Error Correcting Codes

2. The Hamming Codes

Priti Shankar

Priti Shankar is with the Department of Computer Science and Automation at the Indian Institute of Science, Bangalore. Her interests are in Theoretical Computer Science.

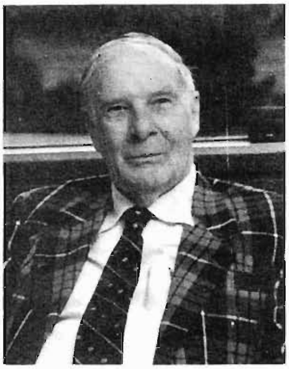
In the first article of this series we showed how redundancy introduced into a message transmitted over a noisy channel could improve the reliability of transmission. In this article we describe one of the earliest linear codes invented for this purpose, and show how the algebraic structure of the code enables easy decoding.

Introduction

Imagine two individuals, call them the *sender* and the *receiver*. The sender wishes to transmit a sequence of binary digits across a medium called a *channel* to the receiver. If she sends a one, a one will probably be received, and if she sends a zero, a zero will probably be received. The channel, however, is not totally reliable. Therefore, occasionally, a transmitted one will be received as a zero, and a transmitted zero as a one. The sender is unable to prevent the channel from making such errors. Can something be done to reduce their effects?

It was such a problem that confronted Richard W Hamming in 1947. Hamming was then at Bell Telephone Laboratories, New Jersey, and part of a Mathematics Department that included Claude Shannon, now widely acknowledged as the father of information and coding theory. A large scale relay computer had failed to deliver the expected results due to a hardware fault. Hamming, one of the active proponents ofcomputer usage, was determined to find an efficient means by which computers could detect and correct their own faults. A mathematician by training, Hamming soon came up with a solution based on parity checking. Adding on extra check bits to a block of data would allow not only detection of faulty bits, but their location as well.

His techniques for finding and correcting a single error in a block of data, as well as detecting two errors and correcting a single error, were to become known as the*Hamming codes*. The codes were used by Bell Labs in their computers and in their switching systems.



Richard Hamming

The problem described in the opening paragraph could well be encountered in a communication system, where communication is over *time* instead of over*space*. For example, a computer memory prone to errors, could be the unreliable channel. Storage of data into memory could be the sending process, and the reading of data from memory the receiving process. Protecting data in computer memories was one of the earliest applications of Hamming codes. We now describe the clever scheme invented by Hamming in 1948. To keep things simple, we describe the binary length 7 Hamming code.

Encoding in the Hamming Code

In a code where each codeword contains several message bits and several check bits, each check bit must be some function of the message bits. In the Hamming code each check bit is taken to be a mod 2 sum of a subset of the message bits. Assume that the rate of the code is 4/7, so that for every four information symbols transmitted, there are three check symbols introduced in the codeword. Call these the *parity check symbols.* It is customary to index the bits from left to right beginning with 0. Let $\mathbf{c} = (c_0, c_1, \dots c_6)$ denote the vector representing a seven bit codeword, with the first four bits being information bits.

Techniques for finding and correcting a single error in a block of data, as well as detecting two errors and correcting a single error, were to become known as the Hamming codes.

If c_0, c_1, c_2 and c_3 are the information symbols, let us define the check symbols c_4, c_5, c_6 as follows:

$$c_4 \equiv c_0 + c_1 + c_2 \ (mod\ 2)$$
$$c_5 \equiv c_0 + c_1 + c_3 \ (mod\ 2)$$
$$c_6 \equiv c_0 + c_1 + c_3 \ (mod\ 2)$$

In a code where each codeword contains several message bits and several check bits, each check bit must be some function of the message bits.

Thus, if the message bits are 1010, then the codeword is 1010011. After the message sequence is encoded into the codeword **c**, the codeword is transmitted across the noisy channel. The channel adds to the codeword, an *error pattern* $\mathbf{e} = (e_0, e_1, \ldots e_6)$ to yield the received pattern $\mathbf{r} = \mathbf{c}+\mathbf{e}$. Thus, for example, if the bit with index 4 of the codeword above is garbled by the channel, the error pattern is 0000100, and the received pattern is 1010111.The decoder then has to estimate the original message from the garbled codeword.

