

Project 2: Error-Correcting Codes

1 Introduction

Computer and digital communications are permeating the whole fabric of our technological society. There are too many examples to list them all here but some include satellite transmission of data, intercontinental communications, and buying products online. By digital communication, we mean that information is transmitted in strings of 0's and 1's. Such strings are called binary messages and are encoded in such a way as to convey information. There are three reasons to encode the data that is about to be transmitted—efficiency, error detection and/or correction, and secrecy. We want our transmitted messages to be efficient in that we want to compress data as much as possible in order to save transmission time or storage space. Also, it is not hard to imagine that errors are sometimes introduced into such messages, by static or other types of interference. Thus, we want messages to be sent so that errors can be detected and corrected, if possible. In addition, we may want our messages to be secret, so that unauthorized persons cannot read the data.

In this project, we are going to learn about error detection and correction. A first attempt to determine an error in the message was made by augmenting the message with an extra numeral, 0 or 1, to make the number of 1's even. For example, if the data to be sent was “10111”, then the augmented message “101110” was sent since the number of 1's in “10111” is even. Similarly, if the message was “10110”, then “101101” was sent. This type of error detecting is called a **parity check**. If a message was received with an odd number of 1's, then the receiver would know that there was an error. However, the receiver would not be able to tell where the error was, nor if there were three or five errors instead of just one. Worst of all an even number of errors would go undetected.

Error-correcting codes generalize the idea of parity checks in such a way that you can tell where the errors are (and hence correct them). The theory was pioneered by Richard W. Hamming in the early 1950s when he was working at Bell Laboratories. In this project, we will study one of the simplest of all such codes.

2 A New Number System and its Vector Spaces

Definition: A **word** (of length n) is an n -tuple of 0's and 1's. Such a word is also call a **string of length n** .

Since we shall be using only 0's and 1's, we will want a number system that reflects this. One such number system is the **integers mod 2**, denoted \mathbb{Z}_2 . In $\mathbb{Z}_2 = \{0, 1\}$, addition and multiplication are almost as usual, except that $1 + 1 = 0$. The addition and multiplication tables for \mathbb{Z}_2 are given below.

$$\begin{array}{c|c|c} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ \hline 1 & 1 & 0 \end{array} \qquad \begin{array}{c|c|c} * & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 1 \end{array}$$

All the arithmetic properties (associativity, commutativity, distributivity, etc.) hold, except the number 1 is its own negative (since $1 + 1 = 0$).

Just as \mathbb{R}^n is the set of all n -tuples of real numbers, we denote by \mathbb{Z}_2^n the set of all n -tuples of numbers from \mathbb{Z}_2 , that is n -tuples of 0's and 1's. **Thus \mathbb{Z}_2^n is the set of all words of length n .** Addition and scalar multiplication of vectors (component wise) is defined in \mathbb{Z}_2^n exactly as in \mathbb{R}^n , except that scalars must come from \mathbb{Z}_2 (i.e., scalars are either 0 or 1) and the component addition and multiplication is done as in \mathbb{Z}_2 .

1. For $\mathbf{v} = (1, 1, 0, 0, 1, 0)$ and $\mathbf{w} = (0, 1, 1, 0, 1, 1)$ in \mathbb{Z}_2^6 , find $\mathbf{v} + \mathbf{w}$, $-\mathbf{v}$, and $a\mathbf{v}$ for all a in \mathbb{Z}_2 .
2. Using the definitions of addition and multiplication in \mathbb{Z}_2 , show that if we restrict scalars to elements of \mathbb{Z}_2 , then \mathbb{Z}_2^n satisfies all of the axioms in the Definition of a Vector Space on pp. 490 of your text. Thus, we can say that \mathbb{Z}_2^n is a vector space of \mathbb{Z}_2 .
3. Is the standard basis $\{\mathbf{e}_1 = (1, 0, \dots, 0), \mathbf{e}_2 = (0, 1, \dots, 0), \dots, \mathbf{e}_n = (0, 0, \dots, 1)\}$ a basis for \mathbb{Z}_2^n ? Show \mathbb{Z}_2^n has dimension n .
4. List all the vectors in \mathbb{Z}_2^3 . Count them.
5. How many vectors are there in \mathbb{Z}_2^4 ? In \mathbb{Z}_2^n ?

3 Encoding Messages

We now start to answer the question: How do we *encode* a message so that if a single error occurs in transmission, then that error can be detected and corrected at the receiving end? For our purposes, we are going to take a word of length four (a longer word could be padded with leading zeros as needed and then broken up into segments of length four) and add on three parity checks to accomplish our task.

A message of length four plus three parity checks yields a word of length seven. For this reason, we will work in \mathbb{Z}_2^7 . There are four particular vectors that are important to us:

$$\mathbf{u}_1 = (1, 0, 0, 0, 0, 1, 1), \mathbf{u}_2 = (0, 1, 0, 0, 1, 0, 1)$$

$$\mathbf{u}_3 = (0, 0, 1, 0, 1, 1, 0), \mathbf{u}_4 = (0, 0, 0, 1, 1, 1, 1)$$

These vectors were chosen to give the specific encoding we are going to work with, that is, sending the word of length four plus the three parity checks.

