**Information and Coding Theory** **Autumn 2014**

Lecture 11: November 4, 2014

Lecturer: Madhur Tulsiani Scribe: Ridwan Syed

# 1 Finite Field Algebra, Vector Spaces, and Polynomials

## 1.1 Finite Fields

**Definition 1.1** *A field is a set $F$ along with two binary operations $+$, addition, and $\cdot$, multiplication, such that the following "field axioms" hold:*

- *For all $x, y \in F$, $x + y = y + x$ and $x \cdot y = y \cdot x$.*
- *For all $x, y, z \in F$, $(x + y) + z = x + (y + z)$ and $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.*
- *For all $x, y, z \in F$, $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.*
- *There exists a unique element $0 \in F$ such that for all $x \in F$, $x + 0 = x$.*
- *There exists a unique element $1 \in F$ such that for all $x \in F$, $x \cdot 1 = x$.*
- *For all $x \in F$, there exists an element $(-x) \in F$ such that $x + (-x) = 0$.*
- *For all $x \in F$ (excluding $x = 0$), there exists an element $x^{-1} \in F$ such that $x \cdot x^{-1} = 1$.*

*We often write $xy$ or $x(y)$ for $x \cdot y$. If $F$ is finite, we say it is a finite field.*

**Definition 1.2** *Let $p$ be a prime integer. Then we define the finite field of order $p$, $\mathbb{F}_p$ also denoted $\boldsymbol{GF}(p)$ as follows:*

- $\mathbb{F}_p = \{0, 1, ..., p - 1\}$
- *The operations $+, \cdot$ are performed modulo $p$ (which we denote $(mod\ \ p)$) i.e. $x + y$ is the remainder of $(x + y)/p$ and similarly for $x \cdot y$ is the remainder of $(x \cdot y)/p$.*

The above definition can also be generalized to prime powers $q = p^n$ for some prime $p$ and integer $n$, but we will be mostly concerned with fields of prime order. We will frequently use the fact that elements of $\mathbb{F}_p$ have multiplicative inverses, and it is useful to verify this fact.

**Proposition 1.3** *For all non-zero $a \in \mathbb{F}_p$, there exists $b \in \mathbb{F}_p$ such that $ab = 1\ (mod\ p)$.*

**Proof:** Consider the multiples of $a$ in $\mathbb{F}_p$: $a, 2a, ..., (p-1)a \pmod{p}$. There are $p - 1$ of these multiples, and each of these is distinct. Moreover, none of these products is 0. Hence, exactly one of these products is equal to 1, and thus there exists $b$ such that $ab = 1 \pmod{p}$. ■

**Exercise 1.4 (Fermat's Little Theorem)** *Let $p$ be prime. Prove that for all $a \in \mathbb{F}_p$, $a^p = a\ (mod\ \ p)$.*

## 1.2 Vector Spaces

**Definition 1.5** *A vector space $V$ over field $F$, is a a collection of tuples from $\mathbb{F}_p^n$. For $v, w \in V$, we define addition coordinate-wise i.e. $v + w = (v_1 + w_1, v_2 + w_2, ..., v_n + w_n)$. For $v, w \in V$, we define their inner product $\cdot$, as $v \cdot w = \sum_{i=1}^{n} v_i w_i \ (mod\ p)$.*

Note that in this case, $V$ is not a Hilbert Space. In particular, there is no good notion of angles between vectors in this space.

## 1.3 Univariate Polynomials

**Definition 1.6** *A polynomial $P$, is an expression of the form,*

$$P(x) = c_0 + c_1 x + c_2 x^2 + ... + c_{p-1} x^{p-1}$$

*where the coefficients $c_0$, ..., $c_{p-1}$ are constants in $\mathbb{F}_p$, and $x$ is a variable in $\mathbb{F}_p$. We denote by $\mathbb{F}_p[x]$ the collection of all univariate polynomials over field $\mathbb{F}_p$.*

Notice that by Fermat's Little Theorem, all powers of $x$ greater than $p-1$ collapse to lower powers. For $d \leq p-1$ we say $P$ is of degree $d$ if the largest power of $x$ with non-zero coefficient is $d$. We will make use of the following two facts about polynomials:

