CS 59000 CTT Current Topics in Theoretical CS Oct 2nd, 2012

Lecture 13

Lecturer: Elena Grigorescu *Scribe: Nader Alawadi*

1 Introduction

We have so far mostly studied combinatorial objects (such as graphs) and very fast algorithms for certain tasks associated to these objects.

In the following few lectures we'll focus on other combinatorial/algebraic objects, namely error-correcting codes, and we'll show a couple of applications of these objects across theoretical CS.

We will study parameters of interest as well as algorithmic questions related to efficient (and super-efficient) decoding, testing, and correction of codes.

Error-correcting codes are widely used in practice. To ensure the correctness of digital data against the errors that occur during transmissions through a communication channel, error-detecting/correcting codes are needed. Errors could be introduced by various types of random noise, distortion of signals and interference. In order for the receiver to detect and/or eliminate such errors, the sender needs to add some extra information to the messages they want to send. As we will see, many different types of codes can be constructed, and we will see that different codes could differ in the amount of error they can detect or correct.

In their seminal papers Shannon and Hamming invented the areas of information theory and coding theory in the early 50's. They proposed two models of noise, respectively: stochastic noise (errors are random) vs. adversarial noise (any pattern of errors is possible). Our treatment in what follows focuses on the Hamming model where the errors could be worst-case.

The general setup is the following: Alice want to sent messages to Bob. Her messages are k bit strings $m = (m_0, m_1, \ldots, m_{k-1})$. She will encode her message into a new $n > k$ bit string $E(m) \in \{0,1\}^n$ containing some redundant information and send it over some channel to Bob. Bob receives a distorted message $r = E(m)+e$, where e is the bitvector with 1's at the locations where the errors have occurred. Bob needs to decode r in order to find m and he can do so as long as he can figure out what the error pattern is (i.e. e).

2 Definitions

Definition 1 (Alphabet) *An alphabet Σ is a finite set of symbols.*

Definition 2 (Error-correcting code) *An error-correcting code over an alphabet Σ is a collection of strings (called codewords) whose symbols belong to Σ.*

Definition 3 (Hamming Distance) *The Hamming distance between two words $w_1, w_2 \in \Sigma^n$, denoted $d_H(w_1, w_2)$, is the number of positions in which they differ. We'll often use the related quantity relative Hamming distance $\delta_H(w_1, w_2) = d_H(w_1, w_2)/n$.*

For example between $d_H(1010, 0011) = 2$ and $\delta(1010, 0011) = 1/2$.
We note here that Hamming Distance is a metric because:

1. $d(x, y) = d(y, x)$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) \leq d(x, z) + d(y, x)$

$\forall x, y, z \in \Sigma^n$, where Σ is an alphabet.

Definition 4 (Minimum Hamming Distance) *The minimum Hamming distance of a block code C, denoted $\Delta(C)$, is the minimum distance among all possible pairs of codewords in C.*

That is $\Delta(C) = \min_{c1,c2 \in C} d(c1, c2)$

Definition 5 (Hamming Weight) *The Hamming weight of a codeword c, denoted $wt(c)$, is the number of nonzero symbols in c.*

For example, both the words (1010) and (0011) have Hamming weight 2.

Definition 6 *The minimum Hamming weight of a code C, denoted $wt_m(C)$, is the smallest weight among all the weights of nonzero codewords in C.*
That is $wt_m(C) = \min_{c \in C - \{0\}} wt(c)$

Definition 7 ($(n, k, d)_q$ code) *A $C = (n, k, d)_q$-code is a code with the following parameters:*

- *Its alphabet Σ is of size $|\Sigma| = q$*
- *Has block length n*
- *Has message length k (this is also called the* dimension *of C). In other words $k = \log |C|$. A related parameter is the rate $R = \frac{k}{n}$*
- *Has distance $\Delta(C) = d$ and so, it has relative distance $\delta(C) = \frac{d}{n}$*

Good codes have the characteristics of small amount of redundancy (and so they have high rate) and large distances between codewords. In building good codes one aims at the following optimizations:

- For fixed n, d we want (n, k, d)-codes that will maximize k.
- For fixed n, k we want (n, k, d)-codes that will maximize d.
- For fixed k, d we want (n, k, d)-codes that will minimize n.

