ABSTRACT ALGEBRA FINAL PROJECT: GROUP CODES AND COSET DECODING

RENÉE DOSICK

1. Preface: Stumbling Blocks and the Learning Process

Initially, this topic was a bit intimidating to me. It seemed highly technical at first, especially to someone inexperienced with binary. However, upon attempting to piece together information from a couple of different books, the real problem turned out to be getting varying (and sometimes fraught with typos) notation to agree, and understanding the reasoning behind each step in the decoding process, which the authors of these texts tended to gloss over. Though it took me a while to wrap my mind around, the most important realization was that the things I was dealing with, these collections of strings of 0's and 1's of a certain length, were *groups*, and that the codes I was studying were simply subgroups. This made them seem much more manageable, and instead of seeing the process of coding and decoding as something utterly unfamiliar, I was then able to apply my knowledge of concepts like cosets, order, and Lagrange's Theorem to a new environment, filling in the gaps in my sources until the basic process of coding theory seemed logical and believable.

2. Introduction

Coding theory is an important concept in the modern world, since it is used in the transmission of information by radio, telephone, compact discs, and many other forms of technology. The term coding theory does not refer to secret codes or cryptography; it is merely a method of simplifying information (words, for example) and putting it in a form that can be easily transmitted and then accurately reconstituted by a receiver of some sort (generally as strings of digits). For the sake of simplicity, in this paper I will only discuss binary codes, those that represent

Date: May 12, 2010.

information by codewords composed of 1's and 0's.
The best coding methods are those that are reliable and efficient, that is, they are able to detect and/or correct as many interference-induced errors as possible, and they contain as little excess or redundant information as possible.

2.1. **An Inefficient Code.**

An example of a code that is less than ideal is a repetition code; such a code increases the probability of a word being correctly decoded by sending each code word many times. Thus, using maximum-likelihood decoding, i.e., interpreting a received word as the codeword sharing the largest number of digits with it, the chance of a correct reception increases with every time a message is repeated (assuming, of course, that the probability of an error-free transmission is greater than .5). While this method can make for fairly reliable code, it is extremely inefficient: for a codeword of length n, $\frac{n-1}{n}$ of the information sent is redundant!

Using ideas from group theory, we can find much better codes.

3. Linear Codes

A group code is a code (a set of codewords) that is a subgroup of $\mathbf{Z}_2^n$, i.e., the sum of any two elements in the code is also in the code (we already know that the inverse of each codeword is in the code since in binary, everything is its own inverse—for example, $(01001) + (01001) = (00000)$—and it is also unecessary to check that 0 is a codeword for the same reason).
An (n, k) linear code over $\mathbf{Z}_2$ is a k-dimensional group code that is the null space of some matrix $H \in \mathbf{M}_{(n-k)\times n}(\mathbf{Z}_2)$, called a parity-check matrix. It is easy to see that such a code must be a group code since we know that kernels are subgroups.

3.1. **Example of a Linear Code.**

Let C be the (6,3) linear code whose parity-check matrix is

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

Then C consists of codewords in the null space of H:

(000000) (001101) (111000) (110101) (101110) (100011) (010110) (011011)

4. Coset Decoding: Standard Arrays

When we have a binary linear code, we can use the fact that it is a subgroup of $\mathbf{Z}_2^n$ to decode it. To do this, we construct a standard array, a table made up of a particular code's codewords and all cosets of the code in $\mathbf{Z}_2^n$.
The first row of the table contains the codewords. There are 2^k codewords, since the rank of an $(n-k) \times n$ parity-check matrix is $n-k$, so its nullity is k, and there are 2^k unique combinations of these k codewords, since each codeword is its own inverse. In our example code C, for instance, pick any three linearly independent codewords, say a, b, and c; the set of all $2^3 = 8$ posible combinations of these codewords is $a, b, c, ab, ac, bc, abc, aa = bb = cc = 0$.
The remaining rows of the standard array consists of cosets of C. Lagrange's Theorem tells us the number of distinct cosets: since the order of $\mathbf{Z}_2^n$ is 2^n and the order of C is 2^k, the number of distinct left or right cosets of C in $\mathbf{Z}_2^n$ (the two are equivalent) is 2^{n-k}. Thus, in our example code, there are $2^{6-3} = 8$ rows as well.
The first entry of each row is a representative of the least possible Hamming weight for that row's coset, i.e., an item with the least number of nonzero components. These entries are referred to as coset leaders. The remaining entries in a row are made by adding the coset leader of that row to the codeword heading the column to which the entry belongs.

