ALGEBRAIC CODING TECHNIQUES FOR ERROR DETECTION IN DATA TRANSMISSION

By: Nasaruddin Zenon
ITM Cawangan Pahang

ABSTRACT:

The purpose of this paper is to describe currently available algebraic coding techniques for error detection in computer communication and the problems associated with them. We will first discuss the simplest technique being employed followed by complicated ones involving cyclic codes. We will also introduce a simple comparison method for information flow rate in data transmission. This method can be used to compare all (n, k) - codes with a fixed error rate.

1.0. INTRODUCTION

In computer communication, it is vital that all messages be delivered as error-free as possible. A single bit error in a message in processes such as the transmission of computer programs and file transfer can have serious consequences. However experience shows that it is not easy to build equipment that is highly reliable even with the advancement in computing technology today. Furthermore, communication systems are limited in their performance by the available signal power, the inevitable background noise, and the need to limit bandwidth (TAUB). Error detection techniques are developed to check the presence of noise (error) that is usually present whenever messages are communicated over a distance via a medium.

The fundamental method of data transmission in an electronic medium is the conventional signaling system which is shown in Figure 1.0 (HAMMING):

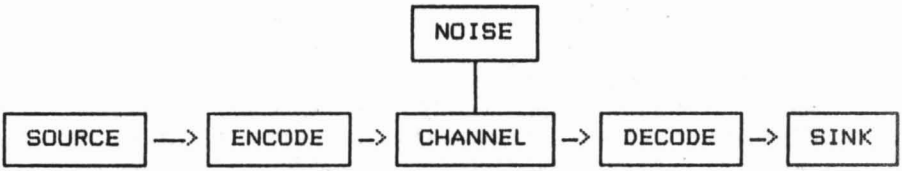

Figure 1.0. Conventional Signaling System

SOURCE referred to here is the information to be transmitted in a channel or a medium. The ENCODE is broken up into two stages, one the encoding of the source and the other the further encoding to fit the channel. In computing terms, this process corresponds to representing the characters, typed in at the keyboard, with binary digits of 0's and 1's by the ocmputer. Before being transmitted over a telephone line, these binary digits are futher converted into analog signals by a modem.

The DECODE block in the figure is a process of recovering the original information that have passed through the transmission medium. It is here that any modification of the data stream by some noise in the media channel is detected before the message finally reaches its destination.

This paper deals with the available techniques being used to code and decode messages into bit strings at the ENCODE and DECODE stages of the signaling system. However, we will not discuss transmission errors caused by human interception as these fall into the category of data security and involve the study of cryptology as a whole.

2.0. Parity Checks Technique

The simplest of all error detection techniques is the single parity check technique. An extra bit is appended to the character code to be transmitted for error detection. For example, in the ASCII character code, characters are mapped into strings of seven bits and then a parity check bit is appended as an eighth bit which we will call *cb* (please refer to (Figure 2.0.).

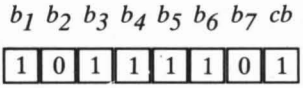

Fig. 2.0. Single Parity Check,

The final bit *cb* is a parity check bit which is the modulo 2 sum of b_1 through b_7. In other words, this parity check bit has the value 1 if the number of 1's in the bit string is odd, and has the value 0 otherwise. The entire message is therefore of even parity. (For simplicity, we will only consider even parity check code.) At the receiving end (DECODE), the count of the number of 1's is made, and an odd number of 1's in the strings indicates that at least one error has occurred.

Despite the remarkable simplicity of the single parity check, it is still inadequate for reliable detection of errors for the following reasons:

1) For an even check bit it can only detect odd number of errors.

2) If the probability of an error in any binary possition is $p < 1$, and errors in different positions are assumed to be independent, then for $n << 1/p$, the probable number of single error $P(E_i) = np$. The probability of a double error $P(E_i \mid E_j) = n(n-1)p^2/2$ where i and j are the positions of errors. Simply stated, the single parity check technique only detects errors in about half of the encoded strings where errors occur!

