9
error correcting codes

A main idea of this sequence of notes is that accounting is a legitimate field of academic study independent of its vocational implications and its obvious importance in the economic environment. For example, the study of accounting promotes careful and disciplined thinking important to any intellectual pursuit. Also, studying how accounting works illuminates other fields, both academic and applied. In this chapter we begin to connect the study of accounting with the study of codes. Coding theory, while theoretically rich, is also important in its applications, especially with the pervasiveness of computerized information transfer and storage. Information integrity is inherently of interest to accountants, so exploring coding can be justified for both academic and applied reasons.

9.1 kinds of codes

We will study three kinds of codes: error detecting, error correcting, and secret codes. The first two increase the reliability of message transmission. The third, secret codes, are designed to ensure that messages are available only to authorized users, and can't be read by the bad guys. The sequence of events for all three types of codes is as follows.

1. determine the message to transmit. For our purposes, we will treat messages as vectors.

2. The message is encoded. The result is termed a codeword, or cyphertext, also a vector.

3. The codeword is transmitted through a channel. The channel might inject noise into the codeword, or it might be vulnerable to eavesdropping.

4. The received vector is decoded. If the received vector is no longer the transmitted codeword, the decoding process may detect, or even correct, the error.

Setting aside secret codes until chapter 10, this chapter will concentrate on error detecting and correcting codes. Applications of these types of codes are numerous. In a business environment codes are used to design account numbers, inventory part numbers, and all sorts of identification numbers. Codes enable efficient transmission of television pictures as well as pictures sent from the farthest parts of the solar system. The cause of noise in a codeword may be as mundane as a typing mistake, or as cosmic as interference from sunspots.

We will study a popular class of error detecting and correcting codes called linear codes. Linear codes employ the same techniques we have used for understanding the linear transformations in accounting. Both encoding and decoding are accomplished by matrix multiplication. Furthermore, decoding is a direct application of the concept of a nullspace.

Consider decoding. Every linear code can be specified by a matrix called a parity check matrix, denoted H. Decoding is accomplished by multiplying the received vector by H. If the received vector is in the nullspace of H, then the received vector is a legal codeword. Notice the connection with accounting. H is "like" the accounting transformation matrix A. The legal codewords are "like" the set of looping transactions which leave the account balances unchanged. Similar to the accounting applications, the received vector, denoted y, is in the nullspace of H if H times y is a vector of zeros.

Definition 9.1 *The matrix product Hy is termed the syndrome.*

If the syndrome contains a non-zero element, an error has been detected. If we are clever in our analysis of the syndrome, we may be able to infer the position and amount of the error, thereby allowing error correction.

Before moving to examples of codes, we need to acquire another tool; the notion of a finite field.

9.2 modular arithmetic

We will restrict ourselves to a finite set of messages which will, in turn, imply a finite number of errors. This allows for efficient error detection and correction. The mechanism to accomplish this is modular arithmetic. The main idea is pretty simple, and the notation for it (devised by Gauss) is straightforward. An excellent introduction to modular arithmetic and number theory is in Ore, 1948.

Definition 9.2 *Modular arithmetic reduces the set of integers under consideration to a finite number.*

$$a = b + cm \Longleftrightarrow a \equiv b(\mathrm{mod}\ m) \textit{ where } a, b, c, \textit{ and } m \textit{ are integers}.$$

The second equivalence is read "a is congruent to b modulo (or simply 'mod') m."

Any integer has a corresponding element in a finite set. Simply divide by m and report the remainder. There will only be m integers in the reduced set.

A convenient example is "binary." The arithmetic is done modulo 2; all integers are equivalent (congruent) to either 0 or 1. For example,

$$\begin{aligned} 2 &\equiv 4 \equiv 6 \equiv 0(\mathrm{mod}\ 2) \\ 1 &\equiv 3 \equiv -1 \equiv 1(\mathrm{mod}\ 2) \end{aligned}$$

Multiplication and addition work pretty much the way we are used to. The key is the answer is always a member of the original set. For binary the answer is either 1 or 2. For example,

$$1 + 1 \equiv 0\,(\mathrm{mod}\ 2)$$

Matrix multiplication also works in modular arithmetic.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} (\mathrm{mod}\ 2)$$

In preparation for the first error detecting example consider arithmetic modulo 11, in which every number is divided by 11, and the remainder reported.

$$\begin{aligned} 7 + 6 &= 13 \equiv 2\,(\mathrm{mod}\ 11) \\ 7 \times 6 &= 42 \equiv -2 \equiv 9\,(\mathrm{mod}\ 11) \end{aligned}$$

Here's the entire multiplication table modulo 11.

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	1	3	5	7	9
3	3	6	9	1	4	7	10	2	5	8
4	4	8	1	5	9	2	6	10	3	7
5	5	10	4	9	3	8	2	7	1	6
6	6	1	7	2	8	3	9	4	10	5
7	7	3	10	6	2	9	5	1	8	4
8	8	5	2	10	7	4	1	9	6	3
9	9	7	5	3	1	10	8	6	4	2
10	10	9	8	7	6	5	4	3	2	1

There are two important things to notice about the multiplication table.

1. There are no zeros. When two non-zero numbers are multiplied, the result is non-zero. This property does not hold for all moduli. For example,

$$6 \times 6 \equiv 0 \,(\mathrm{mod}\ 9)$$

2. There are no "repeats." Each row (and column) consists of all the possible 10 numbers.

The two properties have important applications in coding, and they follow directly from the concept of a prime number and what is known as the fundamental theorem of arithmetic.

Definition 9.3 *A prime number is divisible (evenly, that is, leaving no remainder) only by 1 and the number itself.*

Theorem 9.1 *Any integer can be factored into a product of prime numbers. Furthermore, the factorization is unique.*

Consider property 1 for some prime modulus, p. A zero entry in the multiplication table means

$$rs \equiv 0 \,(\mathrm{mod}\ \mathrm{p}) \Longleftrightarrow rs = pn$$

where r and s are integers less than p and n is some integer. The equality on the right implies two different factorizations of rs, violating unique factorization. Hence, the fundamental theorem implies property 1 is true for all p.

