Mathematics in Error Detection and Correction

W. Wesley Peterson

I would like to show you how mathematics comes into solving problems in error detection and correction.

In practical error detection, the first idea used was the "parity check". The data are all binary—1's and 0's. We can represent a seven-bit message M as $M = (a_6a_5a_4a_3a_2a_1a_0)$, where each a_i is either 1 or 0. We say that the parity of M is even if there are an even number of 1's and the parity is odd if the number of 1's is odd. If you always send messages that have even parity and you always check the parity of the received messages, then you will detect any single error in a message, because the parity will become odd.

From here on I will assume arithmetic modulo 2, which means considering 2 to be equal to zero. The ordinary rules of algebra are true with modulo 2 arithmetic and we can use many ideas learned from algebra. Now if $M = (a_6a_5a_4a_3a_2a_1a_0)$ then $a_6+a_5+a_4+a_3+a_2+a_1+a_0$ is equal to zero if M has even parity and is equal to 1 if M has odd parity.

Assume that $M = (a_6a_5a_4a_3a_2a_1a_0)$ again and assume that a_6,a_5,a_4, and a_3 are information, and that a_2,a_1, and a_0 are parity checks on subsets of the symbols, as follows:

(2) $a_2 = a_6 + a_5 + a_4$, or $a_6 + a_5 + a_4 + a_2 = 0$

(1) $a_1 = a_5 + a_4 + a_3$, or $a_5 + a_4 + a_3 + a_1 = 0$

(0) $a_0 = a_6 + a_5 + a_3$, or $a_6 + a_5 + a_3 + a_0 = 0$

Let us look at this in a different way. In the following matrix, there is a 1 in column i and row j if variable a_i is in equation j.

Col.	(6)	(5)	(4)	(3)	(2)	(1)	(0)
(2)	1	1	1	0	1	0	0
(1)	0	1	1	1	0	1	0
(0)	1	1	0	1	0	0	1

Suppose $M = (1\ 0\ 1\ 0\ 0\ 1\ 1)$. You calculate parity check 2 as follows:

(M)	1	0	1	0	0	1	1
(2)	×1	×1	×1	×0	×1	×0	×0
	1	+0	+1	+0	+0	+0	+0 =0

This M has zero parity for all three equations. It is a suitable message to send. Now suppose it is sent and received with one error:

$M = (1\ 0\ 0\ 0\ 0\ 1\ 1)$

If you calculate the three parity checks for M, they come out as follows:

check (2) 1

check (1) 1

check (0) 0

So assuming that there was a single error, the digit with the error must have been in equation 2 and in equation 1 but not in equation 0. Looking at the table, we see that the parity checks match column 4. Variable a_4 is in equation 2 and in equation 1 but not in equation 0. The error must be in variable a_4.

Since every column in the table is different, every single error will result in a different pattern of parity check failures. Therefore this code can correct all single errors (including errors in the check bits). This code in a little different form was first found by Richard Hamming.

The calculation of the parity checks for a message M using the 3 by 7 array of 1's and 0's is exactly what is known in mathematics as matrix multiplication. A good deal of the mathematical theory of matrices can be applied here and is helpful as a theoretical framework for studying codes.

Corresponding to a message $M = (a_6a_5a_4a_3a_2a_1a_0)$ you can make a polynomial

$$A(x) = a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

For some codes, the symbols can be arranged so that the parity check equations will be satisfied if and only if $A(x)$ is divisible by some polynomial $P(x)$. I have arranged the symbols in this code so that the parity equations are satisfied if and only if $A(x)$ is divisible by $P(x) = x^3 + x + 1$. The remainder after dividing the received message by $P(x)$ turns out to be exactly the three parity checks calculated as before.

Thus polynomials come into the theory of error detection and correction, and with them, factoring, finding roots of equations, etc. These concepts have led to very significant developments in error correction and detection.

Error detection is particularly simple with codes of this kind. You choose a polynomial $P(x)$ and then send messages that as polynomials are evenly divisible by $P(x)$. You divide the received message by $P(x)$, and if the remainder is not zero you have detected an error. Dividing by $P(X)$ and finding the remainder is quite simple. The required circuit is a simple feedback shift register. This is the basis of the detection scheme known as the CRC, and it is still being used quite widely because of its simplicity and strong error detection capability. In particular, it is used in all ethernet networks and all diskettes.

This, I hope, gives some idea of the way mathematics comes into error detection and correction. I am still studying algebra, and I am still impressed with what mathematicians have been able to do.