International Journal of Pure and Applied Mathematics
Volume 120 No. 5 2018, 721-730
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

STUDY THE DESIGN AND IMPLEMENT ERROR-CORRECTING CODES BY SUING MATRIX ALGEBRA

[1] Evan Abdulkareem Huzam, [2] Hasan Ali Naser
[1,2] Thi-Qar University ,Faculty of Education for pure Science,Thi-Qar , Nasiriyah, Iraq
[1]evan_krm@yahoo.com, [2]haalna28@yahoo.com

Abstract

The obliterating redundant statistics of science and art in a smart way such that it can be stored in less space and memory. It still can be expanded to the original message which called compression of data or coding of source. It is not the focus of this study. Therefore, it has been compacted data yet an error occurred in a compacted text would provide a dissimilar message. The codes of error-correcting idea is the converse. One improves redundant info in such way so that it is probable to identify or even correct mistakes after transmission. Additional, via systems of linear algebra over intend, the basic concepts of linear codes is developed such as least distance, dimension, and capabilities of error-correcting. This study focus on indicating the techniques of algebraic applied for improve codes of error detecting or correcting. Most of these techniques are according on standard abstract algebra, mainly the group theory and other algebraic structures. Take for example, vector spaces, rings and matrices on the basics of finite fields. These mathematical concepts are discussed, emphasizing on their relation to coding of error control, as the basis for creating effective codes of error correcting or detecting are provided.

Introduction

The algebraic codes incentive is the improvement of errors in electronic communication which may arise because of the inadequacies in the transmission physical medium. The scenario in which Alice sends the eight bits 01000001 (representing the character `A' in the ASCII character

scheme) is considered as a part of a message to Bob through a noisy transmission line. The fifth bit is flipped and Bob instead receives the string 01001001 because of the random physical fluctuations [1]. There will be an extraordinary possibility that Bob will not obtain what Alice sent if there is the extent of the communication between Alice and Bob, as an error changes the entire communication meaning. Codes of algebraic try to find cure for this problem by encoding redundant statistics into the transmission, giving a means of ensuring correctness.

The simplest kind of redundant data is repetition. Alice and Bob agree beforehand that each 16 bit string sent over the line of transmission will be the concatenation of two copies of the 8 bit message that Alice desires to send to implement this scheme. According to the example above, Alice would transfer the string 0100000101000001. At that moment, Bob will obtain the string 0100100101000001, if the fifth bit is turn over during transmission again. Now, Bob is able to detect the transmission error and therefore request that the message be retransmitted since he knows that the first 8 bits and the second 8 bits are supposed to be identical. Thus, Bob can identify any single-bit error of transmission.

Multiple bit errors may slip by its notice due to the agreement simplicity for redundancy. Take for example, if synchronized errors happened at the 5th and 13th bits, Bob will get the string 0100100101001001. Now, he cannot tell if the character `I' is transmitted by Alice or an `A' is transmitted after two well-placed errors happened in transmission. Besides, this agreement is only can detect errors - if Bob detected an error but desired to know what Alice really wanted to say, he would need to request an additional 16 bits transmission from her, significantly growing the time needed especially given the real-world transmission lines latency. An agreement between Alice and Bob with which Bob could not only identify but also correct transmission errors would be useful while retransmission is impractical or slow. We could repeat the pattern 3 times to do this. In this situation, Bob can correct a 1-bit error by taking the two copies which agree. A 2-bit error can be detected. Additional capability of detection or correction can be added by just increasing the number of repeats. Nevertheless, this is an extremely inadequately use of transmission resources. There are methods to construct more effective agreements.

The correcting or detecting codes of error aims to decrease the opportunity of receiving messages which different from the original message. The redundancy is the main concept behind error control coding. That is, adding further symbols to the original message that do not add info yet serve as control or check symbols. Correcting or detecting codes of error insert redundancy into the message, at the end of transmitter, in a systematic, analytic manner in order to allow the

original message reconstruction, at the end of receiver, if it has been distorted during transmission.