Let us rewrite the equations above as

$$c_0 + c_1 + c_2 + c_4 \equiv 0 \ (mod\ 2\ )$$
$$c_0 + c_1 + c_3 + c_5 \equiv 0 \ (mod\ 2\ )$$
$$c_0 + c_2 + c_3 + c_6 \equiv 0 \ (mod\ 2\ )$$

Every codeword satisfies these equations. Therefore if $\mathbf{c} = (c_0, c_1, c_2, c_3, c_4, c_5, c_6)$ is a codeword, then the matrix equation below

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

is just a restatement of the three equations above.

The first matrix on the left hand side is called the *parity check* matrix H. Thus every codeword **c** satisfies the equation

$$H\mathbf{c}^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Therefore another way of describing the code is by specifying its parity check matrix H. Note that the seven columns of the parity

Things to Remember

Binary block code: A collection of codewords. Each codeword is a fixed length pattern of 0's and 1's.

k: The number of message bits in each codeword.

r: The number of check bits in each codeword.

n: The block length of the code. $n = k + r$

Information Rate of a code: The ratio k/n

Hamming distance between codewords: The number of corresponding positions in which the codewords differ.

Minimum distance of a code : The minimum of the Hamming distances between all pairs of codewords.

Hamming weight of a vector : Number of non-zero components of the vector.

Result : A linear code can correct all error patterns of Hamming weight t or less if and only if it has a minimum distance at least $2t+1$. It can detect all error patterns of weight d or less if and only if it has a minimum distance at least $d+1$.

check matrix are the seven *distinct* non zero combinations of three bits.

The Hamming code of length 7 is an example of a linear algebraic code. In the last article, we mentioned that a binary code of length n was a subspace of the space of all vectors with n components, over the field $F_2 = \{0,1\}$, i.e. each component being either a zero or a one. Since this Hamming code has four information symbols, and as the check symbols are completely determined by the information symbols, the subspace has $2^4 = 16$ vectors. These sixteen vectors constitute the code, which is itself a subspace of the vector space containing a total of $2^7 = 128$ vectors, each of length 7 bits.

Syndrome Decoding

Given the received pattern $\mathbf{r}$, the decoder must eventually decide what the transmitted codeword was. If the decoder is able to find the error pattern $\mathbf{e}$, then the codeword is $\mathbf{c} = \mathbf{r} + \mathbf{e}$. To estimate $\mathbf{e}$, it first forms the product $H\mathbf{r}^T = H\mathbf{c}^T + H\mathbf{e}^T = H\mathbf{e}^T$ This product is called the *syndrome*, and reveals the pattern of parity check *failures* on the received pattern.

The *syndrome*, reveals the pattern of parity check *failures* on the received pattern.

Suppose a single error has occurred during transmission. Then the error pattern will have a single 1 in the position in which the error has occurred and zeros everywhere else. Thus the product $H\mathbf{e}^T$ will be the i^{th} column of H if i is the error position. Since the seven columns of H are all the distinct non-zero combinations of three bits, each of the seven possible single errors will have a distinct syndrome which will uniquely identify it. For example, if the bit with index 4 is in error, the error pattern is 0000100, and the syndrome will be 100 (the column of H with index 4, if the columns are indexed beginning with 0). Since the single error syndromes by themselves exhaust all the seven possible non-zero syndromes (there are only seven non-zero combinations of three bits), any double error will necessarily have the same syndrome as that of a single error. For example, the double error pattern 0010001, among others, also gives the syndrome 100. How does the decoder decide which error pattern to choose?

Under the assumption that the probability of the channel flipping a bit during transmission is less than 1/2, and that bit errors occur independently of one another, the more probable error pattern is the one with fewer 1's. Thus the decoder follows what is known as a *maximum likelihood* strategy and decodes into the codeword that is at the smallest Hamming distance from the received pattern. In other words, even if there is a double error, the decoder will mistake it for a single error, as there will be a single error with an identical syndrome, which the maximum likelihood strategy will choose as its estimate of the error pattern. Thus if this Hamming code is used for single error correction, it cannot correctly *detect* double errors.

Decoding in the Hamming code then consists of the following three steps:

1. Form the syndrome $H\mathbf{r}^T$ from the received vector **r**.

2. If the syndrome is the all zero vector, assume no errors have occurred. If it is not, then find out which column of H the

syndrome matches. If the column index is i, then the estimated error pattern $\hat{e}$ is the vector with a 1 in the ith position and zeros everywhere else.