1. Show that the vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ are linearly independent.

Since $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ are linearly independent, they form a basis for the four-dimensional subspace that is spanned by $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$. Let $C_{7,4}$ denote the subspace spanned by $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$, that is, $C_{7,4} = \text{Span}\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4\}$.

Definition: A **code** is a k -dimensional subspace of \mathbb{Z}_2^n . The code $C_{7,4}$ is called a $(7, 4)$ **Hamming code**.

To encode a word,

$$(x_1, x_2, x_3, x_4) = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3 + x_4\mathbf{e}_4,$$

we send

$$x_1\mathbf{u}_1 + x_2\mathbf{u}_2 + x_3\mathbf{u}_3 + x_4\mathbf{u}_4 = (x_1, x_2, x_3, x_4, x_2 + x_3 + x_4, x_1 + x_3 + x_4, x_1 + x_2 + x_4).$$

Thus we sent the original word (x_1, x_2, x_3, x_4) and three parity checks $x_2 + x_3 + x_4$, $x_1 + x_3 + x_4$, and $x_1 + x_2 + x_4$.

For example, to encode $\mathbf{w} = (1, 0, 1, 1)$, we send

$$1 * \mathbf{u}_1 + 0 * \mathbf{u}_2 + 1 * \mathbf{u}_3 + 1 * \mathbf{u}_4 = (1, 0, 1, 1, 0, 1, 0).$$

2. Let $C = \{(x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n \mid x_1 = 0\}$. Is C a code? If so, give a basis for C and state the dimension. If not, show why not.
3. Let $C = \{(x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n \mid x_1 = 1\}$. Is C a code? If so, give a basis for C and state the dimension. If not, show why not.

Consider the following matrix G whose rows are the vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ and \mathbf{u}_4 .

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

4. Show that to encode the word $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$, we send $x^T G$. The matrix G is called a **generating matrix** for the code $C_{7,4}$.
5. List all the vectors in $C_{7,4}$.
6. Encode the following words into a code word in $C_{7,4}$.
- (a) $(0, 1, 1, 0)$
 - (b) $(1, 1, 0, 1)$
 - (c) $(0, 1, 1, 1)$
 - (d) $(1, 0, 0, 1)$

4 Decoding Messages

To *decode* a message means to check the message to determine if there has been an error, to correct any error, and finally to extract the original message. We first discuss the check. Consider the 3×7 matrix

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

with entries from \mathbb{Z}_2 . This matrix is not chosen arbitrarily, but chosen specifically as we shall see.

1. Show that $\text{rank } H = 3$.
2. Show that $H\mathbf{u}_i = \mathbf{0}$ for $i = 1, 2, 3, 4$, where \mathbf{u}_i ($1 \leq i \leq 4$) are as in the previous section.
3. Find a basis for the null space of H (hint: use the previous question). Conclude from this that the null space of H is $C_{7,4}$.

It turns out that if any vector in $C_{7,4}$ is altered in exactly one coordinate, then the resulting vector is not in $C_{7,4}$. Thus, using the $C_{7,4}$ code, we detect a word with a single error.

4. Using the matrix H , determine if the following words are in $C_{7,4}$. (Note: we could compare x , y , and z with the list of vectors obtained in the previous section, but those computations

will tell us how to correct any word with a single error.)

$$\mathbf{x} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \mathbf{z} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

5. Looking at the matrix H , you will notice that the columns represent, in order, the binary representation of the numbers 1 through 7. Also, every nonzero vector of \mathbb{Z}_2^3 is a column of H . (Why?) Thus, if \mathbf{w} is a word in \mathbb{Z}_2^7 and $H\mathbf{w} \neq \mathbf{0}$, then $H\mathbf{w}$, as a vector in \mathbb{Z}_2^3 , is a column of H . Show that if $H\mathbf{w}$ is the k th column of H and the k th entry of \mathbf{w} is changed, then the corrected vector \mathbf{w}_c will be in $C_{7,4}$.
6. Change an entry of the vectors \mathbf{y} and \mathbf{z} of Question 4 so that the resulting vectors are in $C_{7,4}$.
7. Based on the things you have learned so far in this section, write down the complete decoding process for decoding a word \mathbf{w} of length seven.
8. Consider the following 7-tuples. Consider each to be a message with at most one error in \mathbb{Z}_2^7 . Determine if each vector is in $C_{7,4}$. If it is, decode it. If it is not, correct it and decode the corrected message.
 - (a) $(0, 1, 1, 0, 0, 1, 1)$
 - (b) $(0, 1, 1, 1, 0, 1, 1)$
 - (c) $(1, 0, 0, 1, 1, 1, 1)$
 - (d) $(1, 1, 0, 0, 1, 1, 1)$

At this point, we must emphasize the fact that this code cannot handle all possible combinations of errors (no code can). If there are two errors, the decoding process for this code will give the wrong message and certain combinations of three or more errors will go undetected.