Error-Correcting Codes

Error-correcting codes are part of Information Theory, a branch of mathematics with an identifiable founder. Claude Shannon accomplished this when he published a paper titled "A Mathematical Theory of Communication." Laymen at the time (1948) must have wondered how one could mathematicise communication, which many considered the process of staying on speaking terms with relatives they despised.

Today we are used to bits, bytes and whatnot. We understand a low-quality phone line has a low bit rate, useful only and especially for speaking to certain relatives.

Once launched, information theory led to mathematical techniques for encryption, error correction and data compression. I dealt with encryption in Public Key Encryption I and II. Here I deal with error correction. Information Theory as a whole is too advanced for this book.

Your electronics would not work without error correction. Disk drives, SSDs and other flash memory with their microscopic bits are not reliable enough for writing and later faithfully retrieving your data. All modern storage includes heavy duty error correction to attain the desired reliability, not that this is advertised. "We correct 100% of our errors!" is not a slogan likely to win customers.

The idea of error correction is simple. When we have a message to send over a noisy communication channel (all channels have some noise), we send extra information so if the channel changes something the original message can still be recovered.

Suppose we want to send a single bit, either 1 or 0, perhaps standing by agreement for Yes and No, respectively. Because the channel might change the bit, converting 0 to 1 or 1 to 0, we can't be certain of the accuracy of what is received. Instead, let's send either 000 or 111. Even if any single bit is changed, by looking at the majority of received bits in, say, 011, it can be concluded the original message was 111 not 000. Sending redundant information is the key, but how do we minimize the redundant information and not overload the channel?

Here is the world's first error correcting code. A message is a series of 4-bit sequences, each representing a number in the range 0-15. Instead of tripling each of the four bits as above for a total of twelve bits, we send only seven bits according to this table.

4-Bit Sequence		What we send
0	0000	0000000
1	0001	1101001
2	0010	0101010
3	0011	1000011
4	0100	1001100
5	0101	0100101
6	0110	1100110
7	0111	0001111
8	1000	1110000
9	1001	0011001
10	1010	1011010
11	1011	0110011
12	1100	0111100
13	1101	1010101
14	1110	0010110
15	1111	1111111

Why these bit patterns? Compare any two of them and you will see they differ in at least three places. If the noisy communication channel changes any one of the seven bits, we can find what the original pattern of seven bits had to be. This corrects any one error. By adding more bits, we can correct any two errors and can carry this as far as we like. If a noisy channel corrupts any combination of up to 19 bits out of 20, we can design a code to correct all these errors.

Each seven-bit codeword is at the center of a sphere in a mathematical space and the spheres do not overlap. A corruption of any one bit moves us

from the center of the sphere to another point within that same sphere. Because the spheres do not overlap the closest center must be the original message. Finding a new code involves finding a new set of non-overlapping spheres in some high-dimensional space. The bigger the spheres, the more errors that can be corrected.

If you ask a mathematician about his work and he obscurely answers (this has happened to me), “I work on sphere packing in N dimensions,” you should immediately reply, “You mean you work on error-correcting codes. Why didn’t you just say so?”