AN INTRODUCTION TO ALGEBRAIC CODING THEORY

JONATHAN SHAW

ABSTRACT. In this project, we will present a survey of the field of algebraic coding theory. We will begin by introducing some definitions and fundamental principles. We will then look at how the algebraic group structure is desirable in coding. Finally, we will examine an encoding scheme that produces a group code, as well as a decoding scheme that takes advantage of group properties.

1. WHAT IS CODING THEORY?

Given the name of the discipline, one would be forgiven for thinking that coding theory is related to secret codes. Indeed, the objective of coding theory is not to obscure communications—rather, it is to ensure that messages arrive successfully and are interpreted by the recipient as intended. The following (rather contrived) scenario illustrates a major problem that may occur in communication over a noisy channel:

Suppose that Bob, an airline pilot, and Alice, an air traffic controller, have agreed upon the following communication scheme:

- When Bob receives the binary string "000" from Alice (via satellite), he knows it is safe to land his plane.
- When Bob receives the string "001", he knows that the landing conditions are unsafe; he should remain airborne.

Now, suppose that a surprise solar storm were to momentarily interfere with the satellite, resulting in the last bit in Alice's "001" being flipped. Bob now receives "000" and believes it is safe to land, when in fact it is not.

Can we find a way to structure our messages so that we can reliably detect when a transmission error has occurred? Moreover, can we find a way to correct those messages we feel have been transmitted incorrectly? These are the fundamental problems we address in coding theory [2, Chapter 1]. In this paper, we shall establish a vocabulary with which we can discuss these questions, and then explore the ways in which the algebraic group structure can help us achieve our goals.

2. FUNDAMENTAL CONCEPTS OF CODING

Before we formalize coding theory, it is instructive to give an example of one very simple coding scheme that may have helped Alice and Bob in the introductory scenario.

Date: December 6, 2019.

Key words and phrases. coding, encoding, decoding, coding theory, group codes, groups, cosets.

Example 1. Instead of sending their messages (000 and 001) directly, Alice and Bob could agree to append an extra bit to their messages, where the extra bit is the sum (modulo 2) of all the 1's that appear in the original message. Now, Alice and Bob have two *code words*—0000 and 0011—that they can send. These code words have an interesting property: the number of 1's that appear is always even. If any single bit flips in one of the messages, the number of 1's will be odd, which will alert the recipient that a single-bit transmission error has been made. This scheme is called the *(4,3) parity-check code* [5, Example 16.20]. Note that this scheme cannot detect an error in which an even number of bits are flipped. Moreover, in general there is no way to tell which bit has flipped: therefore, this scheme provides no error-correcting capabilities [5, Section 16.5].

1. **Basic Notation.** We are concerned with **messages**—the original content we wish to send—as well as their corresponding **code words**, which are transmitted. In our treatment of coding theory, an m-bit message is considered an element of the external direct product $\bigoplus_{i=1}^{m} \mathbb{Z}_2$, which we shall write as ${\mathbb{Z}_2}^m$. Similarly, an n-bit code word is an element of ${\mathbb{Z}_2}^n$. For brevity, we will write these elements as binary strings: for example, $(1, 0, 0, 1) \in {\mathbb{Z}_2}^4$ will be written simply as 1001.

- $W \subseteq {\mathbb{Z}_2}^m$ denotes the set of all m-bit messages we might wish to transmit.
- $C \subset {\mathbb{Z}_2}^n$ denotes the set of all n-bit code words corresponding to the messages in W. We will refer to C simply as a **code**. For our purposes, $n > m$—that is, code words are longer than the messages they encode.
- $E : W \to C$ is an injective function that maps messages to their corresponding code words. It is called the **encoding function**.
- When a sender sends a given code word c, $T(c)$ denotes the string in ${\mathbb{Z}_2}^n$ received by the recipient. T is not a function because it is non-deterministic.
- $D : {\mathbb{Z}_2}^n \to {\mathbb{Z}_2}^m$ denotes the **decoding function**. If we wish our decoding function to be error-correcting, then its codomain should be W.