3) Dimitry and Gallager pointed out that;
In physical situation, this poor behavior is exaggerated by the fact that many modems map several bits into a single sample of the physical channel input, and an error in the reception of such a sample typically causes several bits errors. Also, many kinds of noise, such as lightning and temporarily broken connection, cause long bursts of errors rendering a single parity check as ineffective due to an even number of errors occurring almost as likely as odd number ones (DIMITRY).

Although single parity check technique is not very reliable in detecting errors in transmission, it is rather universally used in digital computers for error detection. This method is effective in an application that has a low probability of errors in one bit. An example of this is the checking of memory chips in a computer system. Parity check bit technique also provides us a foundation upon which a generalization of arbitrary parity check codes can be developed.

3.0. Improvements On The Parity Check Bits.

We have seen how the simple and intuitive approach of adding a check bit at the end of a message string bit can provide us an idea for developing useful coding techniques. Questions that still need to be addressed are:

1) How many of these check bits are needed in order to detect more errors in a message?

2) How does the number of check bits used affect the probability of errors occuring in a message?

3) Can we obtain a code that allows us to increase the rate at which information may be transmitted through a channel while maintaining a fixed error rate?

It turns out that the first two questions can be answered by polynomial representations of binary digits which lead us to what is known as Cyclic Redundancy Checks or simply CRC method. The method is presently implemented in most Data Link Control (DLC) in computer networks today. However, the answer to the third question requires us to study deeper into *channel capacity* (the maximum reliable data rate in bps) and Shannon's Theorem. This will lead us to the Golay Code and the Bose, Chaudry and Hocquenghen (BCH) Codes that are still being studied by coding and information theorists worldwide.

4.0. Polynomial Representations of *n* binary digits.

Definition 4.1. An *(n,k)* - code is a binary-coded message in blocks of *k* digits. Each block is encoded with *(n-k)* digits of binary check bits.

In an *(n-k)* - code there are 2^n possible words that could be received of which 2^k are code words.

Definition 4.2. A code rate or information rate, R in an *(n,k)* - code is defined to be the number of redundant digits divided by the number of possible words or, simply, $R = k/n$.

With the given definitions we can represent a word *n* binary digits by means of a polynomial in $Z_2[x]$. ($Z_n[x]$ is a set of integers modulo n, of degree less than n.) The binary bits $a_0 a_1 \ldots . a_{n-1}$ can be represented by the polynomial

$$a_0 + a_1 x + \ldots . + a_{n-1} x^{n-1} \in Z_2[x].$$

Definition 4.3. Let $p(x) \in Z_2[x]$ be a polynomial of degree $n-k$. The *polynomial code generated by* p(x) is an $(n-k)$ - code. The code words formed are polynomials of degree less than n which are divisible by p(x).

A message of length *k* is represented by a polynomial s(x), of degree less than *k* where

$$s(x) = s_{k-1} x^{k-1} + s_{k-2} x^{k-2} + \ldots . + s_0$$

The coefficients $s_{k-1}, s_{k-2}, \ldots . , s_1, s_0$ which denote the data bits have the value of either 0 or 1.

The parity check bits are called the CRC with length $l = n-k$. The CRC can be represented by another polynomial

$$c(x) = c_{l-1}x^{l-1} + c_{l-2}x^{l-2} + \ldots\ldots + c_1 x + c_0.$$

For the higher order coefficients in a code polynomial to be able to carry the message digits, we have to multiply s(x) by x^{n-k}. The effect of this is that the message is shifted n-k places to the right. Since $l = n-k$, the entire frame of transmitted information and the CRC can then be represented by

$$y(x) = s(x)x^l + c(x)$$

$$= s_{k-1}x^{l+k-1} + \ldots\ldots + s_0 x^l + c_{l-1}x^{l-1} + \ldots\ldots + c_0$$

If we devide $s(x)x^l$ by $p(x)$ we will form a polynomial

$z(x) = x^l s(x) + r(x)$ where $r(x)$ is the remainder.

