Error detecting and error correcting codes: the new mathematics of shopping

Kaye Stacey
University of Melbourne

When I was a child there were no supermarkets and no cash registers in our town. When we went shopping, the shop-keeper got what we asked for from the shelves behind the counter. He wrote down the prices of the things we bought with a pencil, kept behind his ear, and then added them up on paper. Some of the prices had to be worked out first too, like one and a half pounds of apples at so much per pound. We had to be careful that we weren't overcharged, so we checked by adding up the prices as well as we could mentally. When the shop got a cash register, the adding up didn't seem so important, but we still needed to be careful that the right prices were put in and that we were being given the right change. Arithmetic and shopping were clearly linked.

Nowadays, there is little reason to associate routine shopping with arithmetic. The bill is added up automatically and the amount of change appears on the screen. Sometimes, you need to estimate how much something will cost, perhaps if you don't have quite enough money for what you want to buy - but usually asking for a subtotal solves the problem.

Most people would probably agree that routine shopping does not now require many calculations for most people. It once was an activity where many people used or saw being used both simple and complicated arithmetic (try finding the price of one and three quarter yards of material at twelve shillings and six pence per yard). Today, it is not. However, the surprising fact is that there are many more calculations going on in a supermarket today than ever before. Mathematics has gone underground, invisible, embedded into the machines – the cash registers, the EFTPOS links and the many other components of the supermarkets computer systems.

The obvious arithmetic in the supermarket is still in adding up the prices although even the cash register now clearly does a lot more: telling how much change to give, perhaps calculating a discount, working out the number of customer loyalty scheme points, etc. Even so, most of the calculations being done are not happening there. In this article, I will outline one way in which important but hidden calculations are being done for the communications in commerce. Two key concepts will be explained – error detecting codes and error correcting codes - and illustrated with two familiar examples – barcodes and ISBN numbers.

Barcodes on supermarket items – an error detecting code.

Codes are used when transmitting digital information to make sure that the information is as accurate as possible. The barcode is a common example of an error detecting code, which aims to make sure that information received or transmitted is accurate. It is not a particularly powerful code but it is suitable for the situation where it is used. Supermarkets want barcodes so that the shop computer can quickly tell what item it is to display the price and also to record that the item has been sold so that more can be ordered.

Bar codes are thick and thin black and white stripes. Modern barcodes represent 13 numbers, from 0 to 9. When the sales assistant scans the item, the 13 numbers are read. But sometimes the computer knows that there is an error and then it beeps and the sales assistant has to do it again. Somehow the scanner knows there has been an error. There are many reasons why there might be an error reading a barcode. Perhaps one of the stripes was not properly drawn from the beginning. The package might be crumpled or folded. Perhaps the condensation on frozen goods makes it hard to read. But how does the scanner know that there has been an error?

Supermarket barcodes actually have 12 digits containing the information required (country, manufacturer, type of goods etc) and the last one is a "check digit". The check digit is calculated from the other ones in this way:
Add up the first plus 3 times the second plus the third plus three times the fourth plus the fifth and so on until three times the twelfth. Then choose the thirteenth digit so that it makes all of this add up to a number ending in zero.

Example1: If the first twelve digits are 111111111111 the bar code would be 1111111111116. The reason is that $1 + 3 + 1 + 3 + 1 + 3 + 1 + 3 + 1 + 3 + 1 + 3 = 24$ so the check digit will be 6 so that the total sum is 30.

Example 2. If the first twelve digits are 930061509570 the bar code would be 9300615095707. The reason is that

$$9 + 3 \times 3 + 0 + 3 \times 0 + 6 + 3 \times 1 + 5 + 3 \times 0 + 9 + 3 \times 5 + 7 + 3 \times 0 = 63$$

so the check digit will be 7 to make the total 70.

When the supermarket computer reads the number, it instantly does the calculation to see if the check digit is right. So if it read the above bar code correctly as 9300615095707, it would think the number was correct and then give you the price for that item. However if it reads 8300615095707 instead, it would calculate that the check digit should be 8, not 7 and the machine will beep, requesting the sales assistant to scan the item again. This is what is meant by an error detecting code.

