Coding Theory
Lecture 1

Lecture by Dr. Elette Boyle
Typeset by Steven Karas

2017-10-26
Last edited 18:24:06 2017-10-26

Disclaimer These lecture notes are based on the lecture for the course Coding Theory, taught by Dr. Elette Boyle at IDC Herzliyah in the fall semester of 2017/2018. Sections may be based on the lecture slides written by Dr. Elette Boyle.

1 Admin

The textbook for the course is Essential Coding Theory by Guruswami, et al.

Grades 20% based on weekly homework. 80% based on final exam.

Homework will be posted every Thursday evening/Friday morning, and due by 15:00 the next week as PDF.

Lowest homework score will drop, but must be submitted to be eligible for the final exam.

Agenda - Course

- What is coding theory / error correction codes
- Linear codes
- Upper and lower bounds
- List decoding
- Reed Solomon codes and decoding algos
- Expander (LDPC) codes and decoding algos
- Applications of coding to other areas of (theoretical) Computer Science

Agenda - Today

- What are error-correcting codes?
- Definitions - code / block length / dimension / rate / distance
- Error detection / correction
- Hamming code
- Hamming bound

2 What are error correcting codes

A method for adding redundancy to data so that if partial information is lost or corrupted, the original data can still be recovered.

Given a message m, we will add some redundancy using $C(m)$ such that $D(C(m) + e)$ where e is the noise added by the communication channel gives us m.

Example applications

- Spoken language - our brains organically perform this.
- Internet traffic - TCP checksums and request retransmission.
- TV/Cellphones/phones.
- Deep space transmission - extreme high latency communication.
- ID/credit cards numbers - Luhn check.
- CD/DVD - scratches.
- Flash media - charge failures.
- RAID - parity.
- Cloud storage - server failures.

It's important to recall that different applications will have different classes of errors, which may be addressed optimally by different code schemes.

Our goal is to correct the most errors while introducing minimal overhead in redundancy.

3 Formal definitions

3.1 Code

A code of block length n over an alphabet Σ is just a subset $C \subseteq \Sigma^n$. We mark $q = |\Sigma|$

Example: $\Sigma = \{0, 1\}$ would be bits whereas $\Sigma = \{0, 1\}^8$ would be bytes.

As a special case, if $\Sigma = \{0, 1\}$, we call it a binary code.

An alternative view of a code is as a mapping $C : [M] \to \Sigma^n$ if $|C| = M$.

3.2 Dimension

The dimension of a code $C \subseteq \Sigma^n$ is $k = \log_q |C|$ where $q = |\Sigma|$.

3.3 Rate

The rate of a code with dimension k and block length n is $R = \frac{k}{n}$.

We use the rate as a measure of the quality of the code.

3.4 Error Correction/Detection

Given a message m, we will add some redundancy using $C(m)$ such that $D(C(m) + e)$ where e is the noise added by the communication channel hopefully maps back to m.

3.4.1 Encoding

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

3.4.2 Decoding

Let $C \subseteq \Sigma^n$ be a code. A mapping $D : \Sigma^n \to [|C|]$ is called the decoding function for C.

3.4.3 Error Correction

Let $C \subseteq \Sigma^n$ and $t \geq 1 \in \mathbb{N}$. C is said to be t-error-correcting if there exists some decoding function D such that $\forall m \in [|C|]$ and for every error pattern $e \in \Sigma^n$ with at most t errors a decoding function $D(c(m) + e) = m$.

3.4.4 Error Detection

Let $C \subseteq \Sigma^n$ and $t \geq 1 \in \mathbb{N}$. C is said to be t-error-detecting if there exists some detection function D_{detect} such that $\forall m \in [|C|]$ and for every error pattern $e \in \Sigma^n$ with at most t errors a detection function $D_{detect}(c(m)) = \text{accept}$, and $D_{detect}(c(m) + e) = \text{reject}$.[1]

Note: t-error-correcting implies t-error-detecting

3.5 Channel noise/Errors

When talking about errors we typically think of Σ as having some additive structure with a 0 element. We express error to a code word $c \in \Sigma^n$ as a vector $e \in \Sigma^n$, where the received (noisy) word is $y = c + e$. We say that e has t errors if the number of non-zero members of e is t.

There are many ways to model channel noise.

Adversarial noise Assuming any worst case error patterns, as long as the number of symbols in the error is bounded.

Stochastic noise Assuming the expected case. Binary symmetric channel of probability $p : 0 \leq p \leq 1$ called BSC_p; each bit is flipped with probability p.

There are lots of different extensions of these. We will focus on adversarial noise.

