QR Codes

— Outline —

Henry Blanchette

Contents

1 Introduction **2**
1.1 The Coding Problem . . . 2
1.2 Transmission Specifications . . . 2

2 Shannon's Theorem **3**

3 Linear Codes **5**
3.1 Linear Codes . . . 5
3.2 Hamming Codes . . . 5
3.3 Weight Enumerators . . . 5

4 The Binary Golay Code **5**
4.1 Hadamard Codes . . . 5
4.2 The Binary Golay Code . . . 5

5 Cyclic Codes **5**
5.1 Definitions . . . 5
5.2 Generator Matrix and Check Polynomial . . . 5
5.3 Zeros of a Cyclic Code . . . 5
5.4 Idempotent of a Cyclic Code . . . 5
5.5 Alternative Representations of Cyclic Codes . . . 5
5.6 Quadratic Residue (QR) Codes . . . 5

1 Introduction

1.1 The Coding Problem

Consider the scenario where a person, Alex, wants to send a message another person, Beth, via a noisy electrical channel. To facilitate such a transmission, a few pieces of equipment and processes are involved. First, Alex comes up with the message that he would like to transmit and writes it down in the form of an m-tuple,

$$\vec{a} = (a_m, \ldots, a_m).$$

Then, Alex uses a machine called a **encoder** that maps $\vec{a}$ to an n-tuple,

$$\vec{x} = (x_1, \ldots, x_n).$$

$\vec{x}$ is a **codeword** - one of some number of possible codewords in the encoder's image. Note that there must be at least as many codewords as there are possible original messages.

Next, x is transmitted to Beth as an electrical signal along a channel. During the transmission, some random noise $\vec{e}$ is added to the signal, where $\vec{e}$ is a n-tuple. The resulting signal that Beth recieves is $\vec{r} := \vec{x} + \vec{e}$.

In an attempt to correctly recover $\vec{a}$ from $\vec{r}$, Beth uses a machine called an **decoder**. The decoder calculates the most likely codeword $\vec{x}'$ that could have resulted in $\vec{r}$, and then outputs the message $\vec{a}'$ that corresponds to $\vec{x}'$ via inverse-encoding. If $\vec{r}$ is exactly a codeword, then $\vec{x}' = \vec{r}$. However, if $\vec{r}$ is not exactly a codeword, then the decoder finds the *closest* codeword to $\vec{r}$ in the space of the encoder's codomain (recall that are a subset of this codomain, the encoder's image).

The **coding problem** is the problem of devising an encoder/decoder pair that efficiently (in regards to some set of concerned features) and accurately faciliates transmissions like the one above. A construction of codewords of length n is referred to as a **code**, C.

1.2 Transmission Specifications

There is one possible "solution" to the coding problem that illustrates why specifying some more bounds on the transmission process is useful. Say there is a similar setup to the one in the previous section, and Alex wants to send Beth information about his coin-tossing prowess. After each toss, Alex sends the result to Beth in the form of a 0 for heads and 1 for tails. Alex tosses his coin at a speed of t tosses per minute. The channel connecting Alex and Beth is noisy such that there is a chance p that a bit is sent incorrectly, and a chance $q := 1 - p$ that bit is sent correctly. This channel is called a **binary symmetric channel**. Also, this channel only allows Alex to send $2t$ bits per minute and only during his coin-tossing session. When Alex gets a heads he transmits 0, and when he gets a tails he transmits 1. Alex decides to carry out his session for T minutes. At the end of the T minutes, Beth looks at the bits she received. She knows that a fraction p of them are incorrect, because of the channel's error rate. How could she reduce her decoding error lower than p?

Consider setup differing only in one aspect: there is no time constraint. Then instead of just sending one 0 or 1 for each toss, Alex can send N 0s or 1s for each toss. Then, Alice's decoder can decodes each section of N bits by taking the most common bit. Using this method, the probability of decoder error is

$$P_e(N) := \sum_{0 \le k \le N/2} \binom{N}{k} q^k p^{N-k}. \tag{1}$$

Furthermore,

$$\lim_{N \to \infty} P_e(N) = 0$$

so Alex and Beth can achieve arbitrarily accurate communication given enough time. The time constraint was an important obstacle after all!

