Error-Correcting Codes

Homework 13 and 14 - MAGEF

Diana Morouço Gaspar, 99407

January 10, 2024

Contents

1 Introduction

Any kind of data that needs to be transmitted from a source to a receiver can be corrupted by errors, for example, in wireless or satellite communication, and data storage in hard disks or CDs.

The idea behind error-correcting codes is to take an initial message and start by representing the message as a string of elements of a field (Source Encoding), for example, as a string of bits of 0 and 1.

Afterwards, we encode this message by adding extra bits, the *redundant* bits, obtaining a *codeword* (Channel Encoding). These extra bits will later help us correcting the errors, so that the codeword basically corrects itself. We pass this codeword through the transmission channel, obtaining an output vector, which may contain errors or not.

Finally, to detect if the received vector contains errors and to correct them, the redundant bits that were added in Encoding will help us restore the original codeword, so that the receiver can estimate the original message, as represented in the figure below.

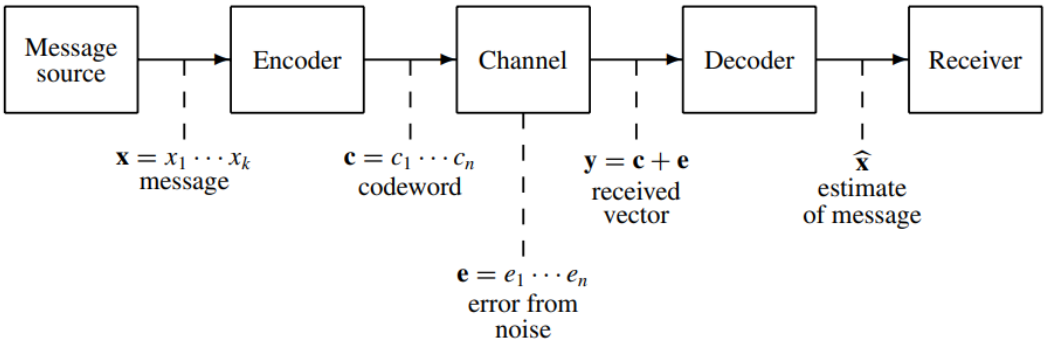

Figure 1: Encoding, transmission and decoding of a message.

1.1 Example - Repetition 3 Code

If we want to transmit the bit **1** through a channel, we repeat it 3 times, encoding it as the string $\mathbf{c} = \mathbf{111}$. If it is corrupted and the received vector is $\mathbf{y} = \mathbf{101}$, the code will detect that there is an error, because the only possible codewords are **000** and **111**.

So, by majority decoding, it will correct the bit that only appears once, decoding it into the correct codeword $\mathbf{c} = \mathbf{111}$. The strings **011** and **110** will also be decoded as **111**.

This code can detect up to two errors in the received message, but it can correct only 1 error. Besides that, from the 3 bits, 2 are used for redundancy, and only 1 for information (we say that the information rate is 1/3), which is not very efficient.

The search for codes that are more complex, but have higher information rates, while still being able to correct some of the errors, motivates the study of Error-Correcting Codes, with countless applications in Engineering and Computer Science.

2 Error-correcting codes

2.1 Linear Code

Definition 1
A [n,k] ***Linear Code*** *C of length n and dimension k over a field with q elements $\mathbb{F}_q$ is a subset $C \subset \mathbb{F}_q^n$ of dimension k, defined by $C = \{x \in \mathbb{F}_q^n : Hx^T = 0\}$.*

*The matrix H is called the **parity check matrix**, with linearly independent rows and dimension $(n-k) \times n$.*

If q is prime, any field $\mathbb{F}_q$ is isomorphic to $\mathbb{Z}_q$, and we work with the integers modulo q.

If q is the power of a prime, $\mathbb{F}_q$ is still a field, but it is no longer isomorphic to the integers modulo q, and the addition and multiplication tables will be different. In sections 2 and 3, we will assume that q is prime.

2.2 Example - ISBN Code

One of the simplest examples of a linear code used in real life is the ISBN 10, used until 2007 to identify every published book, and detect one error that might occur when typing the 10-digit number. The code is defined by

$$C = \{x \in \mathbb{Z}_{11}^{10} : Hx^T = \begin{pmatrix}10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1\end{pmatrix} x^T = 0\}.$$

The first 9 digits can actually only take values between 0-9, but we considered $\mathbb{Z}_{11}$ for a more straightforward definition. The last digit is the "check digit", used for redundancy.

C has length $n = 10$ and dimension $k = 9$, because the parity check matrix has dimension $(n-k) \times n = 1 \times 10$.

Even though this code has a very high information rate, since only 1 of the 10 digits is used for redundancy, it detects only one error (and most 2 errors that result from transposition of 2 digits), and is not able to correct it.

2.3 Weight and distance

To introduce codes with better error correction and detection power, we must define the concepts of distance and weight.