For a given p(x), the mapping from the information polynomial to the CRC polynomial c(x) is given by

$$c(x) = r(x) = \text{Rem}\,[s(x)x^l / p(x)].$$

Now, $z(x) = x^{n-k}\, s(x) + r(x)$ is always a multiple of p(x) because,

$$x^{n-k}\, s(x) = p(x).\, q(x) + r(x) \text{ where } \deg[r(x)] < n\text{-}k \text{ or } r(x) = 0.$$

Hence,

$$\begin{aligned} z(x) &= r(x) + x^{n-k}\, s(x) \\ &= -r(x) + x^{n-k}\, s(x) \\ &= q(x).\, p(x). \end{aligned}$$

$r(x) = -r(x)$ because the coefficients are restricted to be binary and arithmetic is performed in modulo 2.

Polynomial representation of n bits binary number gives us a more generalized coding technique commonly referred to as Cyclic Codes. It turns out many of the earlier coding techniques such as the Hamming Codes and the Golay Codes are special cases of Cyclic Codes. These codes are important because they have algebraic properties which allow them to be easily encoded and ecoded. These codes also have the added property of detecting more errors in message bits. This last statement is proven by the following theorem.

Theorem. If p(x) and q(x) are primitive polynomials of degree m, then for $n \leq 2^m - 1$, *(n,n-m)*-code generated by p(x) detects all single and double errors, and *(n,n-m-1)* - code generated by $p(x) = (1+x)\, q(x)$ detects all double errors and any odd number of errors.

Note: A primitive polynomial is an irreducible polynomial p(x) of degree m over Z_p with the added property $g(x) \mid x^k - 1$ for $k \geq p^m - 1$.

Proof:

Let $z(x) = s(x)x^{n-k} + r(x)$ be the transmitted code word and $a(x) = z(x) + e(x)$ be the received word to be decoded where e(x) is the error polynomial. An error is detectable $\Leftrightarrow p(x) \mid a(x)$. But, since p(x) divides z(x) this implies that it is sufficient to show $p(x) \mid e(x)$.

Note: The symbol $\mid$ means "is divisible by" and $\nmid$ means otherwise.

Case 1: A single eror occurs in a code word.

In this case, e(x) will contain a single term, say x^i, where $0 \leq i < n$. But p(x) is irreducible, i.e. it cannot have 0 as a root; therefore, $p(x) \nmid x^i$, and thus the error x^i is detectable.

Case 2: A double error Occurs

e(x) is of the form $x^i + x^j$ where $0 \leq i < j < n$, i.e., $e(x) = x^i(1 + x^{j-i})$ where $0 < j-i < n$.

But, p(x) being primitive and $p(x) \nmid x^i$

$\Rightarrow p(x) \nmid 1 + x^{j-i}$ if $j-i < 2^m-1$

and, p(x) is irreducible $\Rightarrow p(x) \nmid x^i(1 + x^{j-i})$ for $n \leq 2^m-1$

thus, all double errors are detectable.

The rest of the theorem follows from the above statements and the fact that a polynomial in $Z_2[x]$ has a factor $(x + 1)$ if and only if it has an even number of nonzero coefficients. Thus our theorem is proven.

The importance of the above theorem is that it gives us an idea of how to construct a code that can detect a lot of errors and at the same time the number of messages it will encode is still very large. As mentioned by Gilbert;

... by adding 11 check digits to a message of length 1012 or less using the generator polynomial

$$(1 + x)(1 + x^3 + x^{10}) = 1 + x + x^3 + x^4 + x^{10} + x^{11}$$

we can detect single, double, triple and any odd number of errors. The number of different messages of length 1012 is 2^{1012} , i.e., in base 10, this would be 305 digits, which is an enormous figure (GILBERT).

5.0. Improved Transmission Rate With Constant Errors

The existence of coding techniques that will allow a communication system to transmit information with an arbitrarily small probability of errors is given by the following theorem due to Shannon.

Theorem. There exists a coding technique which is arbitrarily reliable that allows the output of the source to be transmitted over a channel with capacity C provided the information rate R, is such that $R \leq C = B\log_2 (1 + S/ (n_0 B))$, where B is the available badwidth of the channel, S is the allowable signal power as seen by the receiver, and $n_0 B$ is the noise power per unit bandwidth assumed to be uniformly distributed over B.

(For the proof of the theorem, please refer (SLEPIAN].)

