LINEAR ALGEBRA NOTES

Laurence Barker, Department of Mathematics, Bilkent University

version: 22 February 2024

These notes are associated with MATH 224 Linear Algebra 2. The prerequisites are a first course in linear algebra together with some experience in abstract mathematical reasoning founded on set theoretic definitions.

Anyone arriving here from my other current course, MATH 220 Linear Algebra, is very welcome. However, the material is not oriented towards the needs of the 220 course. We shall be approaching linear algebra as a mathematical theory. The focus is on properties of mathematical objects, not methods of calculation.

Part 1: Vector Spaces

This version of the notes is in incomplete form. I intend to add some exercises. And some parts, being in draft, will inevitably have many slips.

1: Rings and fields

In order to give a general definition of the notion of a vector space, we shall be needing the notion of a field. But we shall start with the more general notion of a ring, which will be needed later, for instance, when we come to study Jordan normal form.

We define a **ring** to be a set R equipped with two functions $R \times R \to R$, one of the called **addition**, written as $(a, b) \mapsto a + b$, the other called **multiplication**, written as $(a, b) \mapsto a.b$, the two functions being required to satisfy the following seven conditions. We often write ab instead of $a.b$.

Additive associativity condition: For all $a, b, c \in R$, we have $a + (b + c) = (a + b) + c$. So we can write $a + b + c$ unambiguously.

Additive commutativity condition: For all $a, b \in R$, we have $a + b = b + a$.

Zero condition: There exists an element $0_R \in R$ such that, for all $a \in R$, we have $0_R + a = a$. It is easy to check that 0_R is unique. Indeed, if $0'_R$ is another zero element, then $0'_R = 0^R + 0'_R = 0'_R + 0_R = 0_R$. We call 0_R the **zero element** of R. When no ambiguity can arise, we write 0 instead of 0_R.

Negation condition: For all $a \in R$, there exists an element $b \in R$ such that $a + b = 0$. It is easy to check that, given a, then such b is unique. We call b the **negative** of a and we write it as $-a = b$.

Multiplicative associativity condition: For all $a, b, c \in R$, we have $a(bc) = (ab)c$. So we can write abc unambiguously.

Unity Condition: There exists an element $1_R \in R$ such that, for all $a \in R$, we have $1_R a = a = a 1_R$. It is easy to check that 1_R is unique. We call 1_R the **unity element** of R. We often write 1 instead of 1_R.

Distributivity condition: For all $a, b, c \in R$, we have $a(b+c) = ab+ac$ and $(a+b)c = ac = bc$.

In a more formal language, a ring is sometimes defined as a triple $(R, +, .)$. That notation lists the three things that are needed in order to specify the ring: the set R, the addition operation and the multiplication operation. In practice, though, it is conventional to abuse notation, employing the same symbol R to denote either the underlying set or else the ring. The ambiguity is resolved through context.

A ring R is called a **commutative ring** provided the following further condition holds:

Multiplicative commutativity condition: For all $a, b \in R$, we have $ab = ba$.

We define a **field** to be a commutative ring F satisfying the following further condition.

Inversion condition: We have $1_F \neq 0_F$ and, for all non-zero $a \in F$, there exists an element $b \in F$ such that $ab = 1_F = ba$. It is easily checked that, given a, then b is unique. We call b the **inverse** of a and we write $a^{-1} = b$.

Some examples of fields are the field of rational numbers $\mathbb{Q}$, the field of real numbers $\mathbb{R}$, the field of complex numbers $\mathbb{C}$. Some examples of finite fields are given in an exercise at the end of this section.

Given rings R and S, we define a **homomorphism** $R \to S$ to be a function $\theta : R \to S$ such that the following three cinditions hold:

- we have $\theta(a+b) = \theta(a) + \theta(b)$ for all $a, b \in R$,
- we have $\theta(ab) = \theta(a)\theta(b)$ for all $a, b \in R$,

we have $\theta(1_R) = 1_S$.

When θ is bijective, we call θ an **isomorphism**. When there exists an isomorphism $R \to S$, we say that R and S are **isomorphic** and we write $R \cong S$.

Consider rings R, S, T. Observe that the identity function id_R on R is an isomorphism $R \to R$. Given an isomorphism $\theta : R \to S$, then $\theta^{-1} : S \to R$ is an isomorphism and, given an isomorphism $\phi : S \to T$, then $\phi \circ \theta : R \to T$ is an isomorphism. Thus, isomorphism has the following features of an equivalence relation: reflexivity, $R \cong R$; symmetry, if $R \cong S$ then $S \cong R$; transitivity, if $R \cong S$ and $S \cong T$, then $R \cong T$. Strictly speaking, equivalence relations are defined only on sets, whereas the class of rings is a proper class. In jargon acknowledging that quibble, ring isomorphism is said to be a formal equivalence relation.

