality if and only if $x = 1$. So we have $\ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1$, and hence:

$$\begin{aligned}
\sum_{i \in I} p_i \ln \frac{q_i}{p_i} &\leq \sum_{i \in I} q_i - \sum_{i \in I} p_i \\
&= \sum_{i \in I} q_i - 1 \leq 0 \\
\therefore -\sum_{i \in I} p_i \ln p_i &\leq -\sum_{i \in I} p_i \ln q_i \\
\therefore -\sum_{i=1}^{n} p_i \log p_i &\leq -\sum_{i=1}^{n} p_i \log q_i
\end{aligned}$$

If equality holds, then $\sum_{i \in I} p_i = 1$ and $\frac{p_i}{q_i} = 1$ for all $i \in I$, so $p_i = q_i$ □

Corollary 2.2. $H(p_1, \ldots, p_n) \leq \log n$ *with equality if and only if* $p_1 = \ldots = p_n = \frac{1}{n}$*.*

Proof. Take $q_1 = \ldots = q_n = \frac{1}{n}$ in **2.1**. □

Let $\mathcal{A} = \{\mu_1, \ldots, \mu_m\}$, and $|\mathcal{B}| = a$, where $m, a \geq 2$. The random variable X takes values $\mu_1, \ldots, \mu_m$ with probabilities $p_1, \ldots, p_m$. We say a code $c : \mathcal{A} \to \mathcal{B}^*$ is ***optimal*** if it is a decipherable code with smallest possible expected word length, $\mathbb{E}S = \sum_i p_i \ell_i$.

Theorem 2.3 (Shannon's Noiseless Coding Theorem)**.** *The expected word length* $\mathbb{E}S$ *of an optimal code satisfies:*

$$\frac{H(X)}{\log a} \leq \mathbb{E}S < \frac{H(X)}{\log a} + 1$$

Proof. For the lower bound, take $c : \mathcal{A} \to \mathcal{B}^*$ decipherable with word lengths $\ell_1, \ldots, \ell_m$. Then set $q_i = \frac{a^{-\ell_i}}{D}$ where $D = \sum_{i=1}^{m} a^{-\ell_i}$. Now we have that $\sum_{i=1}^{m} q_i = 1$. By Gibbs,

$$\begin{aligned}
H(X) &\leq -\sum_{i=1}^{m} p_i \log q_i \\
&= -\sum p_i \left(-\ell_i \log a - \log D\right) \\
&= \left(\sum_{i=1}^{m} p_i \ell_i\right) \log a + \log D
\end{aligned}$$

By McMillan, $D \leq 1$, so $\log D \leq 0$, and so $H(X) \leq (\sum_{i=1}^{m} p_i \ell_i) \log a = \mathbb{E}S \cdot \log a$, and we have equality if and only if $p_i = a^{-\ell_i}$ for some integers $\ell_1, \ldots \ell_m$.

For the upper bound, take $\ell_i = \lceil -\log_a p_i \rceil$. Then $-\log_a p_i \le \ell_i \implies p_i \ge a^{-\ell_i}$.

Now $\sum_{i=1}^m a^{-\ell_i} \le \sum_{i=1}^m p_i = 1$. By Kraft, there is some prefix-free code c with word lengths $\ell_1, \ldots, \ell_m$, and the expected word length of c is $\mathbb{E}S = \sum p_i \ell_i < \sum p_i(-\log_a p_i + 1) = \frac{H(X)}{\log a} + 1$. $\square$

Example: ***Shannon-Fano coding***
We mimic the above proof: given probabilities $p_1, \ldots, p_n$, set $\ell_i = \lceil -\log_a p_i \rceil$. Construct the prefix-free code with word lengths $\ell_1, \ldots, \ell_m$ by choosing in order of increasing length, ensuring that previous codewords are not prefixes. For example, if $a = 2, m = 5$ we have:

i	p_i	$\lceil -\log_2 p_i \rceil$	Codewords
1	0.4	2	00
2	0.2	3	010
3	0.2	3	011
4	0.1	4	1000
5	0.1	4	1001

$\mathbb{E}S = \sum p_i \ell_i = 2.8$, entropy $= 2.12$