2 Shannon's Theorem

The obviously unsatisfying aspect of the "solution" in section 1.2, other than the ignorance of a time constraint, is that it is extremely wasteful. There should have to be a good excuse for having to send a message any more than once. In turns out that, in fact, there are much better ways of achieving accuracy even within time and other constraints. Shannon's theorem states that, in the same situation as originally described in 1.2, Alex and Beth can still achieve arbitrarily small error probability.

Definition 2.0.1. Let C be a code with codewords of length n. Then the **information rate** of the code is

$$R := n^{-1} \log_2 |C|.$$

Definition 2.0.2. Let $\vec{x}, \vec{y} \in \mathbb{F}_2^n$. Then their **Hamming distance** is

$$d(\vec{x}, \vec{y}) := |\{i : x_i \neq y_i\}|$$

Suppose we have a binary symmetric channel with transmission-error probability p, and $q := 1-p$. Let $C = \{\vec{x}_i\}$ be a code of M words of length n, where each of the words are encoded to with equal probability. Suppose the decoder uses **maximum-likelihood** decoding i.e. the decoder decodes a received signal to the codeword that was most likely to be the original signal. Let P_i be the probability that the decoder is incorrect given that $\vec{x}_i$ is transmitted. So, the probability of an incorrect decoding is

$$P_C := M^{-1} \sum_{i=1}^{M} P_i \tag{2}$$

Finally, define

$$P^*(M, n, p) := \min \{P_C : C \text{ is a code with } M \text{ words of length } n\} \tag{3}$$

Theorem 2.0.1. (Shannon's theorem)

$$0 < R < 1 + p \log p + q \log q \implies \lim_{n \to \infty} P^*(M_n, n, p) = 0$$

where $M_n := 2^{[Rn]}$ and all logarithms have base 2.

Proof. Observe that the probability of an error pattern with w errors is $p^w q^{-w}$, which depends only on w. Denote the probability of receiving $\vec{y}$ given that $\vec{x}$ is transmitted by $P(y|x)$. Then also note that $P(\vec{y}|\vec{x}) = P(\vec{x}|\vec{y})$.

The number of errors in a received word is a random variable with expected value np and variance $np(1-p) = npq$. Let $\epsilon > 0$ and

$$b := \left(\frac{np(1-p)}{\epsilon/2} \right)^{1/2}.$$

Then by Chebyshev's inequality (TODO: cite 1.4.1), we have

$$P(w > np + b) \leq \frac{\epsilon}{2} \tag{4}$$

Let $\rho := \lfloor np + b \rfloor$. Then since $p < \frac{1}{2}$, ρ is less than $\frac{n}{2}$ when n is sufficiently large (TODO: show this). Define

$$B_\rho(\vec{x}) := \{\vec{y} : d(\vec{x}, \vec{y}) \leq \rho\} \tag{5}$$

which is the ball of radius ρ around $\vec{x}$. Then (TODO: cite Lemma 1.4.3) yields that

$$|B_\rho(\vec{x})| = \sum_{i \le \rho} \binom{n}{i} < \frac{1}{2}\binom{n}{\rho} \le \frac{n}{2}\frac{n^2}{\rho^\rho (n-\rho)^{n-\rho}} \tag{6}$$

We will use the following estimates:

$$\frac{\rho}{n}\log\frac{\rho}{n} = \frac{1}{n}\lfloor np+b\rfloor \log\frac{\lfloor np+b\rfloor}{n} = p\log p + O(n^{-1/2}) \tag{7}$$

$$\lim_{n\to\infty}\left(\left(1-\frac{\rho}{n}\right)\log\left(1-\frac{\rho}{n}\right)\right) = q\log q + O(n^{-1/2}) \tag{8}$$

Define

$$f : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2 \tag{9}$$

$$(\vec{u},\vec{v}) \mapsto \begin{cases} 0 & \text{if } d(\vec{u},\vec{v}) > \rho \\ 1 & \text{if } d(\vec{u},\vec{v}) \le \rho \end{cases} \tag{10}$$

For $\vec{x}_i \in C$ and $\vec{y} \in \mathbb{F}_2^n$, define

$$g_i : \mathbb{F}_2^n \to \mathbb{Z}$$
$$\vec{y} \mapsto 1 - f(\vec{y},\vec{x}_i) + \sum_{j\neq i} f(\vec{y},\vec{x}_j)$$

g_i is a function that counts the number of codewords other than $\vec{x}_i$ such that $d(\vec{x}_i,\vec{y}) \le \rho$.