Property 2 follows from similar logic. "Repeats" imply the first two equations below which, in turn, imply the third.

$$\begin{aligned} rs &= pn + k \\ rt &= pm + k \\ r\,(s-t) &= p\,(n-m) \end{aligned}$$

And the third equation is not allowed by the fundamental theorem.

With these preliminaries we are ready for an example code.

9.3 isbn - an error detecting code

ISBN stands for "international standard book number." Virtually every book published in the world is assigned an ISBN. The numbers are assigned in a way so that some typographical errors which might occur in typing or transcribing an order can be detected. The probability that the wrong book is delivered is thereby reduced.

In 2007 the design of the ISBN was altered slightly. A visible manifestation is that the length of the number (codeword) increased from 10 to 13. We'll start with ISBN 10 as it supplies a nicer illustration of the two properties in the previous section, and catch up with ISBN 13 later.

9.3.1 isbn 10

The ISBN is a linear code in the sense that it is completely specified by its parity check matrix. For ISBN 10 the parity check matrix has one row and 10 columns.

$$H = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}$$

All ISBN codewords reside in the nullspace of H where arithmetic is conducted modulo 11. That is, for an ISBN y vector,

$$Hy \equiv 0 \pmod{11}$$

If y does not satisfy the above equation, an error has been made.

Example 9.1 *The ISBN for The Norton History of Mathematics (published prior to 2007) is 0-393-04650-8.*

The ISBN is divided into 4 parts, possibly of various lengths across countries and companies. The first number is the official language of the country in which the book is published: zero is English. The second set of numbers is the number assigned to the publisher: W. W. Norton Publishing is 393. The next set is an internal inventory number chosen by the publisher. The last number is a check digit which ensures the ISBN resides in the nullspace of H.

Do the arithmetic.

$$\begin{aligned} Hy &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 9 \\ 3 \\ 0 \\ 4 \\ 6 \\ 5 \\ 0 \\ 8 \end{bmatrix} \\ &= 231 = 11 \times 21 \equiv 0 \,(\mathrm{mod}\ 11) \end{aligned}$$

The ISBN code is designed to detect any single error and any transposition error, not necessarily of adjacent digits. Check to see that any such error for Norton results in a non-zero syndrome. More generally, the two error types are always caught because of the two noted properties of the multiplication table.

Consider a single error. Let the received ISBN vector be the sum of the correct ISBN and an error vector with a non-zero entry in position i.

$$y = y_{ISBN} + e = y_{ISBN} + \begin{bmatrix} 0 \\ . \\ . \\ . \\ e_i \\ . \\ . \\ . \\ 0 \end{bmatrix}$$

The syndrome is calculated.

$$Hy = H\left(y_{ISBN} + e\right) = Hy_{ISBN} + He = 0 + He$$

He is e_i times the ith element of H. As both numbers are non-zero, the syndrome is non-zero by property 1, and the error is detected.

Consider a transposition error, not necessarily of adjacent digits. Suppose in the received y that elements y_j and y_k are transposed. The received y vector appears as follows.

$$
\begin{aligned}
y &= y_{ISBN} + \begin{bmatrix} 0 \\ . \\ . \\ . \\ y_k - y_j \\ . \\ . \\ . \\ y_j - y_k \\ . \\ . \\ . \\ 0 \end{bmatrix} \\
&= y_{ISBN} + \begin{bmatrix} 0 \\ . \\ . \\ . \\ y_k - y_j \\ . \\ . \\ . \\ -(y_k - y_j) \\ . \\ . \\ . \\ 0 \end{bmatrix}
\end{aligned}
$$

For the syndrome He to be zero, two distinct elements of H, H_j and H_k, must yield the same answer when multiplied by the same number:

$$y_k - y_j$$

But that is impossible by property 2, so the transposition error is detected.

One more note before moving on to ISBN 13. Sometimes to ensure the ISBN resides in the nullspace of H, the check digit, the 10th element of y_{ISBN}, must be the number 10. Rather than writing the two digit number, the ISBN assigns the Roman numeral X. From watching the super bowl, we know X stands for 10.

9.3.2 isbn 13

ISBN 13 is similar to the universal product code used for all kinds of inventory items. It has a different looking parity check matrix.

$$H = \begin{bmatrix} 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 \end{bmatrix}$$

Furthermore, the arithmetic is done modulo 10, the multiplication table for which follows.

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	0	2	4	6	8
3	3	6	9	2	5	8	1	4	7
4	4	8	2	6	0	4	8	2	6
5	5	0	5	0	5	0	5	0	5
6	6	2	8	4	0	6	2	8	4
7	7	4	1	8	5	2	9	6	3
8	8	6	4	2	0	8	6	4	2
9	9	8	7	6	5	4	3	2	1

It is noticed right away that the two properties of no zeros and no repeats do not hold, in general. They do hold, however, for the rows and columns associated with 1 and 3, among others. And, since those are the only numbers used in H, a single error remains detectable by the syndrome. Further, some transposition errors are detectable, as well. For example, transposition of adjacent digits will usually yield a non-zero syndrome, as the same number multiplied by a 1 and a 3 usually gives a different answer. The exception is 5. So a transposition error of adjacent digits which differ by 5 will not be caught.

$$\begin{aligned} 3\,(y_j - y_k) + 1\,(y_k - y_j) &\equiv 0 \,(\mathrm{mod}\, 10) \\ \text{when } y_j - y_k &\equiv 5 \,(\mathrm{mod}\, 10) \end{aligned}$$

Furthermore, transposition of digits removed by two places will not be caught; the syndrome will still be calculated as zero. Presumably, the designers of the code are less worried about this particular error type occurring often.