The barcode check digit system is a reasonably good code. It is easy to see that if there is one mistake the barcode system will always find that there has been a mistake. Unfortunately the system cannot work out the error. Decreasing any of the digits by one, for example, would have produced the same effect on the check digit as decreasing the first 9 to 8 as in the example above. Because it is easy enough for the sales assistant to scan the item again or type in the number, the barcode is a satisfactory system. If there are two mistakes or more, usually the check digit system will show that a mistake has been made, but sometimes two errors will "cancel out" and the customer might be charged for the wrong item. For example, 8400615095707 has errors in the first two digits which affect the check digit. This error would be detected. In other cases, such as 8310615095707 (coming from 9300615095707 with two errors in the first and third places) the check digit seems OK, because the sum is still the same. In these cases the error would not be detected and the wrong price will be charged.

A very common typing error is to mix up the order of two digits e.g. typing 16 instead of 61 in 9300615095707. The barcode system detects most of these, but not all. The mistake of transposing 6 and 1 will not be found by the check digit because $6 + 3 \times 1$ (9) will be replaced in the sum by $1 + 6 \times 3$ (19) and so the check digit will be the same. However transposing other pairs in this number, such as the initial 9 and 3 would be detected (in the sum $9 + 3 \times 1$ is replaced by $3 + 9 \times 1$). Further details about this, explaining which transpositions will be detected and a proof using the language of congruence are given in Barnett (1995).

ISBN numbers – another type of check digit

The system of ISBN numbers is used to identify books all around the world. It is another example of a code that can always detect one error. For example, Barnett's book which contains a highly readable chapter on codes from which a lot of this information was derived, has the ISBN number 0-13-834094-3. The first digits indicate the country, language and publisher and the later ones the actual book. The last digit is a check digit which is one of the numbers 0 to 9 or X, which stands for 10. Let the digits of the ISBN number be $x_1, x_2, x_3, x_4, \ldots$ up to x_{10} which is the check digit. The check digit is chosen so that $1\,x_1 + 2\,x_2 + 3\,x_3 + 4\,x_4 + 5\,x_5 + 6\,x_6 + 7\,x_7 + 8\,x_8 + 9\,x_9 + 10\,x_{10}$ is a multiple of 11. For example, if the ISBN number is 0-13—834094-3 the sum is $1 \times 0 + 2 \times 1 + 3 \times 3 + 4 \times 8 + 5 \times 3 + 6 \times 4 + 7 \times 0 + 8 \times 9 + 9 \times 4 + 10 \times 3 = 220$, which is a multiple of 11. This example has been included to show another of the many ways of designing check digits.

Codes that find errors and fix them

The barcode is a simple code that performs well enough in the situation where it is used. As long as a large percentage of the common errors are detected, the code is doing its job. Requesting the information again is a simple way of fixing errors made when scanned items, dragging credit cards through card readers etc. There are, however, other situations, where asking for the information again is not desirable. Information may, for example, have to use a very busy link where it is important to minimise the traffic or it may have to travel a long way and hence take a long time, such as for a distance telephone call or when pictures are being sent back to earth from other planets. The information may not even still exist. For these situations, codes have been developed which not only detect a large percentage of the errors, but can also fix many of them. Codes which have these fixing capabilities are called error-correcting codes. There are many different types and they are used in many situations, including the memory banks of computers, in medium and long distance telephones, in data links between banks, in CDs.

Making good error correcting codes relies on two things: very advanced mathematics (of many different sorts – number theory, geometry, matrices, polynomials etc) and knowing what the errors are probably going to be like. If you know what sort of errors there are likely to be, then a code might be able to be constructed to detect and fix errors of this type as a high priority. For example, I mentioned above, how a very common typing mistake is to transpose tow (no – it should be two) different letters. So a code that detects errors in typing should be sensitive to these. In some circumstances, the errors are likely just to be scattered randomly and infrequently. Radiation from radio active elements in the plastic around the memory in a computer, for example, can be emitted at random and destroy a memory cell, but it is unlikely to affect its neighbours. Error correcting codes are built into the memory to safe guard against this. In other cases, errors occur in bursts. Lightening, for example, can cause bursts of errors in signals being transmitted by satellite such as long distance telephone calls. If there is an error in one place, then it is likely that the places near it will have errors. When the laser light in a CD player reads the information from a CD, it is likely that there will be "error bursts". The information is stored in groups (e.g. 7 groups of 3) and if one digit in a group of 3 is wrong, it is likely that the other two are as well. A CD player reads about 1.5 million digits per second of audio information. These directly encode the music of the CD. However about twice as many bits of information are used in various aspects of control, the digital displays etc and also as extra digits to give an error-correcting capacity, which results in the exceptionally high quality sound. The coding procedures to recover from the type of errors likely to be encountered in CD's were discovered only a few years before CD's were produced commercially. Further details are given by Barnett (1995).