The significance of the theorem is that as long as the channel capacity is not exceeded by the number of messages to be transmitted, the probability of errors in the received code word may be made arbitrarily small. The only problem is that a coding technique that will allow the transmission rate to reach the upper bound is yet to be found. Nevertheless, the Shannon's Limit above is widely used to measure the efficiency of a code for data transmission.

Finally, the third question concerning an increase in the flow rate of transmission can be solved by using the most powerful class of techniques of error correction known to date, i.e. the BCH codes.

Definition 5.1. A Galois field of order p^m , denoted by GF (p^m) is a finite field with p^m elements.

Definition 5.2. A t-error-correcting BCH code of length $n = 2^m - 1$ has a generator polynomial p(x) that is constructed as follows. Take a primitive element α in the Galois field GF(2^m). Let $p_i(x) \in Z_2[x]$ be the irreducible polynomial with α^i as a root.

Then,

$$p(x) = LCM(p_1(x), p_2(x), \ldots\ldots, P_{2t}(x))$$

where LCM is the least common multiples.

The number of bits in a code word in the BCH-codes is k + r, where k is the information bits and r is the parity check bits. The number of errors t, which can be corrected in an *(n,k)* - code is r/m where m is given by the relation $n = 2^m - 1$ [BLAKE]. One interesting fact is that the number t is somehow related to the formula R = k/n for information rate. Since r = n-k and k = nR we have r = n - nR = n(1-R). Also, $\log_2 (n+1) = m$ which implies that

$$t = n(1-R)/\log_2 (n+1). \qquad \text{Eq. 1.0}$$

We can easily use this calculation to make comparison of data transmission rates.

Example. The BCH code that corrects three errors is a (31,16)- code generated by

$$p(x) = LCM(p_1(x), p_3(x), p_5(x))$$

$$= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1)$$

$$= x^{10} + x^9 + x^5 + x^4 + x^2 + x + 1.$$

The Golay Code that corrects three errors is a (23,12) cyclic code whose generating function is

$$p(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1.$$

From Eq. 1.0, the information rate is $R = 1 - (t\log_2 (n + 1))/n)$. Substituting n = 31 and n = 23 respectively with t = 3 into the formula we can see that the BCH code gives R = 52% and the Golay code gives R = 40%. Therefore the BCH code of (31,16) gives 12% higher flow rate for transmission than the (23,12) Golay code.

6.0. Conclusions.

One of the advantages of the algebraic coding techniques we have just described is that they allow for alternative techniques which reduce the complexity of decoding information. However these techniques involve considerably deep and subtle results from algebraic number theory. Even with the advancements of computing technology today, we still face the problem of translating

abstract ideas into the design of finite state machine. For example the BCH (135, 101) - code which will correct up to 7 errors adds 34 check digits to the 101 message bits and hence contain 2^{34} *syndromes* (calculations involving parity check matrix). But the question is how to store all these syndromes and their *coset leaders* (most likely error patterns) in a computer of today? It is almost impossible to decode the message by the same encoding scheme. One way to address this problem is to do the decoding by other methods. And finally, a quote from Blake et al;

Any code with a relatively high information rate must be long and consequently, to be useful, must possess a simple algebraic decoding algorithm [BLAKE].

REFERANCES

(BLAKE) *Blake, Ian F. and Mullin, R. C., The Mathematical Theory of Coding, Academic Press, New York, 1975.*

(DIMITRY) *Dimitry, G. and Gallager, R., Data Networks, Prentice Hall Inc., Englewood Cliffs N.J., 1987.*

(GILBERT) *Gilbert, W.J., Modern Algebra with Applications, Wiley, New York, 1976.*

(HAMMING) *Hamming, R.W., Coding and Information Theory, Prentice Hall Inc., Englewood cliffs, N.J., 1980.*

(PETER) *Peterson, W.W. and Brown D.T., "Cyclic Codes for Error Detection,"Proceedings of the Institute of Radio Engineers", 49, 228-235-1961.*

(SLEPIAN) *Slepian, D., ed. Key Papers in the Development of Information Theory, IEEE Press, New York, 1974.*

(TAUB) *Taub, H. and Schilling, D.L., Principles of Communication Systems, Mc Graw - Hill, New York, 1986.*