ERROR-CORRECTING CODES AND COMBINATORIAL GAME THEORY

ISHAN JOSHI

1. Introduction

The theory of error correcting codes, also known as coding theory, began in the late 1940's with the publications of Claude Shannon. The key problem in coding theory is communication over a noisy channel. Suppose 2 people wish to communicate with each other through a connection. However, this connection may not always transmit the correct message, and instead may alter some of the data. The goal of coding theory is to transmit the message in a way such that it is always possible to recover the original message despite the message being corrupted. In order to do this we add redundancy to the code in efficient ways. Coding theory is incredibly and increasingly important in the modern world in order to reliably transfer information. In this paper we discuss the basic elements of coding theory, as well as its surprising and deep connection the combinatorial game theory, particularly the connection between NIM and Hamming codes. Later, we also discuss a generalization of NIM, as well as its connection to linear error correcting codes.

2. Preliminaries

We begin by defining a few terms which are related to the discussion.

Definition 2.1. A codeword is the intended message. The set of all codewords is called a code.

Since we are dealing with computers, messages are represented as binary strings.

Definition 2.2. The word length of a code is the number of digits in each message. This is denoted as n. The base of a code is the number of unique digits used to represent messages, and is denoted as B. We also call a code that corrects e errors an e-error correcting-code.

Definition 2.3. A linear code is a code where a linear combination of any codewords results in another codeword.

Definition 2.4. The Hamming distance between 2 codewords is the number of positions in which they are different. The Hamming distance of a code is the minimal Hamming distance between any two codewords.

Hamming distance is extremely important, as we will use it as a metric to measure how similar two binary strings are. The Hamming distance also helps us determine how many errors can be corrected.

Date: November 17, 2025.

Theorem 2.5. *A code corrects up to e errors if and only if it has a Hamming distance greater than or equal to $2e+1$.*

Proof. We begin with the forward direction. Let A and B be two codewords from the code. Then A and B must differ in at least $2e+1$ locations. If we consider any message M, if M is different from A in at most e positions, then it is different from B in at least $e+1$ positions. However, this is only accounting for e errors, so the M must be interpreted as A. Similarly if a code can be interpreted as B with up to e errors, than it cannot be interpreted as P. Now we prove the reverse direction. If the code is an e-error correcting code, then if M is a message which is interpreted as A, the Hamming distance between A and M can be at most e. However, since M can only be interpreted as A, the Hamming distance between M and any other codeword B, must be at least $e+1$. Therefore, the distance between A and B must be at least $e+(e+1)=2e+1$, as desired. □

We have a similar theorem for error detection.

Theorem 2.6. *A code detects up to e errors if its Hamming distance is greater than or equal to $e+1$.*

Proof. Any two codewords must differ in at least $e+1$ positions, meaning that if a message is less than $e+1$ positions away from a codeword, at least one error has occurred. □

Definition 2.7. A code of length n correcting at most e errors is called a perfect code if, for every string s of length n, there is exactly one codeword such that s differs from that codeword in at most e places.

If the code is a perfect code with word length n and a maximum of e, then for every message sent, there can be $\binom{n}{1}+\binom{n}{2}\dots\binom{n}{e}+1$ errors. Since we cannot have 1 message which corresponds to 2 different codewords, and there are 2^n messages, the amount of codewords must be:

$$\frac{2^n}{\binom{n}{1}+\binom{n}{2}\dots\binom{n}{e}+1}=\frac{2^n}{\sum_{k=0}^{e}\binom{n}{k}}$$

When the code is not a perfect packing, the amount of codewords must be less than this amount.

3. NIM and Hamming codes

We now introduce the famous Hamming code, as well as its connection the the game of NIM.

Definition 3.1. A Hamming code is a 1-error correcting code.

We now show examples of Hamming codes where the word lengths are 3 and 7. When the word length is 3, it is quite simple to construct the Hamming code. We want to construct a perfect code, so there must be $\frac{2^3}{3+1}=2$ code words. This is quite simple, as we have the two codewords be 000 and 111. We can visualize this as such:

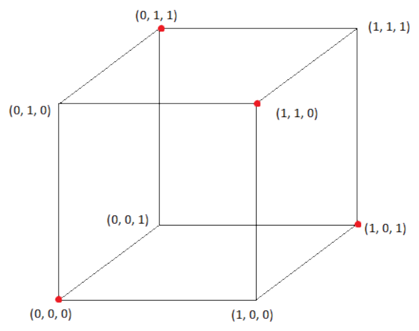

Figure 1. A cube where each vertex represents a 3 bit number

It can be visually seen that the two codewords 000 and 111 have a Hamming distance which is equal to 3. It can also be seen that if any message other than the two codewords were to be transmitted, then it would be very simple to figure out the original message.