What might an error correcting code do?

To give some idea of the power of an error correcting code, imagine that some check digits were added to our telephone numbers. Instead of dialling the current 8 digit numbers, we would have to dial 9, 10 or 11 in a capital city in Australia. The more error correcting capacity was required, the more extra digits would need to be added.
If we used the same system as the barcodes, adding one digit to the existing numbers, then when you dialled one digit wrongly, you would get a message saying " you have dialled the wrong number" Please try again". The computer equipment in the exchange would have carried out the computation and decided the number was wrong. You wouldn't be connected to a wrong number. Additionally, if you transposed two digits, most of the time you would get the same message – occasionally though it would go through and connect to the wrong phone. Making other mistakes, like three digits wrong would generally be picked up, but sometimes you would get a wrong number.

However, if we used a more sophisticated system of check digits than the barcode, we can do better than this. If, for example, we used two check digits and choose the code carefully any single mistake to be detected AND FIXED. So if you dialled a number wrong in one digit, the telephone would know you had made a mistake and it would fix it for you and connect you to the right number. If you transposed two of the digits, the telephone system would always know – but it cant fix it because there are too many possibilities – but then you could dial again. If more check digits were added, it would be possible to detect and check many more errors. In a telephone system, the advantages of having an error correcting capacity are probably quite small in comparison with the nuisance of dialling extra digits. But in other circumstances, it is invaluable. The code used to transmit the pictures of Mars from Mariner 9 in 1971 was a 32 digit code with 26 check digits which could correct up to 7 errors. To transmit these pictures all the way from Mars to the Earth, it was only necessary to use a 20watt transmitter – less than a light bulb. Long distance telephones use BCH codes of length 255 with 24 check digits, which can correct three errors.

A simple error correcting code

By combining the ideas used in barcodes and ISBN numbers, a simple code that can identify errors and fix some of them can be created. Imagine there are 10 digits ($x_1, x_2, x_3, \cdots x_{10}$) containing information and define two check digits called x_{11} and x_{12}. Choose x_{11} so that $S_1 = x_1 + x_2 + x_3 + \ldots + x_9 + x_{10} + x_{11}$ is a multiple of 11 and choose x_{12} so that $S_2 = 1x_1 + 2x_2 + 3x_3 + \cdots + 9x_9 + 10x_{10} + x_{12}$ is a multiple of 11.
The extra digit X (standing for 10) may be needed for x_{11} or x_{12}.

Examples. If the ten digit information is 1111111111 then x_{11} is chosen to be 1 and x_{12} is chosen to be 0. So the full code word would be 111111111110.
If the ten digit information is 1234567890, then x_{11} is chosen to be X(ten) and x_{12} is chosen to be 1 because $1 \times 1 + 2 \times 2 + 3 \times 3 + \ldots + 9 \times 9 + 0 \times 0 = 285$ and 286 is a multiple of 11. The full code word would therefore be 1234567890X1.

The proof that this code can identify and correct all one digit errors can be found in Barnett (1995). This code can also detect all two digit errors, although it cannot correct them. The proof uses simple ideas of modular arithmetic and the very important fact that 11 is prime so that arithmetic modulo 11 forms a field. To obtain an idea of how the error correction works, one example is shown here. Assume that a one digit error is most likely. If the code word 1234561890X1 was received, the first sum S_1 would be 49, which is 5 larger than a multiple of 11 or 6 smaller than a multiple of 11. Assuming there is only one error, this means that one of the digits is 5 too large or 6 too small. In the table below, the effect on S_2 of a number 5 too large or 6 too small in each place is shown.

Effect of error on S_2

Place of digit	if 5 too large	if 6 too small	removing multiples of 11
1	5 too large	6 too small	5 too large, 6 too small
2	10 too large	12 too small	10 too small, 1 too small
3	15 too large	18 too small	4 too large, 7 too small
4	20 too large	24 too small	9 too large, 2 too small
5	25 too large	30 too small	3 too large, 8 too small
6	30 too large	36 too small	8 too large, 3 too small

7	35 too large	42 too small	2 too large, 9 too small
8	40 too large	48 too small	7 too large, 4 too small
9	45 too large	54 too small	1 too large, 10 too small
10	50 too large	60 too small	6 too large, 5 too small

Now the calculation for S_2 is carried out on the number that has been received, 1234561890X1. This is 244, which $22 \times 11 + 2$ or $23 \times 11 - 9$ i.e. 2 larger or 9 smaller than the S_2 check digit ought to be. The table above pinpoints the error as being in the 7^{th} place. The digit 1 must be 6 too small (it obviously isn't 5 too big) , so the correct digit would have been 7.