The **distance** $d(x, y)$ between two vectors $x, y \in \mathbb{F}_q^n$ is the number of coordinates in which they differ. The **weight** $w(x)$ of a vector $x \in \mathbb{F}_q^n$ is the number of nonzero coordinates of x.

An important property that characterizes a code C is its **minimal distance** d, which is the smallest distance between any two distinct codewords of C.

Theorem 1
For any $x, y \in \mathbb{F}_q^n$, $d(x, y) = w(x - y)$. If C is a linear code, and $x, y \in C$, then $x - y \in C$ and the minimal distance d of C is the minimum weight of the nonzero codewords.

Example 2
Consider $x, y \in \mathbb{F}_2^7$, $x = (1, 1, 0, 1, 1, 0, 0)$, $y = (0, 0, 1, 1, 0, 1, 0)$.
So, $w(x) = 4$, $w(y) = 3$ and $d(x, y) = w(x - y) = w((1, 1, 1, 0, 1, 1, 0)) = 5$.

Proposition 1
Let C be a linear code with minimal distance d. Then, C detects u errors if and only if $d > u$. C corrects t errors if and only if $d \geq 2t + 1$.

This allows us to introduce one of the most common decoding algorithms, *Syndrome Decoding.*

3 Syndrome Decoding

3.1 Syndrome Decoding Algorithm

Definition 2
*The **syndrome** of $x \in \mathbb{F}_q^n$ is $syn(x) = Hx^T$.*

From the definition of H, if $x \in C$, then $syn(x) = 0$. Therefore, if the received vector x has nonzero syndrome, then x is not a codeword, and we know that occurred at least one error during the transmission.

Proposition 2
Let $x = c + e$ be the received vector, $c \in C$ the correct codeword, and e the error. Then, $syn(x) = syn(c) + syn(e) = syn(e)$, where we used that $syn(c) = 0$ because it is a codeword.

Syndrome Decoding arises from this last proposition, where u is the vector we want to decode, and we want to determine the error e so that we can decode x as $c = x - e$.

If we receive the vector x, in first place, we compute its syndrome. The goal is to find the the vector e with syndrome $syn(e) = syn(x)$ with smallest weight possible.

For a $[n, k, d]$ linear code over $\mathbb{F}_q$, there are q^n vectors in $\mathbb{F}_q^n$, q^k codewords and, because H has rank $n - k$, every vector in $\mathbb{F}_q^{n-k}$ is a syndrome.

Therefore, because the code C is a dimension k subgroup of $\mathbb{F}_q^n$, from Lagrange's Theorem, the distinct cosets $x + C$ partition $\mathbb{F}_q^{n-k}$ into q^{n-k} sets of size q^k.

The *weight* of a coset is the smallest weight of a vector in the coset, and any vector of smallest weight in the coset is a **coset leader**.

Because every coset of weight at most $t = \lfloor (d-1)/2 \rfloor$ has a unique coset leader, if we find the coset leader of the coset with syndrome equal to $syn(x)$, then the error e is the coset leader, and we decode $c = x - e$.

3.2 Example - [7,4,3] Hamming Code

The [7,4,3] linear Hamming Code, with minimal distance $d = 3$ is a 1-error correcting code, $t = \lfloor (d-1)/2 \rfloor = 1$, defined by the parity check matrix below.

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

If the vector $x = (1, 0, 1, 0, 0, 0, 0)$ is received, we want to find the transmitted codeword $c = x - e$.

We have $syn(x) = Hx^T = (1, 0, 1)$. Because C only corrects one error, e will be the unique vector with weight 1 and syndrome $(1, 0, 1)$. In this case, it is easy to get $e = (0, 1, 0, 0, 0, 0, 0)$, because the syndrome of a weight 1 vector corresponds to a column of H.

Finally, we find out that $c = x - e = (1, 1, 1, 0, 0, 0, 0)$ is the transmitted codeword.

4 Other Codes over Finite fields

In the previous sections, we considered codes over a finite field $\mathbb{F}_q$ with q prime. However, if q is the power of a prime, there are many other types of linear codes, which we will not get into, as they would require a much longer explanation.

Perhaps the most widely used are the **Reed-Solomon** codes, a type of cyclic code. To work with cyclic codes, we make a correspondence between a vector and a polynomial (over

a finite field), so that each entry of the vector is the coefficient of a term of the polynomial. This way, shifting each entry of the vector to the right (and the last entry becomes the first one) is the same as multiplying the corresponding polynomial by x.

Among the many applications of Reed-Solomon codes are QR Codes, space transmission (they were used in the Voyager program) and CDs.

For CDs, an error in the reading of the data may be caused by a scratch in the CD, for example. Consequently, the errors are frequently in consecutive elements of the string, called a **burst**. If we know that the error not only has weight l, but is also a burst, we may be able to correct even more errors than if we knew just its weight.

References

[1] W. Huffman, V. Pless, *Fundamentals of Error Correcting Codes*, 2003.

[2] P. Martins Rodrigues, *Lecture Notes from Introduction to Coding Theory.*

[3] ISBN (Wolfram). `https://mathworld.wolfram.com/ISBN.html`