**Fact 1.7** *If $P$ is non-zero and a degree $d$ polynoimial in $\mathbb{F}_p[x]$, then there are at most $d$ values of $x \in \mathbb{F}_p$ such that $P(x) = 0$. Such points are called the roots of $P$.*

**Fact 1.8 (Lagrange Interpolation)** *A degree $d$ polynomial is uniquely determined by $d+1$ values. In particular: let $a_1, ..., a_{d+1}$ be distinct points in $\mathbb{F}_p$ and let $b_1, ..., b_{d+1} \in \mathbb{F}_p$ be arbitrary. Then, there exists a unique degree $d$ polynomial $Q \in \mathbb{F}_p[x]$ such that for all $i \in [d+1]$, $Q(a_i) = b_i$.*

First we prove fact 1.7:

**Proof:** We argue by induction on degree.
Base case: Suppose $d = 0$. Then, $P(x) = c_0$ is constant. Since $P$ is non-zero, $P$ has no roots.
Induction: Now, suppose all polynomials of degree at most $d-1$ have at most $d-1$ roots. Let $P$ be of degree $d$, and let $a$ be a root of $P$. We can divide out the polynomial $x - a$ and obtain:

$$P(x) = (x - a)P'(x) + R(x)$$

where $P'$ and $R$ are polynomials, and $R$ has degree less than $x - a$. Thus, $R$ is a constant. Then we have that:

$$P(a) = (a - a)P'(a) + R(a) = R(a)$$

Since $P(a) = 0$ and $R$ is constant, $R$ must be identically 0. Therefore,

$$P(x) = (x - a)P'(x)$$

Since $P$ is of degree $d$, $P'$ is of degree at most $d-1$ and by the inductive hypothesis has at most $d-1$ roots. Hence, by induction $P$ has at most $d$ roots. ■

Now the proof of fact 1.8:

**Proof:** Suppose we are given distinct $a_1, ..., a_{d+1} \in \mathbb{F}_p$ and arbitrary $b_1, ..., b_{d+1} \in \mathbb{F}_p$. First, we prove the existence of a polynomial $Q(x)$, such that for all $i \in [d+1]$, $Q(a_i) = b_i$. Define $Q(x)$ as :

$$Q(x) = \sum_{i=1}^{d+1} b_i \cdot \left( \frac{\Pi_{j \neq i}(x - a_j)}{\Pi_{j \neq i}(a_i - a_j)} \right)$$

Notice that when $x = a_i$, $\left( \frac{\Pi_{j \neq i}(x - a_j)}{\Pi_{j \neq i}(a_i - a_j)} \right) = 1$, and otherwise this ratio is 0. Thus for all $i \in [d+1]$, $Q(a_i) = b_i$.
Now we show uniqueness. Suppose for contradiction there exists $Q'$ s.t. $Q \neq Q'$ and for all $i \in [d+1]$, $Q(a_i) = Q'(a_i) = b_i$. Then, $P(x) = Q(x) - Q'(x)$ is of degree at most $d$, but $P$ has $d+1$ roots : $a_1, ..., a_{d+1}$. By contradiction, $Q(x)$ is unique. ■

# 2 Error Correcting Codes

Suppose Alice wants to send Bob a message over a noisy channel, where some of the bits in Alice's message may become corrupted. Let's assume that Alice's message, is a sequence of elements of $\mathbb{F}_p$ for some prime $p$. There are two models for how the message may become corrupted.

## 2.1 Shannon Model

Let $m = m_1 m_2 .... m_k$ be a string where each $m_i \in \mathbb{F}_p$. In the Shannon Model, Alice sends $m$ over the communication channel, and each $m_i$ is corrupted independently (though not necessarily identical distributions for each bit) at random.

**Example 2.1 (Binary Symmetric Channel)** *Suppose the message is a tuple from $\mathbb{F}_2$. Then, for some $\varepsilon \in [0, \frac{1}{2}]$, each bit is flipped independently at random with probability $\varepsilon$, and is transmitted without error with probability $1 - \varepsilon$.*

We will work with a different model, which is more relevant for some of the applicatios in theoretical computer science.