The vital objective is to make sure the message and its redundancy are connected by some algebraic equations set. It is reproduced at the receiver terminal by the use of these equations if the message is disturbed during transmission. Obviously, efficiency of error control is highly related with applying theory of mathematical in the error control schemes design. This study is focused to show the underlying mathematical structure of error correcting or detecting codes.

Algebraic Structure of Error Detection

We turn to structures of algebraic. The discrete digital communication nature certainly offers itself to manipulation through the finite fields theory, the simplest example being the correspondence of 1s and 0s in binary to the F_2 elements. Additional, we can view binary strings of length n as elements of the vector spaceF_2^n. Therefore, the characters discussed above would have been considered elements of F_2^8. We can then create redundancy methods using the linear algebra language by given these connections to algebra. While we began our discussion with binary data, the methods we shall use are equally applicable over all finite fields, so the rest of the paper will consider F_q instead of F_2.
An important thought used in the study of codes of error correction or detection is the group structure, which underlies other structures of algebraic such as rings and fields. Two elements of a set are operated by a binary operation at a time, yielding a third (not necessarily distinct) element. When a binary operation, along with certain rules restricting the results of the operation is imposed on a set, the resulting structure is a group.

The Implementation of Error Detection/ Correction Codes

Codes of error correcting or detecting are implemented in nearly all electronic device which involves information transmission, whether this info is transferred through a channel of communication or kept and retrieved from a storage system like a compact disk. The symbols set– this set always being finite – used to form the information message, set up the code alphabet. A channel, be it physical or not, is necessary in order to send information. A random channel of symmetric is considered in most of the cases presented. A channel is a random channel of symmetric error for each pair of distinct alphabet symbols a,b, there is a fixed probability $p_{a,b}$ that when a is transmitted, b is received and $p_{a,b}$ is the same for all possible pairs a, b $(a ? b)$. The basic operating scheme of error correcting or detecting codes is shown in Figure

1. Suppose that an information sequence of k message symbols or information digits is to be transmitted.

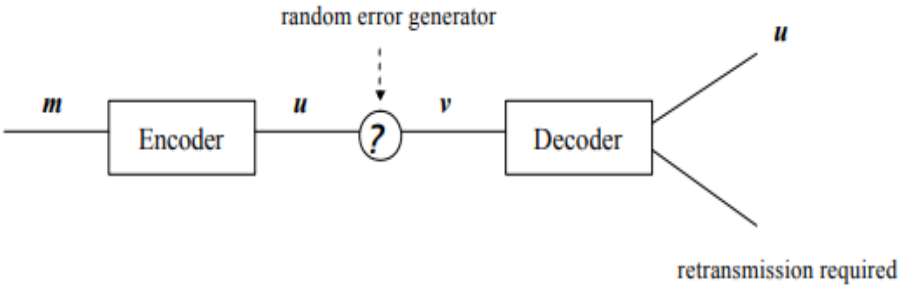

Figure 1: The basic operating scheme of error detecting / correcting codes

This sequence m may be mentioned to as message word. The encoder, at the end of transmitter, adds r check digits from the alphabet according to a assured rule, referred to as encoding rule. The encoder outputs a total sequence u of n digits, known as codeword, which is the real sequence transmitted. The $n - k = r$ additional digits, called parity-check digits, are the redundant digits used at the end of the receiver for errors correction and detection. Errors which happens during transmission alter codeword u to a word v, which is the received word. Whether the received word v satisfies the encoding rule or not is checked by the decoder. The error processing is performed, in an attempt to reproduce the actual transmitted codeword u if the condition is determined to be false. If this attempt be unsuccessful, the received word is ignored and retransmission is needed, else the decoder extracts the original message m from the reconstructed codeword u.

Matrices

To employ matrices in process of the encoding and decoding is common practice in error control systems.

A kxn matrix G over a Galois field $GF(q)$ is a rectangular array with k rows and n columns where each entry g_{ij} is an element of $GF(q)$ $(i = 0..k - 1$ and $j = 0..n - 1)$.

