

Introductory Concepts

In the beginning ...

Coding theory originated with the advent of computers. Early computers were huge mechanical monsters whose reliability was low compared to the computers of today. Based, as they were, on banks of mechanical relays, if a single relay failed to close the entire calculation was in error. The engineers of the day devised ways to detect faulty relays so that they could be replaced. While R.W. Hamming was working for Bell Labs, out of frustration with the behemoth he was working with, the thought occurred that if the machine was capable of knowing it was in error, wasn't it also possible for the machine to correct that error. Setting to work on this problem Hamming devised a way of encoding information so that if an error was detected it could also be corrected. Based in part on this work, Claude Shannon developed the theoretical framework for the science of coding theory.

Bell Labs

Quoted from *From Error-Correcting Codes through Sphere Packings to Simple Groups*, pp.16-17.

The Model V, the penultimate relay computer to be built by Bell, was the largest in the series. Containing over nine thousand relays and over fifty pieces of teletype apparatus, it occupied about one thousand square feet of floor space and weighed some ten tons. Incidentally, the same computing power three decades later is to be found in the more sophisticated hand-held calculators. (Today, in most common wristwatches.)

Bell Labs

Quoted from *From Error-Correcting Codes through Sphere Packings to Simple Groups*, pp.16-17.

... the Model V ... used various k-out-of-six codes, particularly for input and output. These were transmitted essentially in blocks and permitted extensive self-checking. For example, the input was entered in the machine via a punched paper tape, seven-eighths of an inch wide, which had up to six holes per row. Each row was read as a unit. If the sensing relays expected a two-out-of-six code, they would prevent further computation if more or less than two holes appeared in a given row. Similar checks were used in nearly every step of a computation.

Bell Labs

Quoted from *From Error-Correcting Codes through Sphere Packings to Simple Groups*, pp.16-17.

When such a check failed, two results were possible depending upon whether the machine was set for “daytime” use with operating personnel present or for unattended “nighttime” or “weekend” operation. In the first mode, a check failure stopped the machine and sounded an alarm. In the second, a check failure switched the machine immediately to other work. In the first case, the check failure was located by operating personnel, whose efforts were facilitated by an elaborate check light panel. However, in the second case the problem simply had to be rerun.

Bell Labs

Quoted from *From Error-Correcting Codes through Sphere Packings to Simple Groups*, pp.16-17.

Hamming did not have priority use of the Model V. In fact, he had only weekend access, and that only when another department wasn't using it. “When they didn't need it, I got it.” Furthermore, for weekend use the machine was placed in the unattended mode ... Of this unfortunate circumstance Hamming said:

Two weekends in a row I came in and found that all my stuff had been dumped and nothing was done. I was really aroused and annoyed because I wanted those answers and two weekends had been lost. And so I said, 'Damn it, if the machine can detect an error, why can't it locate the position of the error and correct it?'

The Coding Idea

What we have called Coding Theory, should more properly be called the Theory of Error-Correcting Codes, since there is another aspect of Coding Theory which is older and deals with the creation and decoding of secret messages. This field is called Cryptography and we will not be interested in it. Rather, the problem that we wish to address deals with the difficulties inherent with the transmission of messages. More particularly, suppose that we wished to transmit a message and knew that in the process of transmission there would be some altering of the message, due to weak signals, sporadic electrical bursts and other naturally occurring noise that creeps into the transmission medium. The problem is to insure that the intended message is obtainable from whatever is actually received.

The Repeat Code

One simple approach to this problem is what is called a repeat code. For instance, if we wanted to send the message BAD NEWS, we could repeat each letter a certain number of times and send, say,

BBBBBAAAADDDDD NNNNNEEEEEWWWWSSSSS.

Even if a number of these letters got garbled in transmission, the intended message could be recovered from a received message that might look like

BBEBFAAADGDDD . MNNNTEEEEEWWWSWRRSSS,

by a process called *majority decoding*, which in this case would mean that for each block of 5 letters the intended letter is the one which appears most frequently in the block.

Probability

The problem with this approach is economical, the repeat code is not very efficient. The increased length of the transmitted code, and thus the increased time and energy required to transmit it, is necessary in order to be able to decode the message properly, but how efficiently a coding procedure uses this increase depends upon the coding scheme. Suppose, in our example, that the probability that a letter is garbled in transmission is $p = 0.05$ and so $q = 1 - p = 0.95$ is the probability that a letter is correctly received. Without any coding, the probability of our 8 letter (spaces included) message being correctly received is

$$q^8 = (.95)^8 = 0.66.$$

(In this calculation we are assuming that the error in transmitting a single symbol is independent of which position the symbol is in. This is a common simplifying assumption ... which may not be appropriate in real world situations.)