Example 9.2 *The ISBN for Managerial Uses of Accounting Information by Joel Demski (published post 2007) is 978-0-387-77450-3.*

Do the arithmetic.

$$\begin{aligned} Hy &= \begin{bmatrix} 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 9 \\ 7 \\ 8 \\ 0 \\ 3 \\ 8 \\ 7 \\ 7 \\ 7 \\ 4 \\ 5 \\ 0 \\ 3 \end{bmatrix} \\ &= 120 \equiv 0 \,(\mathrm{mod}\ 10) \end{aligned}$$

The universal product code (UPC) is a ubiquitous variation on ISBN 13. The UPC is particularly visible at supermarkets where the checkout scanners read the bar codes on the inventory items. Typically, a UPC code has 12 digits, and the parity check matrix is

$$H = \begin{bmatrix} 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 1 \end{bmatrix} (\mathrm{mod}\ 10)$$

9.4 an error correcting code

So far we have been able to discern when an error exists in the codeword. We have not, however, been able to fix the error, at least not with the information in the syndrome alone. It is possible, by expanding the parity check matrix, to not only detect, but also correct errors in the codeword. The following example, along with other linear codes can be found in Hill, 1986.

Example 9.3 *Define the parity check matrix as follows with modulus 2.*

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Now the syndrome, Hy, has 3 elements, and the extra information can be used for error correction. The logic is not complicated. An error means the element y_i is a one instead of a zero, or vice-versa. For no errors, the syndrome is all zeros, that is, in the nullspace of H. If y_i is one instead of zero, then by the rules of matrix multiplication, the syndrome will be the ith column of H. And, since $-1 \equiv 1 \,(\mathrm{mod}\, 2)$, if y_i is a zero instead of a one, the syndrome will likewise be the ith column of H. Check with an example.

Example 9.4 *Suppose the received codeword is y=*$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}^T$. *Syndrome decoding yields*

$$Hy = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} (\mathrm{mod}\ 2)$$

As the syndrome is the 4th column of H, it indicates y_4 should be corrected to a 1 from 0. The corrected y

$$y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

And calculation of the syndrome verifies the new y is now a legal codeword.

$$Hy = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \pmod 2$$

It would be convenient to generate legal codewords directly, rather than pick a random vector and correct it using syndrome decoding as above. There must be a simpler way, and, indeed, there is: use of the generator matrix.

9.4.1 generator matrix

In the example under consideration, the original vector (inventory number, employee id, etc.) is 4 elements long. Call it x. The matrix which multiplies the original message, x, in order to generate the codeword is called the generator matrix, G.

$$Gx = y$$

In this case, the generator matrix adds three redundant elements to x, enabling syndrome decoding to correct any single error.

It turns out to be fairly easy to construct G given H. Write H in the following block matrix format.

$$\begin{aligned} H &= \begin{bmatrix} B & I_3 \end{bmatrix}, \text{where} \\ B &= \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \text{ and} \\ I_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

For G to be a legitimate generator matrix, the matrix product Gx must yield the zero vector when multiplied by H.

$$HGx = Hy = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ for all possible } x\text{'s}$$

This implies G is 7×4 such that

$$HG = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Here's the one that does the trick.

$$\begin{aligned} G &= \begin{bmatrix} I_4 \\ -B \end{bmatrix} \\ HG &= \begin{bmatrix} B & I_3 \end{bmatrix} \begin{bmatrix} I_4 \\ -B \end{bmatrix} = BI_4 - I_3B \\ &= B - B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Substituting for B, and recalling that -1 $\equiv 1 \pmod 2$, the generator matrix is as follows.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Example 9.5 *Let the original message be* $x = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^T$.

Calculate the codeword by multiplying by the generator matrix, G.

$$Gx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \pmod 2 = y$$

Notice that G keeps the original message, x, intact and adds three redundant elements. Decoding verifies that y is a legal codeword (no noise has been injected).

$$Hy = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \pmod 2$$

9.4.2 perfect codes

The example code has another property which is convenient, on occasion.[1]

Definition 9.4 *A perfect single error correcting code is one in which the number of possible received vectors is equal to the number of legal codewords plus the number of vectors which can be changed into a legal codeword with exactly one correction.*

For a perfect code, in other words, there are no wasted vectors. Every vector is either a legal codeword, or just one element removed from a legal codeword.

To demonstrate the example code satisfies the definition for perfect requires two steps. First, it is verified there is no vector which can simultaneously be corrected to two legal codewords. That is, no vector is one change from two different legal codewords. But that can't happen. Inspection of the parity check matrix verifies there is no ambiguity about which element should be corrected: the columns of H are distinct. Whichever column is equal to the syndrome specifies the element of the received vector to correct.

Now it is a matter of counting the number of vectors in each of the categories.

- the total number of possible received vectors: $2^7 = 128$.
- the total number of legal codewords: $2^4 = 16$.
- the total number of ways a received vector can be one off from a legal codeword: $16\,(7) = 112$.

The last calculation is the number of legal codewords times the number of positions available for an error to occur. Perfectness is verified by the sum: $112 + 16 = 128$.

[1] It is particularly easy, for example, to construct examination questions.

9.5 another set of examples

The example code of the previous section, while illustrating some nice properties, is not a very large code. That is, if the codewords is meant to characterize different inventory items, for example, the size of the inventory is limited to 16 units. It is relatively straightforward, however, to increase the size of the code by increasing the modulus. As usual, the way to specify the code is to write down the parity check matrix. For the next example, arithmetic is done modulo 5.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \pmod 5$$

With modulo 5 there are 5 possible values for each element. The syndrome, then, must supply information, not only about the position of an error, but the amount of the error, as well. When there is a zero in the syndrome, the error is easily positioned as in element 5 or 6.

Example 9.6 *Let the received vector be* $y = \begin{bmatrix} 2 & 1 & 2 & 1 & 3 & 1 \end{bmatrix}^T$. *Perform syndrome decoding.*

$$\begin{aligned} Hy &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 9 \\ 15 \end{bmatrix} \\ &\equiv \begin{bmatrix} 4 \\ 0 \end{bmatrix} \pmod 5 \end{aligned}$$

The syndrome can be fixed by subtracting 4 times the 5th column of H. And that can be accomplished by subtracting 4 from the 5th element of y. Since $-4 \equiv 1 \pmod 5$, the correction is to add 1 to the 5th element.

$$\begin{aligned} \text{corrected } y_c &= \begin{bmatrix} 2 & 1 & 2 & 1 & 4 & 1 \end{bmatrix}^T \\ Hy_c &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 15 \end{bmatrix} \\ &\equiv \begin{bmatrix} 0 \\ 0 \end{bmatrix} \pmod 5 \end{aligned}$$

And the codeword is corrected.