Example 1 *A simple example of a code is obtained by adding one parity check bit to the messages. The code $C = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$ is a $(4, 3, 2)_2$- code, consisting of 8 codewords. For each codeword, the first three bits are the information bits and the 4th is a parity bit i.e. it is set to 0 if the number of 1's in the information bits are even, and 1 otherwise.*

Example 2 *A Repetition code is a code where a message is encoded by repeating it a few times. If the message space is say* $\{0,1\}^3$ *and* uvw *is encoded to* $(uuuvvvwww)$, $\forall u, v, w \in \{0,1\}$ *then we obtain a* $C = (9,3,3)_2$ *block code. Its rate is* $R = 1/3$ *and distance* 3.

Example 3 *We'll next define the Hamming code* $C = (7,4,3)_2$. *Let*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}_{4\times 7}$$

A message $m \in \{0,1\}^4$ *is encoded by multiplying it by the matrix* G *(the operations are binary operations, i.e addition and multiplication is performed (mod* 2*)):* $(m_1, m_2, m_3, m_4) \xrightarrow{encoded\ to} (m_1, m_2, m_3, m_4) \cdot G \in \{0,1\}^7$. *So the codewords of* C *are* $C = \{c \in \{0,1\}^7 |\ \exists m \in \{0,1\}^4, c = m \cdot G\}$. G *is called the generator matrix of the code.*

A matrix related to the generator matrix is the parity check matrix, denoted H, *where* $G \cdot H = 0$. *For the Hamming code* H *is:*

$$H = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}_{7\times 3}$$

The codewords of C *can be also described at* $C = \{c \in \{0,1\}^7, c \cdot H = 0\}$.

Proposition 8 *For the Hamming code* C *defined above, we have* $\Delta(C) = 3$

Proof One can easily find a codeword of weight 3, which says that $\Delta(C) \leq 3$. We'll now show that $\Delta(C) \geq 3$.
Suppose not, i.e $\exists c_1, c_2 \in C$ *s.t* $d(c_1, c_2) \leq 2$

We have

$$\begin{aligned} c_1 \cdot H &= 0 \\ c_2 \cdot H &= 0 \\ \hline (c_1 - c_2) \cdot H &= 0 \end{aligned}$$

Let $c = c_1 - c_2$, so $d(c_1, c_2) = wt(c)$. We have two cases here:

1. $d(c_1, c_2) = 1$:

$$\begin{aligned} c \cdot H = 0 &\implies \exists\ i \text{ where } 1 \leq i \leq 7 \text{ s.t } c[i] = 1 \text{ and } c[j] = 0 \forall j \neq i \\ &\text{and so } \begin{pmatrix} 0 & \cdots & 0 & \underset{i}{1} & 0 & \cdots & 0 \end{pmatrix} \cdot H = 0 \\ &\implies \text{the } i^{th} \text{ row of } H = (0\ 0\ 0) \\ &\text{Contradiction.} \end{aligned}$$

2. $d(c_1, c_2) = 2$: $c \cdot H = 0 \implies \exists i, j, i \neq j$ *where* $1 \leq i, j, \leq 7$ *s.t* $c[i] = c[j] = 1$ *and* $c[k] = 0 \forall k \neq i, j$

and so $\begin{pmatrix} 0 & \cdots & 0 & \underset{i}{1} & 0 & \cdots & 0 & \underset{j}{1} & 0 & \cdots & 0 \end{pmatrix} \cdot H = 0$

$\implies \exists$ 2 *rows of* H, *namely* h_i *and* h_j, *s.t* $h_i + h_j = (0\ 0\ 0)$

Contradiction.

■

3 Linear codes

The Hamming code defined above is an example of a linear code. We will next define these codes more generally. For linear codes we will often work with codes whose alphabet is a finite *field*. A field is just an algebraic object closed under addition and multiplication. $\mathbb{Z}_2 = \{0, 1(\mod 2)\}$ and $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1(\mod p)\}$ are common examples of finite fields. We give a formal definition next.

