The Search for New Quinary Quasi-Cyclic Codes

Ryan Ackerman '10 and Nuh Aydin

Kenyon College Department of Mathematics

Abstract

Coding and information theory focuses on new ways to store information efficiently, but at the same time guaranteeing the integrity of that information. For example, by using codes we are able to read a CD despite small scratches on the surface. Each linear code can be described by three numbers, two of which describe the efficiency of the code and the third describing the code's ability to detect and correct errors which occur during transmission. For each efficiency, there exists a theoretical upper bound for the code's error correction capability. Quasi-Cyclic codes are a class of linear codes which provide many good codes, some of which have a greater error correcting capability than any other code found previously. This project uses this observation to try to find new record-breaking codes by generating many Quasi-Cyclic codes of a fixed efficiency, and then finding the ability of each code to detect errors and comparing it to previously found linear codes.

Introduction to Codes

Coding theory is a relatively new field of mathematics, coming largely with the onset of computers and the need to transmit large amounts of data accurately and efficiently over channels which may change the message. Sending emails, listening to the radio, and even reading a CD all involve codes. Each code will expand the message, turning the raw bits into a string of numbers which is organized like a language and allows the receiver to determine what has changed, if anything, in the message during transmission. For example, if I were to say "I took my dog to the parl," you would know that there was some sort of mistake in the transmission of the message because "parl" is not a word in the English language. You might even recognize some words which are close to "parl," which you could assume I was saying (if you ignore context) such as pail, park, or Karl. When mathematically constructed, we can guarantee a distance between the words in our language, that is, construct the code so that the number of changes to get from one word to the next is high.

Linear Codes

Linear Codes are an important class of codes. Codes are made up of words, each word being a string of numbers. Linear Codes provide an efficient mathematical method to encode and decode a message. Consider the matrix below, which generates the [7,4,3] Hamming Code. If we add any number of the rows (also called a linear combination) in this generator matrix, we will get a vector (or word) which is in our code. Notice that there are 7 digits in each word; this is the length of our code and it represents how long each encoded word is. There are also 4 rows, which is called the dimension of the code. This is going to be the original length of each word in our message. The last number, 3, tells us that there needs to be three changes to a single word get from one word to another. For example, to get from the third row to the fourth we change the 3rd digit to a 0, the 4th digit to a 1, and the 5th digit to a 1. To encode a message using a Linear Code, all we need to do after changing our message to a string of 0's and 1's is to break up the message into words, each of length 4, and then multiply our words by the generator matrix. This results in a linear combination of the rows, so the resulting string of 7 digits is a word in our code. To decode a message, if the received word is in our language then we multiply the received word by the generator matrix to get our original string of 4 digits. If the received words is not in the language, then we find the word which is in our language and requires the least number of changes to get our received word. We then assume that this is the word that was sent and decode from there.

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

Figure 1. The generator matrix of the [7,4,3] Binary Hamming Code.

Figure 2. Image taken by the Voyager 1 spacecraft on its flyby of Jupiter. This image was sent back to Earth using a Reed-Solomon code, which is a type of Linear Code. The Reed-Solomon code was chosen because it has a large error correcting capability and because the Reed-Solomon code tolerates a large number of errors in a small part message (solar flares can produce such an effect). The code accomplishes this by scrambling the encoded words so that if a large number of digits in a row are all changed, the errors are spread out amongst a large number of words. The same method is used on CDs and DVDs to make them resistant to small scratches.

Image from http://en.wikipedia.org/wiki/Voyager_1

Cyclic Codes

Like Reed-Solomon codes, Cyclic Codes are a type of Linear Code. They are described in much the same way, using three numbers to indicate the length, dimension, and distance of the code. However, these codes can be generated from one vector, or word. This is accomplished by taking the last digit in the generator vector and sending it to the front. It's easy to see that eventually we will arrive back at our generator vector if we take enough cyclic shifts, however it is only necessary to take k cyclic shifts, where k is the dimension of the cyclic code. We can comprehensively find all of the Cyclic Codes for a given length and dimension using polynomials. If we represent a vector $[v_0,v_1,v_2,\dots,v_m]$ by the polynomial $p(x) = x^0v_0 + x^1v_1 + x^2v_2 + \dots + x^mv_m$, then we can say that all the generators of Cyclic Codes of length n will be products of the prime factors of the polynomial x^n-1. It is then possible to pick the best Cyclic Codes for each length and dimension and create a list, from which we can generate Quasi-Cyclic Codes.