2. **Hamming Distances and Weights.** In Bob and Alice's scheme, $C = \{000, 001\}$. This presents a glaring problem: the two messages in this message space differ by only one bit—they are very "close" to one another. American mathematician Richard Hamming (1915-1998) introduced the following terminology to formalize the notion of "closeness" in codes [4, Chapter 31]:

Definition 1. Let $a = a_1a_2 \cdots a_n$ and $b = b_1b_2 \cdots b_n$ be elements of ${\mathbb{Z}_2}^n$ for some $n \in \mathbb{N}$.

- The **weight** of a, denoted $\text{wt}(a)$, is the number of components a_i of a for which $a_i = 1$ [3, Definition 1.9].
- The **distance** between a and b, denoted $d(a, b)$, is the number of components a_i for which $a_i \neq b_i$ [3, Definition 1.4].

Lemma 2. *For every* $x, y \in {\mathbb{Z}_2}^n$, *wt*$(x + y) \leq$ *wt*$(x) +$ *wt*(y) [5, Lemma 16.2].

Proof. Suppose $\text{wt}(x + y) > \text{wt}(x) + \text{wt}(y)$. Then there exists at least one i, $1 \leq i \leq n$, such that $x_i + y_i$ = 1 while $x_i = 0$ and $y_i = 0$. This is a contradiction, since $x_i + y_i = 1$ necessarily implies that exactly one of x_i or y_i is equal to 1. □

Lemma 3. *For every* $x, y \in {\mathbb{Z}_2}^n$, *wt*$(x + y) = d(x, y)$ [5, Example 16.22].

Proof. For $1 \leq i \leq n$, (x_i, y_i) contributes 1 to the count of $d(x, y)$ if and only if $x_i \neq y_i$. Also, $x_i + y_i$ contributes 1 to the count of $\text{wt}(x + y)$ if and only if $x_i \neq y_i$. It follows that $\text{wt}(x + y) = d(x, y)$. □

Definition 4. Let $n, k \in \mathbb{N}$ and $x \in {\mathbb{Z}_2}^n$. The **sphere** of radius k centred at x, denoted $S(x, k)$, is defined [5, Definition 16.10] as

$$S(x, k) = \{y \in {\mathbb{Z}_2}^n : d(x, y) \leq k\}.$$

Armed with these definitions, we can now discover the properties that make a given encoding function desirable.

3. **Error Detection and Error Correction.** When choosing a coding scheme to help us transmit information over a noisy channel, our goal is typically one of the following [2, Chapter 1]:

(1) **Error Detection**: Detecting when a transmission error has occurred so that incorrect transmissions can be disregarded and perhaps a re-transmission can be requested
(2) **Error Correction**: Attempting to map an incorrectly-transmitted code word to the intended message in the presence of transmission errors

If our goal is to *detect* transmission errors, then we would hope that code words transmitted within a certain margin of error would fall outside of C. The following result helps us design codes that satisfy this requirement.

Proposition 5. *Given a code $C \subset {\mathbb{Z}_2}^n$ and a positive integer k, we can detect all transmission errors of weight $\leq k$ if the minimum distance between code words is at least* $k + 1$ [4, Theorem 31.2] [5, Theorem 16.12].

Proof. Let $k + 1$ be the minimum distance between code words in C. Suppose a code word a is transmitted incorrectly and is received as $T(a)$, where $T(a) \neq a$.
If $T(a) \in S(a, k)$—that is there are at most k errors in—then $d(a, T(a)) \leq k$, so $T(a) \notin C$ by the minimum distance property of C we defined. □

Example 2. Suppose that Alice and Bob adopt an encoding scheme for their messages $W = \{000, 001\}$ where $C = \{000000 = E(000), 001011 = E(001)\}$. Here, the minimum distance between code words is 3, so Proposition 5 tells us that if any two bits are flipped for either of our code words, the resulting value falls outside of C. For example, if $c = 000000$ is incorrectly transmitted as $T(c) = 000011$, we find that $T(c) \notin C$. Indeed, one would have to flip 3 bits for one codeword to be construed as the other.

In many applications, we do not have time to request re-transmission. In these cases, we need to *correct* errors on the fly. To do this, we need assurance that transmitted code words $T(c)$ within a certain sphere of their closest code word $c \in C$ will still map to the correct message w. The following result tells us how we can achieve this.

Proposition 6. *Given a code $C \subset {\mathbb{Z}_2}^n$ and a positive integer k, we can construct a decoding function $D : {\mathbb{Z}_2}^n \to W$ that corrects all transmission errors of weight $\leq k$ if the minimum distance between code words is at least* $2k + 1$ [4, Theorem 31.2] [5, Theorem 16.13].