Definition 9 (Field) $\mathbb{F}$ *is a field if:*

- $+, *$ *are functions:* $\mathbb{F} \times \mathbb{F} \implies \mathbb{F}$
- $+, *$ *are associative*
- $+, *$ *are commutative*
- $*$ *is distributed over* $+$
- *0 and 1 are identity elements where:*
  - $a + 0 = a$
  - $a * 1 = a$
- $\forall a \in \mathbb{F}, a + (-a) = 0$
- $\forall a \in \mathbb{F}^*, (a^{-1})a = 1, where\ \mathbb{F}^* = \mathbb{F} - \{0\}$

A finite field with q elements is denoted $\mathbb{F}_q$. We note that for the field $\mathbb{F} = \mathbb{F}_q$ to exist, q must be of the form $q = p^s$ for some prime p, and for any such q there is a corresponding finite field $\mathbb{F}_q$.

Definition 10 (Linear Codes) *A block code* $C \subset \mathbb{F}_q^n$ *is a linear code if:*

- $\forall c_1, c_2 \in C, c_1 + c_2 \in C$
- $\forall \alpha \in \mathbb{F}_q^n, c \in C, \alpha \cdot c \in C$

We note here that any $(n, k, d)_q$ code that is linear is written $[n, k, d]_q$

Linear codes admit multiple representations: they could be defined by their generator as well as by their parity check matrices. We'll talk some more about these representations in the next lecture.

We next show a useful fact about linear codes.

Proposition 11 *$wt_m(C) = \Delta(C)$ in a $[n, k, d]_q$ linear code*

Proof

$$\begin{aligned}
\Delta(C) &= \min_{\substack{c_i, c_j \in C \\ i \neq j}} d(c_i, c_j) \text{ [by definition]} \\
&= \min_{\substack{c_i, c_j \in C \\ i \neq j}} d(0, c_i - c_j) \text{ [since C is a linear code, } c_i - c_j \in C] \\
&= \min_{\substack{c \in C \\ c \neq 0}} wt_m(c) \text{ [} where \ c = c_i - c_j]
\end{aligned}$$

■

4 Error detection vs error correction

Proposition 12 *Let e be the maximum errors that can occur during the transmission of messages, then a block code C can detect whether some error occurred if $\Delta(C) \geq e + 1$*

Proof
Suppose $\exists x, y \in C \ s.t \ d(x, y) < e + 1$. If x is sent and $y = x + \hat{e}$ was received (where $\hat{e}$ is the error vector introduced by the transmission channel) then the receiver could erroneously assume that y is sent with no error. Therefore we cannot distinguish whether x or y was being sent. ■

Proposition 13 *Let e be the maximum error that can occur during the transmission of messages. Then a block code C can correct such errors if $\Delta(C) \geq 2e + 1$.*

Proof Similar to the proof of proposition 12, if $\Delta(C) < 2e + 1$ then the receiver will not be able to unambiguosly determine the error pattern if the distance between x and y is $< 2e + 1$ and at most e errors occurred. ■

How can we detect if there were errors in the transmission? It turns out that knowing the parity check matrix of the code could give us a lot of information about this.

Consider the Hamming code $(7, 4, 3)_2$ from above. One can easily now check that this is a linear code $[7, 4, 3]_2$. As we've seen above, since its distance is 3 it can detect at most 2 errors and correct 1.

Suppose a block $r \in \{0, 1\}^7$ is received and that at most two errors can be introduced by the communication channel. If $r \cdot H = 0$ then $r \in C$ (by definition) otherwise r was not the

transmitted codeword. If only 1 error is introduced, we can correct it by identifying its position by the following steps:

$$\begin{aligned} r &= c + e_i,\ c \in C,\ e_i \text{ the error at position i.} \\ r \cdot H &= (c + e_i) \cdot H \\ &= c \cdot H + e_i \cdot H \\ &= 0 + h_i, \quad \text{where } h_i \text{ is the } i^{th} \text{ row of the matrix H} \\ &= h_i. \end{aligned}$$

Therefore the error occurred at the index j of r whose binary representation is h_i (by our choice of the matrix H).