If $k \leq n$ and the k rows of a matrix G are linearly independent, the q^k linear combinations of these rows form a k-dimensional subspace of the vector space V_n of all n-tuples over $GF(q)$. Such a subspace is known as the row space of matrix G.

Moreover, through the dual space notion, previously presented, the existence of an $(n-k)xn$ matrix H for each kxn matrix G with k linearly independent rows, the matrix description of a code are suggested by a significant theorem, has as follows:

Theorem 1 [2]: For any kxn matrix G over $GF(q)$ with k linearly independent rows, there exists an $(n-k)xn$ matrix H over $GF(q)$ with $(n-k)$ linearly independent rows such that for any row g_i in G and any h_j in H, $g_i \cdot h_j = 0$. The row space of G is the null (dual) space of H, and vice versa.

Linear Block Codes

Most known codes are codes of block (or codes of fixed length). The data stream is divided by these codes into blocks of fixed length which are then preserved independently. There also exist codes of non-constant length like the codes of convolutional which a substantially different approach to error control is proposed. Redundancy can be introduced into an information sequence in these codes through the use of linear shift registers which entire data stream is renovated regardless of its length, into a single codeword. In common, decoding and encoding of convolutional codes depends more on planning the suitable shift register circuits and less on structures of mathematical. Thus, in this study, attention is confined to block codes, which invoke techniques of algebraic mostly according to the groups theory to insert redundancy into the sequence of information[3].

Block Coding

The sequence of information is segmented into message blocks of fixed length k in block coding. These blocks of message are encoded independently at the end of transmitter, decoded in the same manner at the end of receiver and then combined to retrieve the original message.

Using a of q symbols alphabet, where the collection of these q symbols is considered a Galois field of order q, $GF(q)$, there are q^k distinct blocks of message.

The encoder, based on the certain rules, transforms each message block m of length k into an n-tuple u which is the codeword to be transmitted, as shown in Figure 2.

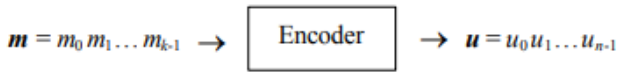

Figure 2: The transformation of the encoder

The length n of the codeword is bigger than k and these $(n - k)$ additional digits, frequently referred to as parity-check digits, are the redundancy added.

In order to make sure that the encoding process can be reversed in the end of receiver in order to retrieve the original message, there must be a one-to-one correspondence between a block of message and its corresponding codeword. This implies that there are exactly q^k codewords. The set of q^k codewords of length n is known as an (n,k) block code.

Vector Representation

In the block codes study, it is beneficial to associate codewords with vectors. Each codeword of length n, can be symbolized by a vector of dimension n

$u = u_0u_1...u_{n-1} \leftrightarrow u = (u_0, u_1, ..., u_{n-1})$

A codeword u, which is an n-tuple is symbolized by vector u whose n coordinates are the codeword components[4].

Definition of Linear Block Codes

Genrally, decoding and encoding of q^k codewords of length n may become prohibitively complex processes for large n and k. All codewords should be kept and searched through for every received word. The structure of linear algebraic of a codes major class, called codes of linear block, can be exploited in designing their scheme of decoding and encoding. The inherent linearity property of these codes means that they have structure of mathematical causing in a reduction in the complexity of their analysis and implementation.

These codes can be defined over a general finite field alphabet as follows according to the definition of linear block codes over $GF(2)$ in [2],

Definition 1 [2]: A block code of length n and q^k codewords is known as a linear (n,k) code C, if and only if its q^k codewords form a k-dimensional subspace of the vector space of all n-tuples over the field $GF(q)$.

Matrix Description

The each codeword vector representation combined with Definition 1 allows for the an *(n,k)* code matrix description. The processes of decoding and encoding can be reduced to matrix multiplication.