When there is no zero in the syndrome, the error resides in one of the first 4 elements.

Example 9.7 *Let the received vector be* $y = \begin{bmatrix} 3 & 2 & 4 & 1 & 2 & 3 \end{bmatrix}^T$. *Perform syndrome decoding.*

$$\begin{aligned} Hy &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 4 \\ 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 12 \\ 26 \end{bmatrix} \\ &\equiv \begin{bmatrix} 2 \\ 1 \end{bmatrix} (\mathrm{mod}\, 5) \end{aligned}$$

As the first row of H is all 1's, the first element of the syndrome is the amount of the error. The syndrome can be fixed by finding the column of H, when multiplied by 2, yields the syndrome. Because of the "no zeros - no repeats" properties of a prime modulus, only one column of H will satisfy the condition. There is, then, no ambiguity about the position and amount of the correction.

Searching the columns reveals the syndrome is 2 times the 3rd column of H.

$$2 \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \end{bmatrix} \equiv \begin{bmatrix} 2 \\ 1 \end{bmatrix} (\mathrm{mod}\, 5)$$

The correction is to subtract 2 from the 3rd element of y.

$$\begin{aligned} \text{corrected } y_c &= \begin{bmatrix} 3 & 2 & 2 & 1 & 2 & 3 \end{bmatrix}^T \\ Hy_c &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 2 \\ 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \end{bmatrix} \\ &\equiv \begin{bmatrix} 0 \\ 0 \end{bmatrix} (\mathrm{mod}\, 5) \end{aligned}$$

And the correction results in the appropriate syndrome.

A generator matrix can be constructed using the methods of the previous section.

$$G = \begin{bmatrix} I_4 \\ -B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 4 & 4 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

Example 9.8 *Consistent with the prior example, let the original message* $x = \begin{bmatrix} 3 & 2 & 2 & 1 \end{bmatrix}$. *Calculate the redundant digits.*

$$\begin{aligned} Gx &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & 4 & 4 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 2 \\ 1 \\ 32 \\ 23 \end{bmatrix} \\ &\equiv \begin{bmatrix} 3 \\ 2 \\ 2 \\ 1 \\ 2 \\ 3 \end{bmatrix} (\mathrm{mod}\, 5) \end{aligned}$$

And the appropriate redundant digits are added, as consistent with the syndrome analysis of the problem.

The final thing to do with this example is to verify the perfectness of the code. We already resolved there is no ambiguity in the correction, so just count the vectors in the categories.

- the total number of possible received vectors: $5^6 = 15,625$.
- the total number of legal codewords: $5^4 = 625$.
- the total number of ways a received vector can be one off from a legal codeword: $625\,(6)\,(4) = 15,000$.

The last calculation is the number of legal codewords times the number of positions available for an error to occur times the number of possible error amounts. Perfectness is verified by the sum: $15,000 + 625 = 15,625$.

One more example demonstrates the code can become as large as desired.

Example 9.9 *Perform arithmetic modulo 11, and let the parity check matrix be*

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 0 & 1 \end{bmatrix} (\mathrm{mod}\, 11)$$

The code has $11^{10} = 25,937,424,601$ legal codewords. This coding system could handle, for example, 12 digit phone numbers. If an individual dialed the number with one mistake, the system can correct the error, and the call can still go through to the intended party. There are no wasted phone numbers, as the code is a perfect one. The number of "one off" phone numbers is $11^{10}\,(12)\,(10) = 11^{10}\,(120)$, the number of legal phone numbers times the number of positions available for an error times the number of possible error amounts. The total of 12 digit phone numbers is $11^{12} = 11^{10}\,(1 + 120) = 11^{10}\left(11^2\right)$.

9.6 double error correction

To do *single* error correction the idea is to solve two linear equations for the two unknowns: the amount and position of the error. For *double* error correction there are four unknowns: two error amounts and their respective positions in the received vector. That requires four independent equations which, in turn, implies a parity check matrix with four rows. This double error correcting code is also in Hill, 1986.

Here is a 4 row parity check matrix which defines a linear code in what is called the BCH class.[2]

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2^2 & 3^2 & 4^2 & 5^2 & 6^2 & 7^2 & 8^2 & 9^2 & 10^2 \\ 1 & 2^3 & 3^3 & 4^3 & 5^3 & 6^3 & 7^3 & 8^3 & 9^3 & 10^3 \end{bmatrix}$$

The arithmetic is done modulo 11.

Suppose vector y is sent, but the received vector is $y + e$ where the error vector, e, has 2 errors, a and b, in positions i and j.

$$e^T = \begin{bmatrix} 0 & \ldots & 0 & a & 0 & \ldots & 0 & b & 0 & \ldots & 0 \end{bmatrix}$$

The syndrome $Hy \equiv s$ looks like the following, where s_i is the i^{th} element in the syndrome vector, s.

$$\begin{aligned} s_1 &\equiv a + b \\ s_2 &\equiv ai + bj \\ s_3 &\equiv ai^2 + bj^2 \\ s_4 &\equiv ai^3 + bj^3 \end{aligned}$$

That's the set of equations we have to deal with.

Before embarking on the 4 equation 4 unknown problem, notice that no errors and a single error are fairly simple to deal with. For an error free transmission $Hy \equiv 0$ (i. e., $s_1 \equiv s_2 \equiv s_3 \equiv s_4 \equiv 0$). Then $a \equiv b \equiv i \equiv j \equiv 0$ solves the four equations.

If there is only one error, then $b = 0$, and the equations are

$$\begin{aligned} s_1 &\equiv a \\ s_2 &\equiv ai \\ s_3 &\equiv ai^2 \\ s_4 &\equiv ai^3 \end{aligned}$$

The first two equations can be easily solved for a and i which will then satisfy the other equations.