Now we can construct the Hamming code of length 7. This would have $\frac{2^7}{7+1} = 16$ codewords. Similarly to $n = 3$, the Hamming distance of this code must be at least 3. It turns out that the Hamming code with $n = 7$ is as follows:

0000000	0000111	0011001	0011110
0101010	0101101	0110011	0110100
1001011	1001100	1010010	1010101
1100001	1100110	1111000	1111111

A fair question to ask is: How are these values computed? One method the reader may take is simply take the string 0000000, and then to find the next string (In lexicographic order) such that the Hamming distance from all the other strings is at least 3. This can be done with a computer. However, there is another way to do this, and for that we introduce the game of NIM.

Definition 3.2. NIM is an impartial game where two players alternate taking stones from piles of stones. For any position, there exist piles of stones which the players can take from. On their turn, each player can remove any number of stones from exactly one pile, and the first player to not have a move loses.

Now, we would want to somehow express a NIM position.

Definition 3.3. We can represent a NIM pile of size n as $*n$. We represent a NIM position as a sum of NIM piles: $*a_1 + *a_2 \dots * a_n$. $*n$ is part of a class of game values called the *nimbers*.

Definition 3.4. The outcome class of a game is the types of ending for that game. For impartial games there are 2 outcome classes:

- $\mathcal{N}$ - This means that the next player wins. In other words, the player who starts wins.
- $\mathcal{P}$ - This means that the previous player wins. In other words, the player who starts loses.

Theorem 3.5. *Any NIM game can be represented as a binary string.*

Proof. We can do this by observing that if we have 2 piles of the same size, then deleting them does not change the outcome of the game. Therefore we can assume there is at most 1 pile of every size. So, we can represent a NIM game as $\dots a_5 a_4 a_3 a_2 a_1$, where each a_k is 0 if there are no piles of size k and 1 if there is 1 pile of size k. □

Theorem 3.6. *The codewords of the Hamming code are the same as the binary representation of the NIM positions with outcome class $\mathcal{P}$.*

Proof. We let $\mathfrak{p}$ be the set of codewords, and let $\mathfrak{n}$ be the set of messages. When we consider all the $\mathfrak{n}$ positions (as binary strings interpreted as NIM games), it is at most 2 positions away from some codeword. By making a single move in the corresponding NIM game, we can move from $\mathfrak{n}$ to $\mathfrak{p}$. Now, if we have a codeword, we can only change it by 2 positions or less. Therefore, from a $\mathfrak{p}$ position, we can only move to a $\mathfrak{n}$ position. The $\mathfrak{p}$ and $\mathfrak{n}$ positions behave exactly the same as the $\mathcal{P}$ and $\mathcal{N}$ positions of NIM. By a later theorem, it can be shown that they are the same. □

4. Turning Turtles

We generalize the results in the previous section by using the game Turning Turtles. In fact, NIM is a specific case of Turning Turtles.

Definition 4.1. Turning Turtles is a game in which there are n turtles placed in a row. These turtles can either be upright, or upside-down. On their turn, a player must flip between 1 and k turtles, and the leftmost turtle flipped must be flipped from upside-down to upright.

Turning Turtles can easily be expressed as a binary string. We can just have a 1 where there is a turtle that is upside down, and use a 0 when there is a turtle that is upright.

Theorem 4.2. *The $\mathcal{P}$ positions of Turning Turtles with $k = 2e$ form a linear e-error correcting code.*

Proof. Since we cannot move from a $\mathcal{P}$ position to another $\mathcal{P}$, the distance between any two $\mathcal{P}$ positions must be at least $2e + 1$, otherwise one could flip some number of turtles in order to move between $\mathcal{P}$ positions. Therefore, by Theorem 1.5, we know that the code corrects up to e errors. Now, we must show that the code is linear. If a position is a codeword, it must be a $\mathcal{P}$ position. Therefore it must have value 0. So, when we NIM sum any number of codewords, we get another codeword, meaning the code is linear. □

Definition 4.3. The Golay Code of length 23 is a perfect code which can detect 6 errors and correct 3.

Corollary 4.4. *The $\mathcal{P}$ positions of Turning Turtles form the Golay Code of length 23.*

Proof. This is simply a special case of the previous theorem, where $k = 3$. □

5. Conclusion

There is far more to this subject than what is presented here. There are many more connections between coding theory and combinatorial game theory. An example is lexicographic codes, which a further generalization of some of the codes seen in this paper. Some other codes which the reader may find interesting are Reed-Solomon codes and Group codes.

References

[BdRS09] José Joaquín Bernal, Ángel del Río, and Juan Jacobo Simón. An intrinsical description of group codes. *Designs, Codes and Cryptography*, 51(3):289–300, 2009.

[CS86] John Conway and N Sloane. Lexicographic codes: error-correcting codes from game theory. *IEEE Transactions on Information Theory*, 32(3):337–348, 1986.

[RS] Simon Rubinstein-Salzedo. *Combinatorial Game Theory.*

[SS59] Harold S Shapiro and Daniel L Slotnick. On the mathematical theory of error-correcting codes. *IBM Journal of Research and development*, 3(1):25–34, 1959.

[RS] [BdRS09] [SS59] [CS86]