## 2.2 Hamming Model

Suppose Alice sends $k$ symbols $m_1 m_2 .... m_k$ from $\mathbb{F}_p$. In the Hamming Model, up to $t$ of these symbols are corrupted (but not lost) in some arbitrary manner. There is no way to know which of the symbols have been corrupted, and which symbols were transmitted correctly. We wish to design encoding and decoding schemes to recover up to $t$ errors for some fixed $t$.

**Definition 2.2** *A code is a function $C : \mathbb{F}_p^k \to \mathbb{F}_p^n$, where $n \geq k$. We call $k$ the message length and $n$ the block length. We call $R(C) = \frac{k}{n}$ the rate of $C$.*

Our goal is to introduce some redundancies to our message such that we can uniquely recover the correct $k$ symbols given that up to $t$ bits from the $n$ bit message may be corrupted.

**Example 2.3** *Define* $C : \mathbb{F}_2^4 \to \mathbb{F}_2^{12}$ *as*

$$C(x_1x_2x_3x_4) = (x_1x_1x_1x_2x_2x_2x_3x_3x_3x_4x_4x_4)$$

The rate of this code is $\frac{1}{3}$. In general we'd like to keep this rate close to 1, as this would imply we introduce fewer redundancies to the encoded message. It is easy to see that the above code can handle one error. When Bob receives the string, he simply takes the majority of each block of three symbols and thus recovers the original message. As we shall see, no more than one error can be recovered using this code.

Another way to think about a code $C$ is as a subset of $\mathbb{F}_p^n$:$\mathcal{C} = \{C(x) \ : \ x \in \mathbb{F}_p^n\}$.

**Definition 2.4** *Let* $\delta(x, y)$ *denote the Hamming distance of two strings* $x, y$*. We define the distance of a code* $\Delta(C)$ *as*

$$\Delta(C) = min_{x \neq y}\delta(x, y)$$

*where* $x, y \in \mathcal{C}$*.*

**Remark 2.5** *The hamming distance* $\delta$ *is a metric on strings of a fixed length, and in particular satisfies the triangle inequality.*

**Theorem 2.6** *A code* $C : \mathbb{F}_p^k \to \mathbb{F}_p^n$ *can correct* $t$ *errors if and only if* $\Delta(C) \geq 2t + 1$*.*

Note that for the code in Example 2.3, $\Delta(C) = 3$. This implies that the code from Example 2.3 can handle no more than one error. We now prove the bound:

**Proof:** First, we prove the reverse direction. If $\Delta(C) \geq 2t+1$, then $\frac{\Delta(C)}{2} > t$. For each codeword $x \in \mathcal{C}$ let $H(x, r) = \{y \in \mathbb{F}_p^n \ : \ \delta(x, y) < r\}$. In particular consider the case when we let $r = \frac{\Delta(C)}{2}$. Notice that for two distinct $x, x' \in \mathcal{C}$, $H(x, r) \cap H(x', r) = \emptyset$, since otherwise, $\delta(x, x') < \Delta(C)$. Now, suppose a codeword is corrupted in $t$ bits to the string $y$. Since $t < \frac{\Delta(C)}{2}$, we have that the corrupted message $y$ is contained in precisely one $H(x, r)$. Thus, we can simply check which codeword's $H(x, r)$ $y$ is contained in and obtain the original codeword $x$.
Now, we prove the forward direction. Suppose the codeword $x \in \mathcal{C}$ is corrupted in $t$ bits to $y$. We know that $\delta(x, y) \leq t$. Since we can correct up to $t$ errors, we know that we can find $w \in \mathcal{C}$ such that $w = argmin_{z \in \mathcal{C}}(\delta(y, z))$. Now, this implies that $\delta(w, y) > \delta(x, y) \Rightarrow \delta(w, x) - \delta(x, y) > \delta(x, y) \Rightarrow \delta(w, x) > 2\delta(x, y) \Rightarrow \delta(w, x) \geq 2\delta(x, y) + 1 \Rightarrow \Delta(C) \geq 2t + 1$. ■