---

[2] See, for example, Hill, *A First Course in Coding Theory*, chapter 11.

For the general 2 error case return to the 4 non-trivial equations.

$$\begin{aligned} s_1 &\equiv a+b \\ s_2 &\equiv ai+bj \\ s_3 &\equiv ai^2+bj^2 \\ s_4 &\equiv ai^3+bj^3 \end{aligned}$$

The first step is to eliminate one unknown, a, by multiplying the first equation by i and subtract the second equation. Similarly, multiply the second by i and subtract the third; finally, multiply the third and subtract the fourth. We now have 3 equations in3 unknowns, numbered (1), (2), and (3).

$$\begin{aligned} is_1 - s_2 &= b(i-j) \qquad (1) \\ is_2 - s_3 &= bj(i-j) \qquad (2) \\ is_3 - s_4 &= bj^2(i-j) \qquad (3) \end{aligned}$$

Now convert 3 equations into one quadratic equation. To do this, multiply (1) times (3), and, also, (2) times itself.

(1) times (3):

$$\begin{aligned} b^2j^2(i-j)^2 &\equiv (is_1 - s_2)(is_3 - s_4) \\ &\equiv i^2 s_1 s_3 - i(s_1 s_4 + s_2 s_3) + s_2 s_4 \end{aligned}$$

(2) times (2):

$$\begin{aligned} b^2j^2(i-j)^2 &\equiv (is_2 - s_3)^2 \\ &\equiv i^2 s_2^2 - 2is_2 s_3 + s_3^2 \end{aligned}$$

Combining:

$$i^2\left(s_2^2 - s_1 s_3\right) + i(s_1 s_4 - s_2 s_3) + \left(s_3^2 - s_2 s_4\right) \equiv 0 \,(mod\ 11)$$

We have a quadratic equation modulo 11 which, with a few adjustments for modular arithmetic, we can solve for position i of one of the errors. Actually, as the quadratic equation has two roots, we will get *both* error positions. To simplify the notation let

$$\begin{aligned} p &= s_2^2 - s_1 s_3 \\ q &= s_1 s_4 - s_2 s_3 \\ r &= s_3^2 - s_2 s_4 \end{aligned}$$

And we can specify the two roots as

$$i, j \equiv \frac{-q \pm \sqrt{q^2 - 4pr}}{2p} \,(\mathrm{mod}\, 11)$$

We notice at this point that square root and division are not defined in modular arithmetic, but we can handle that as we shall see in the numerical example. Once i and j are known, use equation (1) to get b:

$$b \equiv \frac{is_1 - s_2}{i - j}$$

And use the first syndrome equation to get a:

$$a \equiv b - s_1$$

Example 9.10 *Suppose* $e^T \equiv \begin{bmatrix} 0 & 2 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$. *Do syndrome to find* a, b, i, *and* j. *Looking ahead we should find*

$$\begin{aligned} i &\equiv 2 \\ j &\equiv 4 \\ a &\equiv 2 \\ b &\equiv 3 \end{aligned}$$

First compute $Hy \equiv s \pmod{11}$.

$$\begin{aligned} s_1 &= 2 + 3 = 5 \\ s_2 &= 2\,(2) + 3\,(4) = 16 \equiv 5 \\ s_3 &= 2\,(4) + 3\,(16) \equiv 8 + 15 \equiv 1 \\ s_4 &= 2\,(8) + 3\,(64) \equiv 5 + 3\,(-2) \equiv -1 \equiv 10 \end{aligned}$$

Plugging into the expressions for p, q, and r:

$$\begin{aligned} p &= s_2^2 - s_1 s_3 = 25 - 5 \equiv 9 \\ q &= s_1 s_4 - s_2 s_3 = 50 - 5 \equiv 1 \\ r &= s_3^2 - s_2 s_4 = 1 - 50 \equiv -5 \equiv 6 \end{aligned}$$

So the solution to the quadratic equation is

$$\begin{aligned} i, j &\equiv \frac{-q \pm \sqrt{q^2 - 4pr}}{2p} \\ &= \frac{-1 \pm \sqrt{1 - 4\,(9\,(6))}}{2\,(9)} \\ &\equiv \frac{-1 \pm \sqrt{1 - 3\,(6)}}{7} \\ &\equiv \frac{-1 \pm \sqrt{5}}{7} \end{aligned}$$

Now we have the two issues of how to divide , and how to extract a square root in modular arithmetic. For division by 7, solve

$$7x \equiv 1 \pmod{11}$$

Then we can multiply times the multiplicative inverse, x, instead of dividing by 7. The modular equation has has a unique solution by the no zeros, no repeats property of a prime modulus. Here

$$7x \equiv 1 \pmod{11} \Longrightarrow x = 8$$

So instead of dividing by 7, multiply times 8.[3]

Similarly for the square root, we can solve

$$x^2 \equiv 5 \pmod{11}$$

for which there are two solutions: $x = 4$ and $x = 7$. But either solution will produce the same positions i and j, just relabeled.

For $x = 4$:

$$\begin{aligned} i &= (-1+4)\,8 \equiv 2 \\ j &= (-1-4)\,8 = -40 \equiv 4 \end{aligned}$$

For $x = 7$

$$\begin{aligned} i &= (-1+7)\,8 = 48 \equiv 4 \\ j &= (-1-7)\,8 = -64 \equiv 2 \end{aligned}$$

Use either pair to get the error amounts a and b.

$$b = \frac{is_1 - s_2}{i - j} = \frac{2\,(5) - 5}{2 - 4} \equiv \frac{5}{9}$$

Once again to divide by 9 use

$$9x \equiv 1 \Longrightarrow x = 5$$

So

$$b = 5\,(5) = 25 \equiv 3$$

And finally

$$a = s_1 - b = 5 - 3 = 2$$

The solution is as we predicted: an error of 2 in position 2, and error of 3 in position 4.