Now, choose M codewords $\vec{x}_1,\dots,\vec{x}_M$ at random independently. Then the decoding algorithm is as follows. Suppose the decoder receives $\vec{y}$. If there is exactly one codeword $\vec{x}_i$ such that $d(\vec{x}_i,\vec{y}) \le \rho$, i.e. $!\exists i : g_i(\vec{y}) = 0$, then decode $\vec{y}$ as $\vec{x}_i$. If there is not such $\vec{x}_i$, then the decoder has detected an error, and if it must decode anyway it outputs $\vec{x}_1$ as a default.

So P_i, the probability of error (as decidedd by the decoder algorithm), is such that

$$P_i \le \sum_{\vec{y}\in\mathbb{F}_2^n} P(\vec{y}|\vec{x}_i)g_i(\vec{y}) = \sum_{\vec{y}} P(\vec{y}|\vec{x}_i)(1-f(\vec{y},\vec{x}_i)) + \sum_{\vec{y}}\sum_{j\neq i} P(\vec{y}|\vec{x}_j)f(\vec{y},\vec{x}_j)$$

where the right term is the probability that the received word $\vec{y}$ is not in $B_\rho(\vec{x}_i)$. By equation 4, $P_i \le \frac{\epsilon}{2}$. Then,

$$P_C \le \frac{\epsilon}{2} + M^{-1}\sum_{i=1}^{M}\sum_{\vec{y}}\sum_{j\neq i} P(\vec{y}|\vec{x}_i)f(\vec{y},\vec{x}_j).$$

Since $\vec{x}_1,\dots,\vec{x}_M$ were chosen at random, we have

$$\begin{aligned} P^*(M,n,p) &\le \frac{\epsilon}{2} + M^{-1}\sum_{i=1}^{M}\sum_{\vec{y}}\sum_{j\neq i}\mathcal{E}(P(\vec{y}|\vec{x}_i))\mathcal{E}(f(\vec{y},\vec{x}_j)) \\ &= \frac{\epsilon}{2} + M^{-1}\sum_{i=1}^{M}\sum_{\vec{y}}\sum_{j\neq i}\mathcal{E}(P(\vec{y}|\vec{x}_i))\cdot\frac{|B_\rho|}{2^n} \\ &= \frac{\epsilon}{2} + (M-1)2^{-n}|B_\rho|. \end{aligned}$$

Next, applying the estimates 7, we have

$$P^*(M,n,p) \le n^{-1}\log(P^*(M,n,p) - \frac{\epsilon}{2}) \le n^{-1}\log_M -(1 + p\log p + q\log q) + O(\sqrt{n})$$

where $O(\sqrt{n})$ is a polynomial that is asymptotically equivalent to $\sqrt{n}$. Lastly we can substitute M_n for M, allowing the number of words, M in the code to depend on n, and use the restriction on R, $0 < R < 1 + p\log p + q\log q$, to get

$$n^{-1}\log(P^*(M_n,n,p) - \frac{\epsilon}{2}) < -\beta < 0$$

from the definition of R (definition 2), for

$$n > N := \frac{-\log \frac{\epsilon}{2}}{\beta}.$$

In other words,

$$P^*(M_n, n, p) < \frac{\epsilon}{2} + 2^{-\beta n}.$$

Thus

$$P^*(M_n, n, p) < \frac{\epsilon}{2} + 2^{-\beta n} < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

□

This result was first published in C.E. Shannon's paper *Mathematical theory of communication* (1948), and is popularly recognized as origin of coding theory. The key concept that the theory illustrates is that **good codes** exist, where a good code is a code both usefully accurate yet more efficient than the unenightening code presented in section 1.2.

3 Linear Codes

3.1 Linear Codes

3.2 Hamming Codes

3.3 Weight Enumerators

4 The Binary Golay Code

4.1 Hadamard Codes

4.2 The Binary Golay Code

5 Cyclic Codes

5.1 Definitions

5.2 Generator Matrix and Check Polynomial

5.3 Zeros of a Cyclic Code

5.4 Idempotent of a Cyclic Code

5.5 Alternative Representations of Cyclic Codes

5.6 Quadratic Residue (QR) Codes