CODING THEORY

Sayan Kar

August 13, 2025

Contents

Chapter 1

Basics

1.1 Basic Notations and Definitions

Definition 1.1.1: Code

A *code* C is a subset of Σ^n where Σ is an alphabet, where n is the block length of C. We typically use q to denote $|\Sigma|$.

Another way to view the definition of a code to be a map $C : [M] \to \Sigma^n$, where $M = |C|$.

Definition 1.1.2: Dimension of a code

Dimension of a code C, denoted as k, is defined as the following way,

$$k := \log_q |C|.$$

Remark. *Note that,*

1. *For any* $C \subseteq \Sigma^n$, $k \leq n$.

2. k *can be non-integral.*

One way to quantify *Redundancy* in a code is via its rate.

Definition 1.1.3: Rate of a Code

Rate of a code C of block length n and dimension k, denoted as R, is defined as

$$R := \frac{k}{n}.$$

Example 1.1.1. *Define a code* $C \subseteq \{0, 1\}^5$ *that maps a binary string* (x_1, x_2, x_3, x_4) *to* $(x_1, \ldots, x_4, x_1 \oplus \ldots \oplus x_4)$.

1.2 Formalizing Error Correction

Definition 1.2.1: Encoding & Decoding Functions

- Let $C \subseteq \Sigma^n$. An equivalent description of the code C is an injective mapping $E : [|C|] \to \Sigma^n$ called the *encoding function*.
- Let $C \subseteq \Sigma^n$ be a code. A mapping $D : \Sigma^n \to [|C|]$ is called a *decoding function*.

We can refer to the following figure 1.2 from now on.

$$\boxed{m} \xrightarrow{E} \boxed{E(m)} \xrightarrow{Noisy\ Channel} \boxed{ch(E(m))} \xrightarrow{D} \boxed{D(ch(E(m))) = m}$$

Figure 1.1: Encoding-Decoding

1.2.1 Quantifying Error

Definition 1.2.2: Hamming Distance

For any $a, b \in \Sigma^n$, *hamming distance* of a, b is defined as

$$\Delta(a, b) := \#\{i \in [n] : a_i \neq b_i\}$$

where $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$. We also define *relative hamming distance* as

$$\delta(a, b) := \frac{1}{n}\Delta(a, b).$$

Remark. *We can verify easily that this Δ is actually a distance on Σ^n.*

Definition 1.2.3: Hamming Weight

We define *hamming weight* of an element $v \in \Sigma^n$ as, $wt(c) := \Delta(0, v)$.

Refering to the above figure 1.2, we define the error for that transmitted codeword to be $\Delta(c, y)$.

1.3 Performance of an Error Correcting Code

Definition 1.3.1: t-Error Channel & t-Error Correcting Code

- (t-Error Channel) An n symbol *t-Error Channel* is a relation $ch : \Sigma^n \to \Sigma$ such that $\forall c \in \Sigma^n$,
$$\Delta(c, ch(c)) \leq t.$$
- (t-Error Correcting Code) Let $C \in \Sigma^n$ is a *t-Error Correcting code* if $\forall$ t-Error Channel ch and $\forall m \in [|C|]$,
$$D(ch(E(m))) = m.$$

Example 1.3.1. *$C_{3,rep}$ is a 1-error correcting code.*

Example 1.3.2. *$C_\oplus$ is a 0-error correcting code.*

1.3.1 Some weaker Notations

Definition 1.3.2: t-Error Erasure Channel & t-Error Detecting Code

- (t-Erasure Channel) A *t-Erasure Channel* is a mapping $ch : \Sigma^n \longrightarrow (\Sigma \cup \{?\})^n$, where $? \notin \Sigma$, such that $\forall a \in \Sigma^n$,
$$\Delta(a, ch(a)) \leq t.$$
and for all $i \in [n]$ such that $a_i \neq ch(a)_i$, we would have $ch(a)_i = ?$.
- (t-Error Detection Code) A code $C \subseteq \Sigma^n$ is an *t-Error Detecting code* if there exists a detection procedure D such that $m \in [|C|]$ & $\forall$ t-Error Channel,
$$D(ch(E(m))) = \mathbb{1}_{\{ch(E(m))=E(m)\}}.$$

Remark. *Similarly we can also define a t-Erasure code $C \subseteq \Sigma^n$ if $\forall m \in [|C|]$ & t-Erasure Channel ch,*
$$D(ch(E(m))) = E(m) \cong m.$$

Example 1.3.3.

1.3.2 Error Correction Capability of $C_\oplus$ and $C_{3,rep}$

1.4 Distance of a Code

A parameter for quantifying error correction capability of a code is the distance of that code.

Definition 1.4.1: Distance of a Code

For a code $C \in \Sigma^n$, we define its *distance* as the following
$$d(C) := \min_{\substack{c_1 \neq c_2 \\ c_1, c_2 \in C}} \Delta(c_1, c_2).$$

Example 1.4.1. $d(C_\oplus) = 2$ *and* $d(C_{3,rep}) = 3$.

Proposition 1.4.1

Given a code C, the followig are equivalent.

1. C has minimum distance $d \geq 2$.
2. If d is odd, then C can correct upto $\frac{d-1}{2}$ many errors.
3. C can detect $d - 1$ many errors.
4. C can correct $d - 1$ many erasures.

Proof. □