9.7 generator matrix and the fundamental theorem of linear algebra

As it is a little bit more difficult to construct than in the earlier examples, the generator matrix for the double error correcting code has yet to appear in the discussion. Some consideration of orthogonality and vector subspaces may ease the

[3]Here the equation for the multiplicative inverse can easily be solved by inspection. For larger moduli, Euclid's algorithm, as presented in chapter 10, will yield a solution.

analysis. The generator matrix is composed of the null space to the parity check matrix. The fundamental theorem of linear algebra, then, is useful for describing the construction of the two matrices.

Reconsider the single error correcting code specified by the parity check matrix, H.

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} (\mathrm{mod}\, 2)$$

H consists of 3 linearly independent y element vectors. According to the fundamental theorem the null space will have 4 independent 7 element vectors. That is, the dimension of the nullspace (here, 4) plus the dimension of the row space (3) will equal the 7 dimensions of the vectors. G, as constructed earlier in the chapter, has that form.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} (\mathrm{mod}\, 2)$$

Orthogonality of the two spaces is assured by the parity-generator relationship.

$$HG \equiv 0$$

(For the syndrome $Hy \equiv 0$, we must have $HGx \equiv 0$ for all possible messages, x. Therefore, $HG \equiv 0$ must hold.)

As a tangential observation, there is a difference between orthogonal vectors in Euclidean spaces and spaces defined by modular arithmetic. A vector in the latter space can be orthogonal to itself; that is, the vector product can be zero. For example, all 3 vectors in H have this property. Hence, they belong to both the row space and the nullspace.

One implication of this perhaps seemingly odd state of affairs is that there are not a total of 7 independent vectors in the combination of the rows of H and the columns of G; the rows of H, for example, are not independent of the columns of G. In other words, the space of 7 element vectors is not spanned by the H,G combination, and not all vectors in the 7 element space can be formed by linear combinations of the row and null vectors.

Spanning is not an issue that arises in the coding problem. But it is central for the understanding of other topics we have considered: decomposition of journal entries, uniqueness of arbitrage free prices, and the mutual information theorem, for example. So it doesn't hurt to confront the concept of spanning on occasion.

Return to the double error correction example with

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2^2 & 3^2 & 4^2 & 5^2 & 6^2 & 7^2 & 8^2 & 9^2 & 10^2 \\ 1 & 2^3 & 3^3 & 4^3 & 5^3 & 6^3 & 7^3 & 8^3 & 9^3 & 10^3 \end{bmatrix} \pmod{11}$$

A generator matrix composed of the appropriate number of orthogonal vectors (6 by the fundamental theorem) is

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 10 & 9 & 2 & 1 & 7 \\ 7 & 8 & 7 & 1 & 9 & 6 \\ 9 & 1 & 7 & 8 & 7 & 7 \\ 1 & 2 & 9 & 10 & 4 & 1 \end{bmatrix} \pmod{11}$$

The vectors in G can be found using standard linear techniques. After the 6×6 identity matrix component of G, each vector has 4 unknown elements, for a total of 24 unknowns. Each vector, in turn, must satisfy 4 orthogonality conditions with the rows of H, for a total of 24 equations. 24 independent linear equations with 24 unknowns is a little bit tedious, but once the answer is reduced to modulo 11, the above G appears.

Example 9.11 *Let the message be* $x^t = \begin{bmatrix} 5 & 3 & 0 & 6 & 8 & 0 \end{bmatrix}$. *Use the* G *matrix to append 4 redundant digits rendering the message amenable to double error correction.*

$$y = Gx = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 10 & 9 & 2 & 1 & 7 \\ 7 & 8 & 7 & 1 & 9 & 6 \\ 9 & 1 & 7 & 8 & 7 & 7 \\ 1 & 2 & 9 & 10 & 4 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 0 \\ 6 \\ 8 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 0 \\ 6 \\ 8 \\ 0 \\ 70 \\ 137 \\ 152 \\ 103 \end{bmatrix} \equiv \begin{bmatrix} 5 \\ 3 \\ 0 \\ 6 \\ 8 \\ 0 \\ 4 \\ 5 \\ 9 \\ 4 \end{bmatrix} \pmod{11}$$

It can be verified that the syndrome $Hy \equiv 0$.

9.8 error correction and mutual information

The noisy channel theorem is typically characterized as Shannon's most important result. It laid the foundation for the technical devices of the information age, including smart phones, the internet, and on and on. The theorem is the reason Shannon, himself, if often referred to as the father of the information age.

For the plethora of electronic devices which communicate with each other, the communication channel is inherently error prone; disturbances in the message are common, even typical. Sending the message again doesn't appear to help, as errors occur in the repeated message, as well. And other redundant schemes are subject to the same problem: more symbols in a message means more errors.

Given this state of affairs, it didn't seem like a good idea to even try for error free communication through a noisy channel. But Shannon's noisy channel theorem changed everything. First, he *proved* error free transmission was possible through a noisy channel. He showed redundancy could work to bring the error rate arbitrarily close to zero. And he also proved how much redundancy was necessary to eliminate the errors.

His proof, however, was abstract and not constructive. that is, it didn't specify how to construct an error free code; it just said some must exist. But that was enough to get coding theorists to explore the field, and several efficient codes have been developed. Since Shannon had shown the minimum redundancy needed, actual codes are compared to the Shannon bound, and many, indeed, approach the bound.

We won't go through the theorem here, but we will show how error correcting codes like we have studied can significantly reduce error rates. Mutual information is the relevant measure of error rates. In particular, we know

$$0 \leq I(X;Y) = H(Y) - H(Y|X) \leq H(Y)$$

Dividing both sides by $H(Y)$ we have

$$0 \leq \frac{I(X;Y)}{H(Y)} \leq 1$$

The closer the ratio to one, the closer the communication is to error free. While Shannon showed a ratio of one is (virtually) possible, we will be content with illustrating how redundant codes can move the ratio closer to one.