3.5.1 Erasure

An error where the receiver knows that an error was in a given position.

3.6 Distance

First, we define a useful distance measure between vectors: the number of positions where elements differ.

Let $u, v \in \Sigma^n$. The Hamming distance between u and v, $\Delta(u, v)$ is defined to be the number of positions where u and v differ.

Example: Binary codes Let $\Sigma = \{0, 1\}$, $u = (1, 0, 0, 1)$ and $v = (0, 1, 0, 1)$. $\Delta(u, v) = 2$.

Let $\Sigma = \{0, 1\}^2$, $u = (10, 01)$ and $v = (01, 01)$. $\Delta(u, v) = 1$. Note that given $v' = (11, 01)$, $\Delta(u, v') = 1$

In the HW, we will prove that the hamming distance $\Delta(u, v)$ is a valid metric which satisfies the following properties:

- $\Delta(u, v) \geq 0$
- $\Delta(u, v) = 0$ iff $u = v$
- $\Delta(u, v) = \Delta(v, u)$
- $\Delta(u, w) \leq \Delta(u, v) + \Delta(v, w)$, otherwise known as the triangle inequality.

3.7 Minimum Distance

Let $C \subseteq \Sigma^n$ be a code. The minimum distance of C is $d = \min_{c \neq c' \in C} \Delta(c, c')$

4 Equivalence of distance properties

The following statements are equivalent for any code C:

1. C has distance $d \geq 2$.
2. C can correct $(d-1)/2$ errors if d is odd, or $d/2 - 1$ errors if d is even.
3. C can detect $(d-1)$ errors.
4. C can correct $(d-1)$ erasures.

[1] This definition is different from the one in the text book. The textbook definition is incorrect, as it ignores cases where the error mutates the message into another valid code word

4.1 Intuition

Disregarding complexity, we consider the entire space of fully valid codewords $C \subseteq \Sigma^n$.

Error detection is finding all the invalid code words within the hypercircle with radius $d-1$ around any given valid codeword.

Error correction is the voronoi cell formed by the expanding hypercircles from the valid codewords.

Erasure correction is ???

4.2 Formal Proof

We will show that $(1) \Rightarrow (2), (3), (4)$ and $(\neg 1) \Rightarrow (\neg 2)(\neg 3)(\neg 4)$.

As a tool, we will use a decoding function Maximum Likelihood Decoding (MLD) for C, which given a received word $y \in \Sigma^n$ will output the codeword $c \in C \subseteq \Sigma^n$ that is closest to y by Hamming distance.

$$D_{MLD}(y) = \min_{c \in C} \Delta(c, y)$$

This can be implemented naively, by enumerating over all codewords (recall we ignore efficiency) and taking the one with minimal Hamming distance.

4.2.1 (1) implies (2)

For simplicity, we only show here the case where d is odd. Let $d = 2t+1$ for some t.

Our goal is to show that C is t-error-correcting. That is, there exists some decoding function D such that for all error patterns of $\leq t$ errors, D correctly recovers the original codeword.

We claim that D_{MLD} achieves this. Suppose for contradiction that it doesn't. There exists some $c \in C$ and $e \in \Sigma^n$ of $\leq t$ errors such that $D_{MLD}(c+e) = c' \neq c$. Therefore by the definition of MLD, $\Delta(c+e, c') \leq \Delta(c+e, c)$.

By the triangle inequality, $\Delta(c, c') \leq \Delta(c, c+e) + \Delta(c+e, c')$. Note that $\Delta(c, c+e) \leq t$, which means that $\Delta(c, c') \leq 2t < d$.

However, this contradicts the definition of d.

4.3 (1) implies (3)

Suppose C has distance d.

We demonstrate a $d-1$-error-detecting algorithm:

$$D_{detect}(y) = \begin{cases} \text{Accept} & \text{if } y \in C \\ \text{Reject} & \text{if } y \notin C \end{cases}$$

For example implemented naively by enumeration.

We claim that for all $c \in C$ and $e \in \Sigma^n$ of $t \leq (d-1)$ errors, $D_{detect}(c+e) = \text{Accept}$ iff $\bar{e} = \bar{0}$. Suppose the contradiction, that there exists some codeword $c \in C$ and $e \in \Sigma^n$ with $0 < t \leq (d-1)$ errors such that $c + e = c' \in C$. Therefore, the distance $\Delta(c, c') \leq t \leq (d-1)$. This contradicts the definition of d.

4.4 (1) implies (4)

Suppose C has distance d.