Generator Matrix

By Definition 1, an *(n,k)* code is a vector space *k*-dimensional subspace of all *n*-tuples over *GF(q)*. Therefore, it has potential to find *k* linearly independent codewords $g_0, g_1, \ldots, g_{k-1}$ in *C*. These codewords set forms a basis for code *C* since it consists of k linearly independent elements of *C*.

$$c = m_0 g_0 + m_1 g_1 + \ldots + m_{k-1} g_{k-1}$$

where $\{m_i\}\ i = 0..k-1$ are in *GF(q)*. The above expression is valid for any codeword *c* in *C*, implying that there is a one-to-one correspondence among the set of message blocks of the form $(m_0, m_1, \ldots, m_{k-1})$ and the codewords in *C*. Thus, all codewords [5] in *C* can be formed by the linear combination of *k* linearly independent codewords in *C*. Consequently, for any *(n,k)* linear code there exists a *kxn* matrix *G*, whose rows are these *k* linearly independent codewords

$$G = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

Obviously, the *G* row entirely specify the code *C*. The matrix *G* is known as a generator matrix for code *C* and is used for encoding any message $m = (m_0, m_1, \ldots, m_{k-1})$ as

$$\boldsymbol{c} = \boldsymbol{m} \cdot G = [m_0\, m_1 \ldots m_{k-1}] \cdot \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} = m_0\mathbf{g}_0 + m_1\mathbf{g}_1 + \ldots + m_{k-1}\mathbf{g}_{k-1}$$

The generator matrix is central to the linear block codes description since the encoding process is reduced to matrix multiplication. In addition, only the k rows of *G* need to be stored; the encoder merely needs to form a linear combination of the *k* rows according to the input message $m = (m_0, m_1, \ldots, m_{k-1})$. Similarly, the decoding process can be simplified by the use of another matrix, the parity-check matrix, for any *k*x*n* matrix *G* with *q* linearly independent rows there exists an $(n - k)$x*n* matrix *H* with $(n - k)$ linearly independent rows such that any vector orthogonal to the *H* rows is in the *G* row space and thus a valid codeword.

$$H = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{h}_{n-k-1} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdot & \cdot & \cdot & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdot & \cdot & \cdot & h_{1,n-1} \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdot & \cdot & \cdot & h_{n-k-1,n-1} \end{bmatrix}$$

Such a matrix H is known as a parity-check matrix of the code and is used for decoding since a received word *v* is a codeword if and only if $v \cdot H\,T = 0$**,** where H^T is the transpose1 of *H*. Besides, the $(n - k)$ linearly independent *H* rows span an $(n - k)$-dimensional subspace of the vector space of all *n*-tuples over *GF(q)*. It can be seen that this is the dual space of the vector space formed by the *(n,k)* code. Thus, *H* can be regarded as a generator matrix for an $(n,n - k)$ code. This code, with regard to the notion of dual space, is called the dual code *C*-of *C*.

Conclusion

As observed through this study of error correcting or detecting codes, the process of encoding is likely to be less extensive than decoding. In fact, a single task is performed by the encoder; that is, transforming each message word to a codeword by adding redundancy. In contrast, three tasks are conducted by the decoder; errors detection, error processing in case of occurrence of errors, extraction of the original message word from the codeword. These processes reveal a substantial decoding complexity and imply extensive digital equipment requirements. Given that decoding is primarily based on the rule of encoding, the codes of error correcting or detecting codes design should emphasis on optimizing the process of encoding in such a manner that it simplifies decoding.

References

[1]Cox, David A., and John B. Little. Using algebraic geometry. 2. ed. New York, NY: Springer, 2005.
[2] Lin, S., Costello, J. C. (1983) Error Control Coding: Fundamentals and Applications, Prentice-Hall, ISBN 0-13-283796-X
[3] Nathan Kaplan and members of the tutorial. Coding theory lecture notes. Harvard, Summer 2011. http://users.math.yale.edu/~nk354/teaching.html.
[4] Herstein I (1975) Topics in algebra, 2nd edn. Wiley, Hoboken
[5] Talbot J, Welsh D (2006) Complexity and cryptography: an introduction. Cambridge University Press, Cambridge