Example 9.12 *Consider three possible messages, x, say 0, 1, or 2, with a 10% error rate. That is, if x=1 is sent, the received message, y, will be 1 (correct) with probability .9; 0 or 2 will be received each with probability .05. Messages of 0 or 2 will behave the same way. Compute the mutual information ratio*

$$\frac{I(X;Y)}{H(Y)}$$

if one symbol at a time is sent. Then compute the ratio sending 4 symbols at a time using the linear code with parity check matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix} \bmod 3$$

(Looking ahead, the ratio for one symbol at a time is about 64%. For the 4 symbol block code the ratio increases to 84%.)

First one symbol at a time. The conditional probabilities are as follows where x is the message sent, and y the message received.

		y		
	$p(y\|x)$	0	1	2
x	0	.9	.05	.05
	1	.05	.9	.05
	2	.05	.05	.9

The joint probabilities are used for the entropy computations; and to access the joints, the marginal probabilities, $p(x)$, are used. But the marginals are a decision variable in the code design problem. That is, the code designer can choose the states of the world which generate the message x_i, effectively choosing the probabilities $p(x)$. The objective here, then, is to choose the marginal probabilities which yield the largest mutual information ratio $I(X;Y)/H(Y)$.

The example has a symmetric conditional probability matrix, and, for that case, it is particularly simple to specify the optimizing marginal distribution $p(x)$. The answer is to choose p(x) to be uniform over the possible messages. For the example, $p(x_i) = 1/3$ for all i.

Theorem 9.2 *For a symmetric channel, that is, one where the conditional probability matrix $p(y|x)$ is symmetr4ic, the maximum information ratio is achieved with a uniform distribution on the messages, x.*

A version of the theorem is in Cover and Thomas on page 190 (theorem 8.2.1). Its application in numerical examples can easily be illustrated using a spreadsheet optimizer: when asked to maximize the information ratio, the optimizer returns a uniform marginal distribution. It is also instructive to query the optimizer when the channel conditional probabilities are not symmetric. Then, in general, uniformity is not optimal.[4]

Converting to joint probabilities using uniform marginals:

		y		
	$p(x,y)$	0	1	2
x	0	.3	$\frac{.05}{3}$	$\frac{.05}{3}$
	1	$\frac{.05}{3}$	.3	$\frac{.05}{3}$
	2	$\frac{.05}{3}$	$\frac{.05}{3}$	.3

[4]It is tempting to refer to the theorem as the "dog theorem," as it is illuminated by a semi famous Far Side cartoon by Gary Larson in which all dog barks are decoded as "hey." The dogs could be more informative if they "spread out" their messages.

Using the joint probabilities, $p(x,y)$, mutual information $I(X;Y)$ is computed the usual way.

$$\begin{aligned} H(X) &= H(Y) = \ln 3 \\ H(X,Y) &= -\left[.9\ln\frac{3}{10} + .1\ln\frac{1}{60}\right] \\ &= .9\ln 2 + .9\ln 5 - .9\ln 3 + .2\ln 2 + .1\ln 5 + .1\ln 3 \\ &= 1.1\ln 2 - .8\ln 3 + \ln 5 \end{aligned}$$

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ &= 2\ln 3 - 1.1\ln 2 + .8\ln 3 - \ln 5 \\ &= 2.8\ln 3 - 1.1\ln 2 - \ln 5 \end{aligned}$$

The mutual information ratio is

$$\frac{I(X;Y)}{H(Y)} = \frac{2.8\ln 3 - 1.1\ln 2 - \ln 5}{\ln 3} \simeq .641$$

Approximately 64% of the uncertainty in the sent message, y, is reduced by the receipt of the signal, x.

Before proceeding to using an error correcting code in the noisy channel, let's simplify the information ratio expression for the symmetric channel case. The general conditional probability matrix is

$$\begin{matrix} p & (1-p)/(n-1) & \cdots \\ (1-p)/(n-1) & p & \\ \vdots & & \ddots \end{matrix}$$

where p is the error free rate of transmission, and n is the number of possible messages. Invoking uniform marginals, the general joint probability matrix is

$$\begin{matrix} p/n & (1-p)/n(n-1) & \cdots \\ (1-p)/n(n-1) & p/n & \\ \vdots & & \ddots \end{matrix}$$

By symmetry

$$H(X) = H(Y) = \ln n$$

And the joint entropy:

$$\begin{aligned} H(X,Y) &= -\left[p\ln\frac{p}{n} + (1-p)\ln\frac{1-p}{n(n-1)}\right] \\ &= \ln n + H(p) + (1-p)\ln(n-1) \end{aligned}$$

$$\text{where } H(p) = -[p\ln p + (1-p)\ln(1-p)]$$

Therefore,

$$\begin{aligned}\frac{I(X;Y)}{H(Y)} &= \frac{H(X)+H(Y)-H(X,Y)}{H(Y)} \\ &= \frac{\ln n - H(p) - (1-p)\ln(n-1)}{\ln n}\end{aligned}$$

For the example so far we have $n = 3$ and $p = .9$. Plugging into the expressions yield

$$H(p) = H(.9) \simeq .325$$

and

$$\frac{I(X;Y)}{H(Y)} = \frac{\ln 3 - .325 - .1\ln 2}{\ln 3} \simeq .641$$

as before.

The next step is to use the error correcting code, and see if the amount of information passing through the noisy channel increases. Instead of one symbol at a time, the linear code will send blocks of 4 symbols: 2 symbols in the original message and 2 redundant. The generator matrix for the code is

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2 & 2 \\ 2 & 1 \end{bmatrix} \bmod 3$$

So, for example, if the original message is

$$y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

the 4 symbol block sent is

$$x = Gy = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} \bmod 3$$

The error correcting code will correct any single error, so the probability the message gets through the channel unchanged is the sum of the probability of no errors plus the probability of one error.

$$\begin{aligned}\text{Prob(no error)} &= .9^4 = .6561 \\ \text{Prob(one error)} &= .9^3.1^1\,(4) = .2916\end{aligned}$$

The general expression for t errors is

$$\text{Prob(t errors)} = .9^{4-t}.1^t\binom{4}{t}$$

where the third term in the expression is the number of ways the error can occur in the 4 symbol block.