Constacyclic Codes

Constacyclic Codes are closely related to Cyclic Codes, but are constructed using constacyclic shifts rather than cyclic shifts. A constacyclic shift works much the same as a cyclic shift; the last bit goes to the front, multiplied by some constant a, and the rest of the word is shifted normally. To find all the Constacyclic codes, we factor the polynomial x^n-a and proceed as with Cyclic Codes.

Quasi-Cyclic Codes

Cyclic Codes are a special case of Quasi-Cyclic Codes, which in turn are Linear Codes. Quasi-Cyclic Codes are generated by a vector, just like Cyclic Codes. However, rather than taking all cyclic shifts of the generator vector, Quasi-Cyclic Codes will take only the l^{th} cyclic shift, for some fixed l which divides the length of the code, n. Because when we calculate the distance between (the number of changes to get from one word to the next) two words, we look at each word column by column, we can permute the columns of our matrix and retain a code which has similar properties to the original code, which is called an equivalent code. Because of this, we can generate Quasi-Cyclic Codes using Cyclic Codes; if we have an l-Quasi-Cyclic Code, then through column permutations we can find that the generator matrix can be split up into l blocks of Cyclic Codes. This property allows us to easily search through many Quasi-Cyclic Codes in order to find ones which previously have not been found. Quasi-Twisted Codes are a generalization of Quasi-Cyclic Codes, and are exactly the same except Quasi-Twisted Codes are constructed using constacyclic shifts.

Search Algorithm

It is well known in the scholarly community that Quasi-Cyclic and Quasi-Twisted Codes have a high density of good codes, which makes them a good place to look for new codes. The website www.codetables.de contains a list of all the best known Linear Codes, in addition to storing the theoretical upper bound of distance for a given length and dimension. After finding all the best known Cyclic Codes using the method described in that section, we were able to generate good Quasi-Cyclic Codes by concatenating a Cyclic Code with itself l times, resulting in a l-Quasi-Cyclic Code. Each time we added a cyclic block we multiplied the generator polynomial by a random polynomial which was relatively prime (that is, they share no common factors) to the inverse of the generator. This results in a polynomial which generates the same Cyclic Code, but in a different order. Therefore, the lower bound for the distance of our Quasi-Cyclic Code was ld, where d is the distance of the Cyclic Code. Often times we were able to achieve distances much higher than this.

Using a computer, we were able to run hundreds of thousands of trials for each Cyclic Code to see if it generated a record breaking Quasi-Cyclic Code. We searched through codes in F_5, which means that rather than using the binary alphabet to make our words, the alphabet was {0,1,2,3,4}.

Results

We were able to find 12 record breaking Quasi-Twisted Codes in F_5.

- [56,6,40] a=2
- [56,14,28] a=4
- [56,16,26] a=4
- [68,16,34] a=2
- [68,18,32] a=2
- [84,18,42] a=2
- [84,20,40] a=2
- [92,22,43] a=2
- [102,16,57] a=2
- [102,18,54] a=2
- [104,20,54] a=2
- [116,14,70] a=2

And one Constacylic Code:

- [58,14,30] a=2

$$\begin{matrix} 3 & 2 & 1 & 0 & 0 & 1 \\ 1 & 3 & 2 & 1 & 0 & 0 \\ 0 & 1 & 3 & 2 & 1 & 0 \end{matrix}$$

Figure 3. The first and second cyclic shift of the vector (3,2,1,0,0,1). The second cyclic shift can also be referred to as the first 2-Quasi-Cyclic shift.

Acknowledgements

I would like to thank Prof. Nuh Aydin for his guidance and advice throughout the project. I would also like to extend our thanks to Dr. Markus Grassl* for his help in identifying and working around a bug in our computer software package, MAGMA, and Geoff Bailey** for his help in fixing the aforementioned bug.

*Senior Research Fellow, National University of Singapore

**Research Associate, University of Sydney

References

1. Aydin, Nuh and Asamov, Tsvetan. "Search for Good Linear Codes in the Class of Quasi-Cyclic and Related Codes." Selected Topics in Information and Coding Theory. Ed. Woungang, I., Misra, S., and Chandra Misra, S. World Scientific. 2009.
2. Hankerson, D R, et al. Coding Theory and Cryptography: The Essentials. New York: Marcel Dekker, 2000.