Coding Theory

Boran Erol

March 2024

1 Introduction

You'd like to a send message to another person using a noisy channel by adding redundancy to the original message.

Definition 1. An **error correcting code** is an injective map $Enc : \Sigma^k \to \Sigma^n$.

Σ is called the alphabet, $q = |\Sigma|$, and Σ^k is called the message space. Often, $q = 2$. q might depend on n.

k is called the message length (sometimes dimension). n is called the (block) length. $C = \text{Ran}(Enc)$. C is sometimes called the code by an abuse of terminology. Any string $y \in C$ is called a codeword.

Definition 2. The **rate** of a code is $\frac{k}{n}$.

Notice that larger rates correspond to smaller redundancies.

The ambient space Σ^n can be seen as a sequence of n letters, as a vector of size n or even as a function. Which interpretation we use depends on the context. For example, the vector interpretation is useful when Σ is a field.

You can't flip more than half of the bits – it's impossible to recover the message. Prove this rigorously (look at previous notes).

We say that a code is asymptotically optimal if the ratio of the lengths of the message and the codeword becomes 1.

Definition 3 (Hamming Distance)**.** The

The Hamming Distance turns Σ^n into a metric space.

The capabilities of a code crucially depend on the noise model.

Hamming's model of bounding the total number of errors assumes nothing about the nature of noise, whereas Shannon's binary symettric channel with crossover probability p assumes perfect knowledge about the channel.

It should be noted that Hamming's channel is unbounded.

1.1 Basics

2 Linear ECCs

Encoding is always efficient for linear ECCs.

In general, given G and a received word z, finding the y in the rowspace of G which is closest to z is NP-Hard. However, for some codes, decoding can be done in polynomial time.

Let's not talk about decoding. Let's simplify the problem. How can we tell whether $z = y$ for some $y \in C$ efficiently? This is not easy using a spanning set. Instead, we need orthogonality. We're going to use $C^{\perp}$. Notice that $C^{\perp}$ is an $[n, n-k]_q$code and is called the **dual code of C**. Let $C^{\perp}$ act on $x \mapsto xH$. H is called the **parity check matrix for C** and the rows of H are orthogonal to every codeword of C.

In linear codes, $d(C)$ is just the minimum Hamming weight of a non-zero codeword.

Also, $d(C)$ is the minimum number of non-zero columns of H which are linearly dependent.

In F_2, two vectors are linearly dependent if and only if they're the same vector. Therefore, notice that if H has unique non-zero columns, the minimum distance of C is at least 3.

Notice that codes with high rate corresponds to short and fat parity check matrices H.

Based on this discussion, one thing we can do is to put all non-zero unique vectors of size $n - k$ as the columns of H. This gives us the **Hamming code**.

This gives us a $r \times 2^r - 1$ matrix and produces a $[2^r, 2^r - r - 1]_2$ code. Notice that $k \approx n - \log(n)$, so the rate of the Hamming code is great.

The Hamming Code is cute in the sense that we can immediately detect which bit was flipped using H. This is because $H(y + e_i) = He_i$, so flipping the ith bit produces the ith column of H.

Hamming codes are perfect codes. The codespace is fully partitioned into Hamming balls. We're optimally packing codewords into our space.

3 The Codeword Test

Theorem 3.1. There is a time $O(n)$ randomized algorithm T such that

$$\Pr[T(C) = \text{ YES}] = \begin{cases} 1 & \exists x : C = C_x \\ \leq 1 - \delta & \forall x : \Delta(C, C_x) \geq \delta \end{cases}$$

Notice that this is a randomized algorithm with one-sided error, so we can decrease the error probability by repetition.

4 Hadamard Codes

If we use the parity check matrices of Hamming codes as generator matrices and add the all zeros column as the first column (mostly for technical convenience), we get the Hadamard code.

Let $x \in F_2^n$. The Hadamard code takes x to C_x:

$$C_x = \begin{bmatrix} \langle x, 000000000\rangle \\ \langle x, 000000001\rangle \\ \langle x, 000000010\rangle \\ ... \\ \langle x, 111111111\rangle \end{bmatrix}$$

Notice that $C_x \in F_2^{2^n}$. The Hadamard code has terrible rate.

Definition 4. For $x \in F_2^r$, define $L_x : F_2^2 \to F_2$ by

$$L_x : a \mapsto x_1 a_1 + ... + x_r a_r$$

L_x are linear polynomials over F_2.

This is kind of confusing: x's are the coefficients and a's are the variables.

In this view, the Hadamard code takes x to the "truth table" of L_x.

Given an x, we construct a linear r-variate polynomial where the coefficients are x_i and evaluate this polynomial at all 2^r points.

Proposition 4.1. Hadamard is an $[2^r, r, 2^r - 1]_2$ code.

Proof. Notice that there are 2^r possible inputs to the Hadamard code and 2^r outputs for all of the inputs. If you list all the codewords in rows, you get the Hadamard Matrix if you use 1s and -1s instead of 0s and 1s.

Then, to say that the distance between every two codewords is $n/2$ is the exact same thing as to say that every two rows are orthogonal.

This is true because the Hadamard matrix is a unitary matrix. □

Proof. □

In order to decode,

$$x = \begin{bmatrix} C_x(e_1) \\ C_x(e_2) \\ ... \\ C_x(e_n) \end{bmatrix}$$

where $C_x(e_i)$ means the entry corresponding to e_1 in the matrix C_x.

Theorem 4.2. $\forall \delta \in [0, \frac{1}{4}]$, there exists a randomized algorithm that takes as input $C \in F_2^{2^n}$

Proof. Sample $r \in \{0,1\}^(n)$ and compute $\langle x, r\rangle + \langle x, r+u\rangle = \langle x, u\rangle$.

Assume that the bits corresponding to r and $u + r$ are not corrupted. More formally, assume that $C(r) = C_x(r)$ and $C(r + u) = C_x(r + u)$. Notice that these are just bits.

The probability of a single bit being corrupted is at most δ. Therefore, the probability that two bits are corrupted is at most 2δ.

□

The Hadamard code is locally decodable, which is why it's very useful for PCPs.

Codewords of the Hadamard code are precisely the linear functions $F_2^n \to F_2$. There are 2^n such linear functions since every function is fully determined by $\alpha_1, ..., \alpha_n$ such that $x \mapsto \alpha_1 x_1 + ... + \alpha_n x_n$.

Algorithm 1 Algorithm T

```
Input: $C \in F_2^{2^n}$
Output: $C$ is linear.
  $u, v \overset{\$}{\leftarrow} \in F_2^n$
  Output YES if $C(u+v) = C(u) + C(v)$.
```

Hadamard matrices and orthonormal bases.

Proposition 4.3. The functions $(-1)^{C_x}, x \in F_2^n$ form an orthonormal basis for $\mathbb{R}^{F_2^n}$ under the inner product

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{u \in F_2^n} f(u)g(u)$$

$$\langle (-1)^{C_x}, (-1)^{C_y} \rangle = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

5 Locally Decodable Codes (LDC)

5.1 Resources

1. Zeev Dvir IAS Talk on LDCs
2. Zeev Dvir LDC course
3. Private LDCs paper by Rafi and Sahai in 2007
4. Sergey Yekhanin's LDC monograph
5.

Since LDCs require more than regular ECCs, they require longer codewords than classical ECCs.