Probability

Using the repeat code, the probability of correctly decoding a given letter from a block of 5 symbols is

$$q^5 + 5q^4p + 10q^3p^2$$

since there are three ways to decode correctly – 1) all the symbols are correct, 2) one symbol is incorrect (5 ways this can happen) or 3) two symbols are incorrect (10 ways this can happen) [notice that these are just terms in the expansion of $(q+p)^5$]. So we obtain

$$(.95)^5 + 5(.95)^4(.05) + 10(.95)^3(.05)^2 = 0.9988$$

and thus the probability of getting the correct eight letter message after decoding is $(0.9988)^8 = 0.990$, clearly a great increase over the non-coded message ($= 0.66$), but this 1% probability of getting the wrong message might not be acceptable for certain applications.

Terminology

To increase the probability of decoding the correct message with this type of code we would have to increase the number of repeats - a fix which may not be desirable or even possible in certain situations. However, as we shall see, other coding schemes could increase the probability to 0.9999 without increasing the length of the coded message.

Before leaving the repeat codes to look at other coding schemes, let us introduce some terminology. Each block of repeated symbols is called a *code word*, i.e., a code word is what is transmitted in place of one piece of information in the original message. The set of all code words is called a *code*. If all the code words in a code have the same length, then the code is called a *block code*. The repeat codes are block codes.

Detection and Correction

One feature that a useful code must have is the ability to detect errors. The repeat code with code words having length 5 can always detect from 1 to 4 errors made in the transmission of a code word, since any 5 letter word composed of more than one letter is not a code word. However, it is possible for 5 errors to go undetected (**how?**). We would say that this code is *4-error detecting*. Another feature is the ability to correct errors, i.e., being able to decode the correct information from the error riddled received words. The repeat code we are dealing with can always correct 1 or 2 errors, but may decode a word with 3 or more errors incorrectly (**how?**), so it is a *2-error correcting code*.

A Single Error Correcting Code

Before we look at the general theory of error correcting codes, let's consider another simple example. This is the simplest code that Hamming devised in his 1947 paper, *Self-Correcting Codes – Case 20878, Memorandum 1130-RWH-MFW*, Bell Telephone Laboratories. [[The first paper on error correcting codes](#)]

The information that is to be encoded is a collection of binary strings of length t^2 . The code words will be binary strings of length $(t+1)^2$, so $2t+1$ check digits will be added to the information data. To compute the check digits, arrange the data into a $t \times t$ array, add a mod 2 check sum digit to each row and column (including one for the row and column of check sums). The code word is the string obtained by writing out the rows of the extended array.

A Single Error Correcting Code

As a small example, consider the $t = 3$ case. We would code up the data 011010111 by

0	1	1	0
0	1	0	1
1	1	1	1
1	1	0	0

Calculate the check digits,

and produce the codeword 011001011111100.

When a word is received, it is arranged in a 4×4 array, the row and column check sums are recalculated and compared with the entries in the last column and last row. Differences indicate the row and column of any single error, which can then be corrected!

A Single Error Correcting Code

Suppose that the codeword 0110010111111100 is sent, but the word 0110011111111100 is received. We decode this:

0	1	1	0	0	
0	1	1	1	0	←
1	1	1	1	1	
1	1	0	0	0	
1	1	1	0		
					↑

By recalculating the check sums, we locate the row and column of the error. Clearly, any single error in a data digit will change only its row and column checks and so will be located. If an error is spotted only in a row but not a column (or vice versa) the error has occurred in a check digit and so, can again be corrected.

Code Rate

Any coding scheme used for detection or correction of errors will increase the size of the transmission. To the original data, the *information digits*, additional digits are added, called the *redundancy digits*, to obtain these capabilities. A measure of efficiency of the coding scheme, known as the *code rate*, is the ratio of the number of information digits to the size of the codewords. Efficient codes have the higher rates.

The code rate of the 5-repeat code is $1/5$.

The code rate of the PostNET code for ZIP codes is $5/6$.

Note that this rate does not measure detection or correction capability of a code. It is useful in comparing codes which have the same capabilities.