This is what is meant by an error correcting code. The equipment can contain mini computer circuits purpose-built to fix the errors. Providing the errors that occur are the sort that are predicted, a good code will correct the errors in a way that can be done fast and cheaply. This is how we get accurate pictures from Mars and excellent sound quality from CD's and mobile phones.

More sophisticated codes

The mathematical theories that are used to develop the sort of codes above are based on mathematics that has been discovered in the last 500 years. The most sophisticated codes are based on mathematics that is quite new and much remains to discover about them. In fact there are many different families of codes, made for different purposes. They are based on mathematical insights from the areas of number theory (as above), geometry, algebra of polynomials, matrices, statistics and other fields. Other codes are harder to explain in a short article, but the references given are useful.

Some codes are built by thinking about geometry in higher dimensional spaces. For example, modems that use Trellis Coded modulation are using a code that is built from the best way that 8 dimensional spheres can be packed in rows in 8 dimensional space. The code words are made to represent the centres of spheres in 8 dimensional space. Problems of packing higher-dimensional spheres are of interest now for man sort of telecommunications. For example, Conway and Sloane (1988) explain how engineers working on mobile radio needed to know how many 100 dimensional spheres of radius 0.43 can be fitted into a sphere of radius 1 in 100 dimensional space and give other examples. The mathematical theory of many dimensions seems on the surface an abstract and unreal theory, but is now commercially valuable.

An important application of codes developed from geometry in higher dimensional space is described by Barnett (1995). Error correcting codes are used to dramatically improve the reliability of computer memories. The plastic packaging of memory chips contains very small numbers of radioactive molecules. The radiation that can be released from one of these radioactive molecules is enough to alter the contents of a memory cell. Any one memory cell in a memory bank is incredibly reliable. The expected time they could individually operate as memory cells is over a million years. But because a memory bank consists of so many of them (about 8 million in 8 MB), the chance of a failure somewhere is quite large. In fact, the expected time that the memory bank as a whole would last is only 43 days – far too short for practical purposes. The solution to this problem is provided by incorporating a code with 7 check digits) into the memory bank. As there is more memory, this would increase the chance of memory cells being hit by radiation to about 36 days. However, because the code can fix most common errors the frequency of errors in the memory bank comes down to about one every 63 years – a very practical improvement.

Mathematics embedded

There are many different types of codes and many different branches of mathematics are used to develop codes that are good in different circumstances. This article has offered only a very sketchy ideas of some simple codes. However, I hope that it has shown two points about how where to look for mathematics. Firstly, new technology is embedding routine mathematical calculations in machines. In all aspects of life, we will see people doing fewer and fewer calculations, solving fewer equations, drawing fewer graphs and making fewer and fewer logical decisions. Another simple example: when ten pin bowling first became popular, the participants needed to score themselves (writing down the numbers, adding them up) and needed to know the rules about how to score with a spare and a strike etc. Nowadays, all this is orchestrated by a computer, which does the scoring, the adding up and organises who plays next. Decisions about teaching mathematics to tomorrow's citizens must take the new way in which mathematics is being used in the world into account. Specialised knowledge from a few is built into machines to be used by many. In a supermarket, few of the customers will be calculating, but their mobile phones are doing thousands of calculations every second.

References

Arazi, B. (1988) *A Commonsense Approach to the Theory of Error Correcting Codes*. MIT Press: Cambridge, MA.

Barnett, Stephen (1995) *Some Modern Applications of Mathematics*. Ellis Horwood: Hemel Hempstead.

Conway, J.H. & Sloane, N.J.A (1988) *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag.

Recent articles in AMT on related topics:

Holton, D. (1995) RSA Public Key Cryptography. *Australian Mathematics Teacher*, 51 (3), 10 - 13.

Donovan, D. (1997) Secretly Sharing Passwords. *Australian Mathematics Teacher*, 53 (1), 14 - 17.