3. Form $\mathbf{c} = \mathbf{r} + \hat{\mathbf{e}}$ as the decoder's estîmate of the transmitted codeword.

Problem

For the length 7 Hamming code defined above, decode the received vectors $\mathbf{r} = (1110000)$, $\mathbf{r} = (1111000)$.

The Minimum Distance of a Code

The last result in the table displayed indicates that since this Hamming code can correct all single errors, the minimum distance of the code must be at least three. Actually we can deduce that there is a codeword of Hamming weight three from the example given earlier, where flipping one bit in a codeword could give the same syndrome as is obtained by flipping two other bits in some other codeword. For if $\mathbf{e}_1$ and $\mathbf{e}_2$ are the single and double error patterns respectively, that give the same syndrome, then $H\mathbf{e}_1^T = H\mathbf{e}_2^T$ or $H(\mathbf{e}_1+\mathbf{e}_2)^T = 0$. Thus $\mathbf{e}_1+\mathbf{e}_2$ is a codeword with exactly three 1's i.e. it has Hamming weight three, and is at distance three from the all zero codeword. The parity check matrix of a code provides us a means of determining a lower bound on the minimum distance of the code, without either having to examine all pairs of codewords or looking at syndromes.

The parity check matrix of a code provides us with a means of determining a lower bound on the minimum distance of the code, without either having to examine all pairs of codewords or looking at syndromes.

If $\mathbf{u}$ and $\mathbf{v}$ are vectors, then the distance between them is the Hamming weight of $\mathbf{u} + \mathbf{v}$ (where + is a bitwise operation), which is also a codeword by the linearity condition. Thus for a linear code, the minimum distance is the minimum Hamming weight of a non zero codeword. Since every codeword $\mathbf{c}$ satisfies $H\mathbf{c}^T = 0$, if a codeword of the Hamming code had weight two, then two of the columns of H would sum up to the all zero

A code is normally characterized by three parameters: the length, the number of information symbols, and the minimum distance.

vector. But this can happen only if the columns are identical. Since all columns are distinct, the weight must be at least three. We have thus found a rule to obtain a lower bound on the minimum distance of a linear code from its parity check matrix H.

If no linear combination of d or fewer columns of H gives the all zero column vector, then the code has minimum distance at least $d + 1$.

A code is normally characterized by three parameters. These are the length n, the number of information symbols k, and the minimum distance d. A code with these parameters is called a (n, k, d) code.

Binary single error correcting Hamming codes can be defined for any length $2^m - 1$. The parity check matrix of such a code will have m rows and $2^m - 1$ columns consisting of all non-zero binary combinations of m bits. By analogy with the codes of length 7 described above, we can deduce that all these codes have minimum distance three. Thus one can define Hamming codes with parameters (15,11,3), (31,26,3), (63,57,3) and so on. One can check that the information rate goes up as the length of the code increases, for the same error correcting capability.

The Hamming codes have the beautiful geometric property that the spheres containing all vectors at distance 1 from codewords, will be non-intersecting. In fact, the spheres cover the whole space, that is, they together contain *all* the vectors in the space. This generally does not happen for all codes. The Hamming codes happen to belong to a very exclusive class of codes called *perfect codes.*

Problem

Show that the geometric properties mentioned above hold for the binary, length $2^m - 1$ Hamming code, i.e. spheres of radius 1 around the codewords exactly fill the vector space.

The Extended Hamming Codes

We can modify the length $2^m - 1$ Hamming codes slightly, to overcome the limitation of not being able to distinguish between single and double errors.The *extended* Hamming code of length 2^m is defined to be the code obtained from the original Hamming code by adjoining an overall parity-checkbit. If H is the original parity check matrix, the one for the extended code is

$$H' = \begin{array}{c|c|} \cline{2-2} 0 & \\ 0 & \\ & H \\ & \\ 0 & \\ \cline{2-2} 1 & 1 \quad 1 \qquad\qquad 1 \end{array}$$

Problem

Show that the minimum distance of the extended Hamming codes is 4.