Proof. Let $2k+1$ be the minimum distance between code words in C, and let $r \in {\mathbb{Z}_2}^n$ be received. Define the decoding function D as follows:

$$D(r) = \begin{cases} w \text{, where } E(w) = a & \text{if } r \in S(a,k) \text{ for some } a \in C \\ w_{err} \text{ (an arbitrary error message)} & \text{if } r \notin S(a,k) \text{ for any } a \in C \end{cases}$$

We claim that D is a function. If not, then there exists at least one $r \in {\mathbb{Z}_2}^n$ such that $r \in S(a_1, k)$ and $r \in S(a_2, k)$ for distinct codes $a_1, a_2 \in C$. Equivalently, $d(a_1, r) \leq k$ and $d(a_2, r) \leq k$. Applying Lemma 2, we see $d(a_1, a_2) \leq d(a_1, r) + d(a_2, r) \leq k + k < 2k + 1$—a contradiction. So D is, in fact, a function, and each transmission $r \in {\mathbb{Z}_2}^n$ will be mapped to exactly one code word. □

Example 3. Consider again the encoding scheme Alice and Bob developed in Example 2: $C = \{000000 = E(000), 001011 = E(001)\}$. Since the minimum distance between code words is 3, we know (by Proposition 6) that we can detect errors of weight 1. It is easy to see this:

- If there is an error of weight 1 in the transmission of $E(000)$, the received code word will have weight 1.
- If, on the other hand. there is an error of weight 1 in the transmission of $E(001)$, the received code word will have weight 2 or 4.

Hence, if we receive a code word with weight 0 or 1, we can immediately map it to the message 000. If we receive a code word with weight 2, 3, or 4, we can map it to the message 001. We only begin to run into problems if the weight of the transmission error ≥ 2, as then, for example, $E(000)$ could be transmitted as 001001 and decoded as 001.

It is important to note that the results of Propositions 5 and 6 do not imply that we can have the maximum level of error detection and error correction at the same time [4, Chapter 31]. Error-detection and error-correction strategies can sometimes conflict with each other. Recalling the previous two examples, suppose that $E(001) = c = 001011$ were incorrectly transmitted as $T(c) = 001000$. We would believe that the intended code word was 000000, and we would decode as 000—thus failing to detect an error of weight 2.

As we can see from propositions 5 and 6, the minimum distance between code words in C is of the utmost importance to us. The number of comparisons required to compute this number, however, is $\binom{|C|}{2}$—which can become very large for large values of $|C|$. This leads us to ask whether there may be a better way to calculate this minimum distance—a problem we will address in the next section.

3. Group Codes

Definition 7. Let $E : {\mathbb{Z}_2}^m \to C \subset {\mathbb{Z}_2}^n$ be an encoding function, where $m < n$. The code C is called a **group code** if C is a subgroup of ${\mathbb{Z}_2}^n$ [5, Definition 16.11].

The following fact gives us an easy way to check whether a given code is a group code.

Proposition 8. *Given a code C and its encoding function $E : {\mathbb{Z}_2}^m \to C \subseteq {\mathbb{Z}_2}^n$, C is a group code if its encoding function is operation-preserving.*

Proof. ${\mathbb{Z}_2}^m$ is a proper subgroup of itself, so it is nonempty. Therefore, $E({\mathbb{Z}_2}^m)$ is nonempty since E is injective. Let a and b be elements of ${\mathbb{Z}_2}^m$. Obviously, $a + b \in {\mathbb{Z}_2}^m$. Since E is operation-preserving, $E(a) + E(b) = E(a + b) \in C$, so C is a subgroup of ${\mathbb{Z}_2}^n$ by the finite subgroup test. □

Why do we care whether a given code has a group structure? As it turns out, the following result gives us an elegant and quick way to calculate the minimum distance between code words for such a code—a very useful property for large values of $|C|$.

Theorem 9. *If C is a group code, the minimum distance between any two code words in C is the minimum of the weights of the non-identity elements of the code* [5, Theorem 16.14].