The nine by nine conditional probability matrix can be constructed. On the diagonal is the probability of error free transmission

$$p = .6561 + .2916 = .9477$$

And, continuing with the symmetry frame, assume there is no information about y if more than one error occurs. Then the off-diagonal consists of

$$\frac{1 - .9477}{8} = .0065375$$

The entropy computations can be made from the symmetric probability matrices, or the more general relationships can be used for

$$p = .9477 \text{ and } n = 9$$

$$H(p) = H(.9477) \simeq .205$$

and

$$\frac{I(X;Y)}{H(Y)} = \frac{\ln 9 - .205 - .0523 \ln 8}{\ln 9} \simeq .857$$

As predicted in the statement of the example, the fraction of information getting through the channel increased from 64% to approximately 86% by using the single error correcting code. The redundancy rate was 50% since each 4 element block has 2 redundant symbols.

Example 9.12 began with a symbol error rate of 10% or, equivalently an error free transmission rate of $p = .9$. For different error free transmission rates, an improvement in the mutual information ratio also occurs. Below are tabulated some symbol error free rates along with information ratios for single symbol transmission and block transmission using the single error correcting of the example.

symbol error free rate p	.99	.975	.95	.9	.8
$\frac{I(X;Y)}{H(Y)}$ single transmission	.943	.878	.788	.641	.418
$\frac{I(X;Y)}{H(Y)}$ block transmission	.997	.986	.953	.857	.614

Shannon's powerful theorem states there exists a block code which achieves an error rate arbitrarily close to zero, but does not supply the code. For our purposes we will be satisfied with illustrating error correction and documenting improvements in information transmission rates.

9.9 summary

This chapter is the first to deal with the issue of how to preserve data integrity. The error detecting and correcting codes presented herein rely on two primary

academic tools. One is orthogonality and the concept of the nullspace, a tool used extensively in prior chapters. The other tool is number theory which will prove quite useful in subsequent chapters. In this chapter we got our first exposure to prime numbers and the fundamental theorem of arithmetic. We also presented a numerical example which improved the information transfer in a noisy channel, though not as much as Shannon's noisy channel theorem states is possible.

9.10 references

Cover, Thomas M., and Joy A. Thomas, *Elements of Information Theory*. John Wiley and Sons, 1991.

Hill, Raymond, *A First Course in Coding Theory*. Oxford University Press, 1986.

Ore, Oystein, *Number Theory and Its History*. McGraw-Hill Book Company, 1948.

9.11 exercises

Exercise 9.1 *Here are some ISBN's - possibly in error. Check for accuracy.*

1. *Probability Theory* by E. T. Jaynes. 978-0-521-59271-0
2. *Essays in Accounting Theory in Honour of Joel S. Demski*. 0-387-30397-9 and 978-0387-30397-0
3. *Number Theory and Its History* by Oystein Ore. 0-486-65620-9

Exercise 9.2 *Fill in the missing digits in the following UPC numbers.*

$$\begin{array}{cccccccccccc} 0 & 7 & 0 & 5 & 5 & 4 & 0 & 0 & ? & 2 & 3 & 6 \\ 0 & 2 & 8 & 4 & 0 & 0 & 0 & 9 & ? & 9 & 0 & 1 \end{array}$$

Exercise 9.3 *Consider a "perfect" single error correcting code with parity check matrix*

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 \end{bmatrix} (\text{mod } 7)$$

For the following received codewords, detect and correct a single error, if necessary.

$$\begin{array}{cccccccc} 1 & 5 & 3 & 6 & 1 & 3 & 1 & 6 \\ 3 & 2 & 2 & 0 & 0 & 3 & 1 & 2 \end{array}$$

Exercise 9.4 *For the same modulo 7 code as in the previous problem, append the appropriate redundant digits.*

$$\begin{array}{cccccc} 1 & 2 & 4 & 3 & 6 & 6 \\ 5 & 6 & 1 & 1 & 2 & 1 \end{array}$$

Exercise 9.5 *Using the same modulo 7 code in the previous exercises, verify the "perfectness" of the same code by computing the number of possible received codewords, the number of legal codewords, and the number of codewords within one change of a legal codeword.*

Exercise 9.6 *Consider Matthew 5:37. "Let your communication be Yea, yea; Nay, nay: for whatsoever is more than these cometh from evil." The communication is an implied binary code. What is the implied parity check matrix, and the generator matrix? Is the code error correcting? What about error detection?*

Exercise 9.7 *Consider a noisy symmetric channel which transmits one bit at a time, either a zero or a one. The error rate is 10%, or, alternatively, the error free transmission rate is 90%. What is the mutual information ratio for the channel? Now use the single error correcting with parity check matrix*

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \bmod 2$$

What is the mutual information ratio for the 7 element block?

Exercise 9.8 *Consider the double error correcting code in the chapter with parity check matrix*

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2^2 & 3^2 & 4^2 & 5^2 & 6^2 & 7^2 & 8^2 & 9^2 & 10^2 \\ 1 & 2^3 & 3^3 & 4^3 & 5^3 & 6^3 & 7^3 & 8^3 & 9^3 & 10^3 \end{bmatrix} \pmod{11}$$

Suppose the syndrome is $Hy = s \equiv \begin{bmatrix} 3 & 3 & 3 & 3 \end{bmatrix}^T$. *What are the error(s) and position(s) thereof? Suppose* $Hy = s \equiv \begin{bmatrix} 10 & 1 & 2 & 8 \end{bmatrix}^T$.

Exercise 9.9 *Redo example 9.12 with symbol error rate of .025. That is, with probability .0125 one of the incorrect symbols is received, and the same probability for the other incorrect symbols.*

Exercise 9.10 *Consider a double error correcting code with the following parity check matrix.*

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2^2 & 3^2 & 4^2 & 5^2 & 6^2 \\ 1 & 2^3 & 3^3 & 4^3 & 5^3 & 6^3 \end{bmatrix} \pmod 7$$

Suppose the syndrome is $Hy = s \equiv \begin{bmatrix} 4 & 0 & 2 & 1 \end{bmatrix}^T$. *What are the error(s) and position(s) thereof?*