Consider the erasure decoder $D_{erasure} : (\Sigma \cup \{?\})^n \to C$. This decoder outputs the codeword which agrees with its received input $y \in (\Sigma \cup \{?\})^n$ on all non-? symbols.

To prove this, there exists a unique codeword. In other words, there does not exist a codeword $c' \neq c$ such that removing $\leq (d-1)$ symbols gives equality.

4.5 Not (1) implies not (2)

Suppose C has distance $\leq d-1$.

We show there does not exist a unique decoding up to $(d-1)/2$.

Let $c \neq c' \in C$ such that $\Delta(c, c') \leq d-1$. There exists $y \in \Sigma^n$ halfway between c and c' by Hamming distance ($\Delta(c, y) = \Delta(y, c') = \frac{d-1}{2}$). Therefore, there does not exist a decoder which can distinguish between y as a noisy version of c versus that of c'.

The remaining cases are left as an exercise to the reader.

5 Example Codes

5.1 Parity code

A binary code with $2^4 = 16$ code words:

$$C_\oplus : \{0,1\}^4 \to \{0,1\}^5$$
$$C_\oplus : (x_1, x_2, x_3, x_4) = (x_1, x_2, x_3, x_4, x_1 \oplus x_2 \oplus x_3 \oplus x_4)$$

For this code, $q = 2$, $n = 5$, $k = 4$, and $R = \frac{4}{5}$.

Correction Is $C_\oplus$ 1-error-correcting? No.

$$m = (0,0,0,0) \quad e = (0,0,0,0,1)$$
$$m' = (1,0,0,0) \quad e' = (1,0,0,0,0)$$
$$C(m) + e = C(m') + e' = (0,0,0,0,1)$$

Detection Is $C_\oplus$ 1-error-detecting? Yes. Any single bit flip must change the computed parity bit $C(y[0:4]) \neq C(m) + e$

Is $C_\oplus$ 2-error-detecting? No. In fact, any two errors are sufficient.

$$m = (0,0,0,0) \quad e = (0,0,0,0,0)$$
$$m' = (1,0,0,0) \quad e' = (1,0,0,0,1)$$
$$C(m) + e = C(m') + e' = (0,0,0,0,0)$$

Distance The distance of this code is 2. $C(0000) = 00000$ and $C(1000) = 10001$, so $dist \leq 2$. $dist \geq 2$ because for any valid codeword $(y_1, \ldots, y_5)$, $\oplus y_i = 0$.

5.2 3x Repitition code

A binary code with 2 code words.

$$C_{3,rep} : \{0,1\}^4 \to \{0,1\}^{12}$$
$$C_{3,rep}(x_1, x_2, x_3, x_4) = (x_1, x_1, x_1, x_2, x_2, x_2, x_3, x_3, x_3, x_4, x_4, x_4)$$

For this code, $q = 2$, $n = 12$, $k = 4$, and $R = \frac{1}{3}$.

Correction Is $C_{3,rep}$ 1-error-correcting? Yes. Decoding using majority voting.

Is $C_{3,rep}$ 2-error-correcting? No. Proof by demonstrating two valid messages and two error vectors that map them to the same received message.

$$m = (0,0,0,0) \quad e = (110,000,000,000)$$
$$m' = (1,0,0,0) \quad e = (001,000,000,000)$$
$$C(m) + e = C(m') + e' = (110,000,000,000)$$

Because these encoded messages with errors are not unique, we have shown there cannot exist a decoding function that can decode these messages.

Detection Is $C_{3,rep}$ 1-error-correcting? Yes.

Is $C_{3,rep}$ 2-error-correcting? Yes. No 2 errors can push a group of 3 into a dictinct valid group of 3.

Is $C_{3,rep}$ 3-error-correcting? No.

$$m = (0,0,0,0) \quad e = (111,000,000,000)$$
$$m' = (1,0,0,0) \quad e = (000,000,000,000)$$
$$C(m) + e = C(m') + e' = (111,000,000,000)$$

Minimum distance The distance of this code is 3. Any two valid codewords must differ by at least 3 symbols, and there exist codewords in distance 3.

Erasure correcting As per our proposition, this code is 2-erasure-correcting.

5.3 Hamming Code

This binary code will be dealt with in the homework.

$$C_H : \{0,1\}^4 \to \{0,1\}^7$$

$$C_H(x_1, x_2, x_3, x_4) = (x_1, x_2, x_3, x_4, x_1 \oplus x_2 \oplus x_4, x_1 \oplus x_3 \oplus x_4, x_2 \oplus x_3 \oplus x_4)$$

q = 2
n = 7
k = 4
R = 4/7
d = show in HW.