4.1. **Example of a Standard Array.** The standard array for our example code C is:

000000	001101	111000	110101	101110	100011	010110	011011
100000	101101	011000	010101	001110	000011	110110	111011
010000	011101	100101	100101	111110	110011	000110	001011
011000	000101	110000	111101	100110	101011	011110	010011
000100	001001	111100	110001	101010	100111	010010	011111
000010	001111	111010	110111	101100	100001	010100	011001
000001	001100	111001	110100	101111	100010	010111	011010
100100	101001	011100	010001	001010	000111	110010	111111

4.2. **Coset Decoding: Using a Standard Array.** Once we have constructed a standard array for our code, the process of decoding is quite simple. For any received word, the actual codeword that was transmitted is given by the codeword heading the column to which the received word belongs (assuming the least possible amount of interference has occured).
The reason this is possible is that, in constructing the standard array the way we have, it is easy to see that any received word has most likely been formed by adding something of minimum weight (the coset leader heading the row in question) to a codeword, thus changing it minimally, and working backwards will give the original message.

4.3. **Example of Decoding with a Standard Array.** Suppose that some codeword from our example code C has been transmitted, and we receive the word 011100.
Since the codeword heading the column to which this received word belongs is 111000, we can assume that this is the message we were intended to receive, and the error that occured during transmission was the addition of the coset leader 100100.

5. Decoding with Syndromes

Using a code's parity-check matrix H, there is an even simpler way to implement coset decoding! The syndrome of any word $x \in \mathbf{Z}_2^n$ is the product Hx. If we write x as $x = c + e$, where $c \in C$ is a transmitted codeword and e is the transmission error, we can see that the syndrome of x is equal to the syndrome of the error e.
To see why this is true, we use the fact that $Hc = 0$ by definition, since c is in the null space of H, and get:

$$Hx = H(c + e) = Hc + He = 0 + He = He$$

Thus, all elements in the same coset of a code or row of its standard array have the same syndrome: any two vectors x and y are in the same coset if and only if

$x - y \in C$ (by a basic property of cosets), and if $x - y \in C$, then $x - y = 0$, so $H(x - y) = 0$ and $Hx = Hy$.

Syndrome decoding can also be explained in terms of the First Isomorphism Theorem. Since our code C is the kernel of the linear map $H : \mathbf{Z}_2^n \longrightarrow \mathbf{Z}_2^{n-k}$, there is an induced isomorphism $\tilde{H} : \mathbf{Z}_2^n/C \overset{\sim}{\longrightarrow} \mathbf{Z}_2^{n-k}$. When we decode using syndromes, we are really passing a word through the mapping H (this looks like matrix multiplication), then mapping the resulting syndrome through the inverse of $\tilde{H}$ and subtracting the coset leader to get the original codeword; it works because we have an isomorphism!

We can now construct a decoding table, which lists each coset leader and its syndrome. For any received word, we can decode it simply by multiplying with H, matching the result to a coset leader with the same syndrome, and subtracting (equivalent to adding) this coset leader to get the original codeword.

5.1. **Example of Syndrome Decoding.** A decoding table for our (6,3) code looks like:

Syndrome	Coset Leader
000	000000
001	100000
101	010000
011	001000
100	000100
110	000010
111	000001
101	100100

Suppose the word $x = 111010$ is received. Its syndrome is $H \cdot x = 110$, which is also the syndrome for the coset leader 000010. Subtracting 000010 from 111010, we get 111000, which we then can take to be the transmitted codeword. Note that this is also the codeword heading the column to which 111010 belongs, but much less work was required to decode it than when making a standard array.

5.2. **Benefits of Syndrome Decoding.** Although in our example, the standard array was fairly simple to construct, it clearly would be impractical in cases where there are very large numbers of codewords. Since an (n, k) linear code will have 2^n possible received words to be put in a standard array, but only 2^{n-k} cosets, it is much simpler to merely calculate the syndrome of each coset leader, with the same result.

References

[1] Thomas W. Judson, *Abstract Algebra: Theory and Applications* (PWS Publishing Company, Boston, 1994).

[2] Joseph A. Gallian, *Contemporary Abstract Algebra*, (Houghton Mifflin Company, Boston, 2006).