Proof. Let $\vec{0}$ denote the identity element of ${\mathbb{Z}_2}^n$, and let $a, b, c \in C$, where c is the non-identity element in C with minimum weight and $d(a, b)$ is the minimum distance between elements in C. Consider the following two facts:

- $a + b \in C$ by closure. By Lemma 3, $\text{wt}(a + b) = d(a, b)$. We find that $\text{wt}(a + b) \geq \text{wt}(c)$ by our choice of c, so $d(a, b) \geq \text{wt}(c)$.
- By Lemma 3, $\text{wt}(c) = \text{wt}(\vec{0} + c) = d(\vec{0}, c)$. We find that $d(\vec{0}, c) \geq d(a, b)$ by our choice of a and b, so $d(a, b) \leq \text{wt}(c)$.

We conclude that $d(a, b) = \text{wt}(c)$. □

4. Generator Matrices and Coset Leaders

Now that we've laid the groundwork of the importance of the group structure in coding theory, we turn our attention to actual encoding and decoding schemes that are both practical and efficient. In particular, we will define an algorithm for generating codes with a group structure, as well as an algorithm that uses group properties for error-correction in decoding.

1. **Encoding via Generator Matrices.** We wish to obtain an encoding function that allows us to easily recover a message, while still providing redundancy. An excellent way to obtain such a function requires the following definition:

Definition 10. The $m \times n$ **standard generator matrix** G is given by

$$G = [I_m | A]$$

where A is an $m \times (n - m)$ matrix with linearly-independent columns [4, Chapter 31, Example 7].

We can define our encoding function $E : {\mathbb{Z}_2}^m \to {\mathbb{Z}_2}^n$ by $E(w) = wG$ for $w \in {\mathbb{Z}_2}^m$. This encoding function is useful because it preserves the the message w as the first m bits of $E(w)$, while also appending $n - m$ additional bits for redundancy [4, Chapter 31, Example 7].

Example 4. We might define the encoding function $E : {\mathbb{Z}_2}^3 \to {\mathbb{Z}_2}^6$ by $E(w) = wG$, where

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

We find, for example that $E(101) = 101110$ and $E(001) = 001011$. Indeed, for every w in ${\mathbb{Z}_2}^3$, the first three bits of $E(W)$ are equal to the three digits of w.

A major benefit of this encoding scheme is the following result:

Theorem 11. *Consider an encoding function $E : {\mathbb{Z}_2}^m \to C \subset {\mathbb{Z}_2}^n$ given by $E(w) = wG$, where G is a standard generator matrix. The code C resulting from E is a group code* [5, Theorem 16.15].

Proof. Recall from Proposition 8 that, to show that C is a group code, it suffices to show that E is operation-preserving. Let $x, y \in {\mathbb{Z}_2}^m$. $E(x + y) = (x + y)G = xG + yG = E(x) + E(y)$, so C is a group code. □

Encoding via a standard generator matrix, then, allows us to use Theorem 9 to determine the minimum weight between codewords. As such, we can determine the error-detection and error-correction capabilities of our code with relatively few computations.

2. **Decoding by Coset Leaders.** Having found a neat way of encoding messages, we turn our attention to decoding. In particular, we are interested in developing an error-correcting coding scheme that maps a received code word to its nearest actual code word. There are many ways in which this can be done. One common method is to decode using **partity-check matrices**, which are closely related to standard generator matrices. For a discussion on this technique, the interested reader may consult Joseph Gallian's introductory chapter on coding theory [4, Chapter 31]. For the purposes of this paper, however, we consider another decoding technique that relies on the group structure of the code C obtained via the use of standard generator matrices.

It is instructive to demonstrate this algorithm with a specific example. For similar examples, see [4, Chapter 31, Example 11] and [5, Example 16.26].

Example 5. Continuing from Example 4, we will construct a decoding table, or **standard array**, for the group code $C \subset {\mathbb{Z}_2}^6$.

(1) First, we will list each element of C, beginning with the identity.

000000	100101	010110	001011	110011	101110	011101	111000

(2) We select an element x from ${\mathbb{Z}_2}^6$ that does not appear in C and has minimum weight. Such an x is called a **coset leader**. For each $c \in C$, we write the element $x + c \in {\mathbb{Z}_2}^6$ underneath c. The resulting row is the coset $x + C$. Following this procedure with $x = 100000$ yields

000000	100101	010110	001011	110011	101110	011101	111000
100000	000101	110110	101011	010011	001110	111101	011000

(3) We repeat Step 2 until every element of ${\mathbb{Z}_2}^6$ appears in the table. Thus, the cosets invoke a partition on ${\mathbb{Z}_2}^6$.

