Extensions of Channels

The Nth extension of a channel consists of N independent copies of the channel, replicated in space or time.

The input alphabet for this extension, $\mathcal{A}_X$, consists of N-tuples $(a_{i_1}, \ldots, a_{i_N})$. The output alphabet, $\mathcal{A}_Y$, consists of N-tuples $(b_{j_1}, \ldots, b_{j_N})$

Assuming the N copies don't interact, the transition probabilities for the extension are

$$Q_{j_1,\ldots,j_N|i_1,\ldots,i_N} \;=\; Q_{j_1|i_1} \cdots Q_{j_N|i_N}$$

If we use input probabilities of

$$p_{i_1,\ldots,i_N} \;=\; p_{i_1} \cdots p_{i_N}$$

it is easy to show that the input and output entropies, the conditional entropies, and the mutual information are all N times those of the original channel.

Capacity of the Extension

We can maximize mutual information for the extension by using an input distribution in which

- each symbol is independent
- each symbol has the distribution that maximizes mutual information for the original channel.

It follows that the capacity of the extension is N times the capacity of the original channel.

But treating the N copies independently is uninteresting — we gain nothing over the original channel.

The strategy: We *don't* chose the input distribution to maximize mutual information, but rather use one that is *almost as good*, and which lets us *correct almost all errors*.

Codes for the Extension

A *code*, $\mathcal{C}$, for the Nth extension is a subset of the set of all possible blocks of N input symbols — ie, $\mathcal{C} \subseteq \mathcal{A}_X^N$.

The elements of $\mathcal{C}$ are called the *codewords*. These are the only blocks that we transmit.

For example, one code for the third extension of a BSC is the "repetition code", in which there are two codewords, 000 and 111.

The Nth extension together with the code can be seen as a channel with $|\mathcal{C}|$ input symbols and $|\mathcal{A}_Y|^N$ output symbols.

Decoding to a Codeword

When the sender transmits a codeword in $\mathcal{C}$, the receiver might (in general) see any output block, $b_{j_1} \cdots b_{j_N} \in \mathcal{A}_Y^N$.

The receiver can try to *decode* this output in order to recover the codeword that was sent.

The optimal method of decoding is to choose a codeword, $w \in \mathcal{C}$, which maximizes

$$P(w \,|\, b_{j_1} \cdots b_{j_N}) \;=\; \frac{P(w)\, P(b_{j_1} \cdots b_{j_N} \,|\, w)}{P(b_{j_1} \cdots b_{j_N})}$$

In case of a tie, we pick one of the best w arbitrarily.

If $P(w)$ is the same for all $w \in \mathcal{C}$, this scheme is equivalent to choosing w to maximize the "likelihood", $P(b_{j_1} \cdots b_{j_N} \,|\, w)$.

Example: A Repetition Code

Suppose we use the three-symbol repetition code for a BSC with $f = 0.1$. Assume that the probability of 000 being sent is 0.7 and the probability of 111 being sent is 0.3.

What codeword should the decoder guess if the received symbols are 101?

$$\begin{aligned} & P(w = 000 \,|\, b_1 = 1, b_2 = 0, b_3 = 1) \\ & \quad = \frac{P(w = 000)\, P(b_1 = 1, b_2 = 0, b_3 = 1 \,|\, w = 000)}{P(b_1 = 1, b_2 = 0, b_3 = 1)} \\ & \quad = \frac{0.7 \times 0.1 \times 0.9 \times 0.1}{0.7 \times 0.1 \times 0.9 \times 0.1 + 0.3 \times 0.9 \times 0.1 \times 0.9} \\ & \quad = 0.206 \end{aligned}$$

$$\begin{aligned} & P(w = 111 \,|\, b_1 = 1, b_2 = 0, b_3 = 1) \\ & \quad = \frac{P(w = 111)\, P(b_1 = 1, b_2 = 0, b_3 = 1 \,|\, w = 000)}{P(b_1 = 1, b_2 = 0, b_3 = 1)} \\ & \quad = \frac{0.3 \times 0.9 \times 0.1 \times 0.9}{0.7 \times 0.1 \times 0.9 \times 0.1 + 0.3 \times 0.9 \times 0.1 \times 0.9} \\ & \quad = 0.794 \end{aligned}$$

The decoder should guess that 111 was sent.

