ERROR CORRECTING CODES PAPER OUTLINE

ATTICUS KUHN

Date: August 17, 2020.

Contents

Introduction

Error correcting codes are vital to many digital infrastructures today. In this paper, I will introduce the fundamentals of error correcting codes, provide examples of several error correcting codes, and look at the deeper theory behind them. We will then work towards finally proving some results related to those concepts.

Fundamental Concepts

The classic way of looking at error correcting codes is Alice sending a binary message to Bob. However, she is sending over a

Definition 1. *noisy channel, meaning some of her data might be distorted.*

For example, take the ASCII representation of A, 01000001. If The noisy channel corrupted the 5th bit of the message, it could end up like 01001001, or ASCII for I. This is a different message, and we can solve this problem with error correcting codes.

We also assume that the noise is randomly distributed among the message.

Repetition codes

The most basic ECC is the repetition code, where each bit is repeated 3 times. For example, the message 01000001 would be sent as 000111000000000000000111. If there were a distortion in the message, say 000111 000 000 000 100 000111, then Bob would know 100 was really a distortion of 000. This means that for the message to be corrupted, 2/3 bits in a 3 bit section must be flipped, which is far less likely than just 1 bit being flipped. There are also some downsides to this code, the main one being it is 3 times longer than the original message. One of the central trade offs of error correcting codes is making a code more secure by introduce more redundancy, at the cost of making it longer.

Formalizing concepts

We said before that error correcting codes have redundancy to compensate for errors, but how can we quantify this? We use measures such as Hamming distance to quantify this.

Definition 2. *the hamming distance between 2 messages, $m_1 = (a_1...a_n)$ and $m_2 = (b_1...b_n)$ is d(m_1, m_2)=$|i \in \{1...n\}|a_i \neq b_i|$, or it is the number of characters between the 2 messages which are different.*

Linear Codes

To understand linear codes, we must bring in the vector space $\mathbf{F}_p^k$. We consider a subset, which consists of the valid code words. It is called linear because a sum of vectors in the code is also in the code. A linear code is generated by a generator G, which is a n by k matrix. For example, A [5,3] generator could be G = $\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$, and in this case, the code words would be generated by combinations of the 3 vectors represented. The size of the space is 2^k, so in this case it is 8. To encode a message, we apply G, and to decode a code word, we apply the inverse of G. This decoding procedure works when the message has no errors, but how would it work when it does have some errors? First we must discuss some other concepts.

Definition 3. *In $\mathbf{F}_p^k$, the hamming distance , d(C_1,C_2) is the number of differing coordinates between them. The hamming weight of a code C is $w(C) = d(C, O)$. The minimum distance of C is min($\{d(a, b)|a, b \in c\}$)*

Now here is where the correction comes in. A linear code with minimal distance d can correct errors of weight e if $d2e + 1$. We can actually prove this using a proof by contradiction. Let x' be the received message and let an error of weight e occur. Suppose there was another element $y \in C$ such that $d(x', y) \leq d(x', x) = y$. This means that $d(x, y)d(x, x0) + d(x0, y) \leq 2t \leq d$, which contradicts that d is minimal.

Hamming codes

Let's see how these concepts are applied by looking at 1 specific example of a linear code, Hamming(7,4). It encodes 4 bits of information in 7 bits. It works by transmitting the 4 bits, as well as 3 parity checker bits along with them. For example $p_1 = d_1 + d_4 + d_2$, $p_2 = d_1 + d_4 + d_4$, and $p_3 = d_2 + d_4 + d_3$. The distance in this case is 3, so we can write it as [7,4,3].

Definition 4. *More generally defined, let A be a maximal set of linearly independent vectors in $\mathbf{F}_q^m$, and let n = $|A|$. Let H be a m by n matrix with its columns consisting of all vectors in A. A Hamming code over $\mathbf{F}_q^n$ is defined as the set of vectors x which satisfy Hx = 0.*

Definition 5. *A code C of minimum weight d is called perfect if all the vectors in V are contained in the spheres of radius $t = \frac{d-1}{2}$ about the code words. In this case the spheres are said to cover the space.*

This means that a perfect code can correct t errors or fewer. The Hamming codes are perfect.

1. Some Proofs

It is always good to verify results, especially since this is a class based on proofs. Now let us prove some results associated with perfect codes.

Theorem 6. *The minimum distance of a Hamming code is 3*

We will prove this by a proof by contradiction. Assume that less than 3 of the components of some nonzero vector x in the Hamming code are nonzero. If exactly one component is nonzero, that implies that one of the columns of H is zero, which contradicts the definition of H from before. If 2 are nonzero, then a linear combination of them is 0, which is also a contradiction.

Now assume the minimum distance were greater than 3. This means no linear combination of three columns of H are 0, contradicting the maximality of the set of columns of H. Why is this? To see this, we add a column that is the sum of any two columns of H. The columns of H will be pairwise independent, as any linear combination of the new column with the other column is a linear combination of three columns of the original H.

Theorem 7. *Hamming codes are perfect codes.*

. First let us fix an element $x \in \mathbf{F}_n^q$, and look at Hx. If x is not in the Hamming code, then Hx is a non-zero linear combination of some columns of H. Then, by the maximality of the set of columns of H, Hx forms a linear combination with the ith column v_i of H just being to zero for some i. We then have $\frac{a}{v}$i+bHx = 0, or $\frac{a}{b}$ v_i + Hx = 0. Thus, if we add a/b to the ith index of x, we obtain a string in the Hamming code, so any element of$\mathbf{F}_n^q$ is at most distance 1 from an element of the Hamming code.

Theorem 8. *For any linear [n, k, d] code, d $\leq$ n k + 1. - Called the Singleton Bound*

If we use the map C $\rightarrow$ F_q^{nd+1} which is done by by removing d 1 components of an element of C. This is an injective linear map, as every nonzero element has at least d nonzero components, ensuring that only 0 maps to 0. This means that we have $nd + 1 \geq k$, Rearranging this expression give the desired bound.

2. Further Reading

In case you are interested about learning more with Error Correcting Codes, here are several nice books on the subject.
Pless, Vera. Introduction to the Theory of Error-correcting Codes.
Vanstone, Scott A., and Paul C. Van Oorschot. An Introduction to Error Correcting Codes with Applications.

References

[1] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[2] Kauko Lindström. All nearly perfect codes are known. *Information and Control*, 35(1):40–47, 1977.

[3] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error correcting codes*, volume 16. Elsevier, 1977.

[4] William Wesley Peterson, Wesley Peterson, EJ Weldon, and EJ Weldon. *Error-correcting codes*. MIT press, 1972.

[]