In the example of the length 7 Hamming code described above, the decoder decodes every possible received pattern into some codeword, that is, it is a *complete* decoding algorithm. For the extended Hamming code of length 8 and minimum distance four (obtained by setting $m = 3$ in the construction above), we might consider using an alternate *incomplete* decoding algorithm, where the decoder either attempts to correct an error or stops at detection. In some applications, a decoding error (like the one our decoder for the Hamming code of length 7 makes for a double error) may be disastrous, as it may result in an incorrect command being received by a spaceship. However, a decoding failure may result in the command being ignored, and can be overcome by asking for retransmission. Thus one can have a

Some useful code classes (along with the year of discovery).

- Reed Solomon(RS) codes (1960)

Perhaps the most widely used codes today. They were used in the Voyager II deep space probe. They are also used in compact disc players. A key factor responsible for their widespread use is the invention of a clever decoding algorithm by Elwyn Berlekamp in 1966.

- Bose-Chaudhuri-Hocquenghem codes(BCH codes) (1959)

They have good error correcting properties when the length is not too large, and have easy encoding and decoding techniques. They are used in mass storage devices in computer systems. RS codes can be viewed as generalizations of BCH codes.

- Convolutional codes (1955)

These are distinct from the block parity-check codes described in this article, as adjacent blocks of size n are interdependent. They are used extensively in space communication, sometimes in conjunction with RS codes. The most famous decoding technique for these is the Viterbi decoding technique proposed by Andrew Viterbi in 1967.

- Goppa codes (1970)

An elegant natural generalization of BCH codes. In 1975, Y Sugiyama, M Kasahara, S Hirasawa and T Namekawa discovered the beautiful fact that Goppa codes can be decoded by using Euclid's algorithm for finding greatest common divisors (applied to polynomials).

- Algebraic Geometry codes (1981)

A class of codes introduced by V D Goppa, and greatly enriched through the 80's and 90's. The introduction of this class led to a new lower bound on the information rate of good codes.

decoding scheme where the decoder intentionally refuses to decode any sufficiently ambiguous received pattern.

For the extended Hamming code of the example above, every single error will have a 1 in the last component of its syndrome, as every column of H has a 1 in its bottom row. Also, the syndrome for any double error will have a 0 in the last component. Since columns of the parity check matrix correspond to syndromes for single errors, and since all columns are distinct, syndromes for single errors are non-zero and distinct, and different from syndromes for double errors. Thus if one uses an extended Hamming code for transmission, one can have a decoding scheme that reports a decoding failure for double errors, and corrects all single errors. Step 2 of the decoding algorithm above would be modified as follows: If the syndrome is the all 0 vector, then assume that no errors have occurred. If the syndrome matches the i^{th} column of H, then estimate the error as the pattern with a 1 in position i and zeros everywhere else. If the non-zero syndrome does not match any column of H then report a decoding failure, i.e. the detection of an uncorrectable error.

Since *all* double errors have non-zero syndromes, the extended Hamming code is said to be *single error correcting double error detecting*, (that is, it can do both these things simultaneously). Note the difference between *this* Hamming code and the length 7 Hamming code. If the latter code was used for single error correction, it could not *simultaneously* detect double errors as the syndromes for single and double errors were not distinct. Similarly, the length 8 Hamming code cannot simultaneously correct single errors and detect *triple* errors.

If we take a geometric view of this length 8 Hamming code, all spheres of Hamming radius 1 around codewords will be non-intersecting. Also, any sphere of Hamming radius 2 around a codeword will not intersect a sphere of Hamming radius 1 around any other codeword. (The spheres of Hamming radius 2 around codewords however, will intersect one another).

Problem

For the length 8 Hamming code above, show that the spheres of Hamming radius 2 around codewords are not disjoint.

Single error correction is obviously not enough for most applications. In the next article we will study the powerful Reed Solomon codes which were invented in 1960. Though these started off as mathematical curiosities, a clever decoding algorithm invented about eight years later has made them perhaps the most practical and widely used codes.

Suggested Reading

- Robert J McEliece. The Theory of Information and Coding. *Encyclopedia of Mathematics and its Applications.* Addison Wesley, 1977.
- J H Van Lint. Introduction to Coding Theory. *Graduate Texts in Mathematics.* Springer-Verlag, 1992.

Address for Correspondence
Priti Shankar
Department of Computer
Science and Automation
Indian Institute of Science
Bangalore 560 012, India
email:priti@csa.iisc.ernet.in
Fax: (080) 334 1683