Example of Error Correction: A Repetition Code

Suppose we're trying to send a series of bits through a channel that sometimes randomly changes a bit from 0 to 1 or vice versa.

One way to improve reliability despite this "noise" is to send each bit *three times*.

Eg, if we want to send the bit sequence 01101, we actually send 000111111000111.

The receiver looks at the bits in groups of three, and decodes each group to the bit that occurs most often in the group.

An example in which the correct message is decoded despite four transmission errors:

$$\begin{array}{lll} & & \text{messsage transmitted} \\ 01101 & \rightarrow & 000111111000111 \\ & \rightarrow & 010111011100101 \rightarrow 01101 \\ & & \text{noisy message received} \end{array}$$

How Much Does This Repetition Code Improve Reliability?

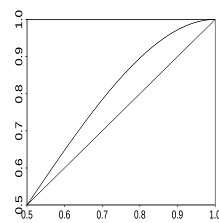
Repeating each bit three times allows us to correct *one error* in each group of three bits, but not more errors.

Suppose each bit has probability P of being received correctly, independently for each bit. The probability that a group of three repeated bits will be decoded correctly is

$$\Pr(0 \text{ errors}) + \Pr(1 \text{ error}) \;=\; P^3 + 3P^2(1-P)$$

Here's a plot of this versus P:

Probability of correctly decoding a group



Probability of a bit being received correctly

Repeating More Times

Suppose we repeat an odd number, n, times. We'll be able to decode correctly as long as there are less than $n/2$ errors in these bits.

As n gets bigger, the decoding error probability goes down. Here's what happens with a 10% probability of transmission error (ie, $P=0.9$):

Number of repetitions	Probability of incorrect decoding
3	0.028
5	0.0086
7	0.0027
9	0.00089
11	0.00030
13	0.000099
15	0.000034

It seems that we can push the error probability as close to zero as we wish by using more repetitions.

But What's the Cost?

Unfortunately, though we can push the error probability to zero, the *rate* at which we transmit useful information goes to zero at the same time!

Fortunately, there are cleverer schemes than repetition codes.

Unfortunately, the simple schemes (and many not-so-simple ones) still have this property — reducing the error probability to zero requires reducing the information rate to zero too.

Is this inevitable?

The Surprising Answer

Shannon proved in 1948 that we actually *can* transmit with arbitrarily small error at any rate up to the *capacity* of the channel.

For the example with $P = 0.9$, the capacity turns out to be 0.531. So we should be able to obtain nearly zero decoding errors while transmitting only twice as many bits as are in the message (rate 0.5).

One catch: To get lower and lower error rates, we need to transmit using bigger and bigger *blocks*.

Eg, for rate 0.5, we encode a block of $n/2$ bits into n bits. Obtaining a very low decoding error probability will require that n be quite big.

Can We Do This in Practice?

Practical methods for transmitting at close to capacity have been developed only in the last few years.

I tried the example with $P = 0.9$ (ie, 10% transmission errors) using a simple type of "low-density parity-check code".

I used blocks of 4000 message bits, encoded into 10000 transmitted bits (a rate of 0.4).

In a test transmitting 1000 such blocks, there were no errors in the decoded messages.

Error-Correction Topics We'll Cover

- Information channels, particularly the *Binary Symmetric Channel*. (Jones & Jones, Sec. 4.1–4.2)
- General aspects of error-correcting codes. (Jones & Jones, Sec. 5.1–5.3, 6.1–6.3)
- Properties of *linear codes*, and some practical examples. (Jones & Jones, Ch. 7)
- **Shannon's noisy coding theorem:** Information can be transmitted with *arbitrarily small probability of error* at any rate below the *capacity* of the channel. (Jones & Jones, Sec. 4.3–4.8, 5.4–5.6)
- A brief look at "low-density parity-check codes", which come close to achieving this theoretical promise in practice. (Class notes)