000000	100101	010110	001011	110011	101110	011101	111000
100000	000101	110110	101011	010011	001110	111101	011000
010000	110101	000110	010011	100011	111110	001101	101000
001000	101101	011110	000011	111011	100110	010101	110000
000100	100001	010010	001111	110111	101010	011001	111100
000010	100111	010100	001001	110001	101100	011111	111010
000001	100100	010111	001010	110010	101111	011100	111001
100010	000111	110100	101001	010001	001100	111111	011010

(4) Each received word $T(c)$ should be found within the table and interpreted as the code word c at the top of its column in the standard array. For example, $T(c) = 010011$ is considered an incorrect transmission of the code word $c = 001011$. It can then be decoded as $D(c) = 001$—the leftmost 3 bits of c.

An astute observer will notice that the table generated in Example 5 isn't unique. In particular, the coset leader in the last row could have alternatively been chosen as 010001 or 001100. As such, this error-correcting decoding scheme can only reliably correct errors where a single bit has flipped. This is to be expected, however: Theorem 9 indicates that the minimum distance between code words in this example is 3, so Proposition 6 tells us that this is all we can expect. A more general result, in fact, tells us that our method of decoding by coset leaders does, in fact, provide nearest-neighbor error correction just as well as any other algorithm.

Theorem 12. *In coset decoding, a received word $T(c)$ is decoded as a code word c such that $d(T(c), c) \leq d(T(c), a)$ for all $a \in C$* [4, Theorem 31.4] [5, Theorem 16.17].

Proof. Let $T(c) = r$ be a received word that is decoded as c. r is in some coset $x + C$, where x is the coset leader. $c + r = x$ so we find that $d(c, r) = \text{wt}(r + c) = \text{wt}(x)$. Also, $r = c + x$—so $a + r = a + (c + x) = (a + c) + x$. Since $(a + c) \in C$, $a + r \in x + C$. Since the coset leader x has minimum weight in its coset $x + C$, $d(a, r) = \text{wt}(a + r) \geq \text{wt}(x)$. We conclude that $d(c, r) = \text{wt}(x) \leq d(a, r)$. □

5. Conclusion

In this discussion, we have provided only a superficial introduction to algebraic coding theory. While the basic notions we introduced are critical throughout the entire field, the encoding and decoding schemes we introduced are simplistic. For a comprehensive, accessible introduction to coding theory, an interested reader may want to begin with Jürgen Bierbrauer's *Introduction to Coding Theory* [3]. Additionally, John Baylis' *Error-Correcting Codes* [1] provides an in-depth analysis of the topics covered in this paper and more. The more advanced reader with a background in computer science may look to the works of Elwyn Berlekamp, whose seminal research recorded in *Algebraic Coding Theory* [2] introduced many of the error-correcting codes used in modern communication technologies [6]. Those interested in coding theory from the perspective of a pure mathematician may also look to J.H. van Lint's *Introduction to Coding Theory* [7], which provides a dense and rigorous treatment of the topic at a graduate level.

REFERENCES

[1] J. Baylis, *Error-Correcting Codes: A Mathematical Introduction*, First Edition, Chapman & Hall, 1998.

[2] E. R. Berlekamp, *Algebraic Coding Theory*, Revised edition, World Scientific, 2015.

[3] J. Bierbrauer, *Introduction to Coding Theory*, Chapman & Hall/CRC, 2005.

[4] J. A. Gallian, *Contemporary Abstract Algebra*, Ninth edition, Cengage Learning, 2017.

[5] R. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, Fifth edition, Pearson Addison Wesley, 2004.

[6] R. Sanders, *Elwyn Berlekamp, game theorist and coding pioneer, dies at 78*: *Berkeley News* (2019), `https://news.berkeley.edu/2019/04/18/elwyn-berlekamp-game-theorist-and-coding-pioneer-dies-at-78/`. Accessed 2019.

[7] J.H. van Lint, *Introduction to Coding Theory*, Third edition, Springer-Verlag, 1999.

Email address: `shaw5@unbc.ca`