Associating Codewords with Messages

Suppose our original message is a sequence of K bits. (Or we might break our message up into K-bit blocks.)

If we use a code with 2^K codewords, we can send this message (or block) as follows:

- The encoder maps the block of K message symbols to a codeword.
- The encoder transmits this codeword.
- The decoder guesses at the codeword sent.
- The decoder maps the guessed codeword back to a block of K message symbols.

We hope the block of decoded message symbols is the same as the original block!

Example: To send binary messages through a BSC with the repetition code, we use blocks of size one, and the map $0 \leftrightarrow 000$, $1 \leftrightarrow 111$.

Decoding for a BSC By Maximum Likelihood

For a BSC, if all codewords are equally likely, the optimal decoding is the codeword differing in the fewest bits from what was received.

The number of bits where two bit sequences, u and v, differ is called the *Hamming distance*, written $d(u,v)$. Example: $d(00110, 01101) = 3$.

The probability that a codeword w of length N will be received as a block b through a BSC with error probability f is

$$(1-f)^{N-d(w,b)}\, f^{\,d(w,b)} \;=\; (1-f)^N \left(\frac{f}{1-f}\right)^{d(w,b)}$$

If $f < 1/2$, and hence $f/(1-f) < 1$, choosing w to maximize this likelihood is the same as choosing w to minimize the Hamming distance between w and b.

An Example Code for the BSC

Here's a code with four 5-bit codewords:

00000, 00111, 11001, 11110

We can map between 2-bit blocks of message bits and these codewords as follows:

$$\begin{array}{llllll} 00 & \leftrightarrow & 00000 & 01 & \leftrightarrow & 00111 \\ 10 & \leftrightarrow & 11001 & 11 & \leftrightarrow & 11110 \end{array}$$

Suppose the sender encodes the message block 01 as 00111 and transmits it, and the receiver then sees the output 00101.

How should this be decoded? We look at the Hamming distances to each codeword:

$$\begin{array}{ll} d(00000, 00101) = 2 & d(00111, 00101) = 1 \\ d(11001, 00101) = 3 & d(11110, 00101) = 4 \end{array}$$

The decoder therefore picks the codeword 00111, corresponding to the message block 01.

The Rate of a Code

A code $\mathcal{C}$ with $|\mathcal{C}|$ binary codewords of length N, is said to have *rate*

$$R = \frac{\log_2 |\mathcal{C}|}{N}$$

Suppose we are sending binary messages through a binary channel, using a code with 2^K codewords of length N. Then the rate will be

$$R = K/N$$

For example, if we encode message blocks of 100 bits into codewords consisting of 300 bits, the rate will be 1/3.

A Preview of the Noisy Coding Theorem

Shannon's noisy coding theorem states that:

> For any channel with capacity C, any desired error probability, $\epsilon > 0$, and any transmission rate, $R < C$, there exists a code with some length N having rate at least R such that the probability of error when decoding this code by maximum likelihood is less than ϵ.

In other words: We can transmit at a rate arbitrarily close to the channel capacity with arbitrarily small probability of error.

The converse is also true: We *cannot* transmit with arbitrarily small error probability at a rate greater than the channel capacity.

Why We Can't Use a BSC Beyond Capacity With Vanishing Error

If we could transmit through a BSC beyond the capacity $C = 1 - H_2(f)$, with vanishingly small error probability, we could compress data to less than its entropy.

Here's how we would do it:

- Divide the data into two blocks: x is of length K and has bit probabilities of 1/2, y is of length N and has bit probabilities of f and $1-f$. The total information in x and y is $K + NH_2(f)$.
- Encode x in a codeword w of length N, and compute $z = w + y$, with addition modulo 2. This z is the compressed form of the data.
- Apply the decoding algorithm to recover w from z, treating y as noise. We can then recover x from w and also $y = z - w$.

Continuing...

We can handle a small number of errors by checking where they would occur and transmitting extra bits needed to identify corrections. This adds only a few more bits to the compressed form of the data.

The result: We compressed a source with entropy $K + NH_2(f))$ into only slightly more than N bits.

This is possible only if $N \geq K + NH_2(f)$, which implies that $R = K/N \leq 1 - H_2(f) = C$.