Consider a ring R and a subset $A \subseteq R$. We call A a **subring** of R, writing $A \leq R$, provided $1_R \in A$ and the addition and multiplication operations on R restrict to functions $A \times A \to A$. In that case, A becomes a ring whose addition and multiplication operations are those two restricted functions.

We define the **centre** of a ring R, denoted $Z(R)$, to be the set of elements $z \in R$ such that $az = za$ for all $a \in R$. Plainly, $Z(R)$ is a subring of R.

♣ ♡ ♢ ♠

Exercise 1.1.A: Let R be a ring. Show that, for all $a \in R$, we have $0_R a = 0_R = a 0_R$.

Exercise 1.1.B: Let $\theta : R \to S$ be a ring homomorphism. Show that $\theta(0_R) = 0_S$.

Exercise 1.1.C: Let E and F be fields. Show that any homomomorphism $E \to F$ is injective.

Exercise 1.1.D: Given a ring R and a positive integer n, we write $\mathrm{Mat}_n(R)$ to denote the ring of $n \times n$ matrices with entries in R. Show that $Z(\mathrm{Mat}_n(R)) \cong Z(R)$.

Exercise 1.1.E: The ring of quaternions $\mathbb{H}$ is defined to be the ring such that $Z(\mathbb{H}) = \mathbb{R}$ and $\mathbb{H}$ has an $\mathbb{R}$-basis $\{1, i, j, k\}$ where

$$i^2 = j^2 = k^2 = ijk = -1 .$$

Show that there is an injective ring homomorphism $\mathbb{H} \to \mathrm{Mat}_2(\mathbb{C})$ given by

$$t + ix + jy + kz \mapsto t + xI + yJ + zK$$

for $t, x, y, z \in \mathbb{R}$, where $I = \begin{bmatrix} -i & 0 \\ 0 & i \end{bmatrix}$ and $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and $K = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$. Show that

$$\det(t + xI + yJ + zK) = t^2 + x^2 + y^2 + z^2 .$$

Hence or otherwise, show that every nonzero element $q = t + ix + jy + kz$ has an inverse and give a formula for q^{-1}.

Exercise 1.1.F For a positive integer n, we write $\mathbb{Z}/n$ to denote the ring of modulo n residue classes of the integers. Show that $\mathbb{Z}/n$ is a field if and only if n is prime.

2: Vector spaces

Let F be a field. We define a **vector space over** F, sometimes called an F**-vector space** to be a set V equipped with functions $V \times V \to V$ and $F \times V \to V$, the first called **addition** and written as $(x, y) \mapsto x+y$, the second called **scalar multiplication** and written as $(\lambda, x) \mapsto \lambda x$, the two functions satisfying the following seven conditions.

Zero Condition: There is a vector $\underline{0} \in V$, called the **zero vector**, such that, for all $x \in V$, we have $x + \underline{0} = x$.

Negation Condition: For all $x \in V$, there exists $-x \in V$ such that $x + (-x) = \underline{0}$.

Commutativity Condition: For all $x, y \in V$, we have $x + y = y + x$.

Additive Associativity Condition: For all $x, y, z \in V$, we have $x + (y + z) = (x + y) + z$. Hence, we can write $x + y + z$ unambiguously.

Multiplicative Associativity Condition: For all $\lambda, \mu \in F$ and $x \in V$, we have $\lambda(\mu x) = (\lambda\mu)x$.

Distributivity Condition: For all $\lambda, \mu \in F$ and $x, y \in V$, we have $(\lambda + \mu)x = \lambda x + \mu x$ and $\lambda(x + y) = \lambda x + \lambda y$.

Identity Condition: For all $x \in V$, we have $1x = x$.

Given F-vector spaces U and V, we define an F**-linear map** $U \to V$, to be a function $\alpha : U \to V$ such that the following conditions hold:

Preservation of addition: We have $\alpha(x + y) = \alpha(x) + \alpha(y)$ for all $x, y \in U$.

Preservation of scalar multiplication: We have $\alpha(\lambda x) = \lambda\alpha(x)$ for all $\lambda \in F$ and $x \in U$.

Those two conditions can be combined as the condition that

$$\alpha(\lambda x + \mu y) = \lambda\alpha(x) + \mu\alpha(y)$$

for all $\lambda, \mu \in F$ and $x, y \in U$. When F can be understood from the context, an F-linear map is called simply a linear map.

A bijective linear map is called an **isomorphism** When there exists an isomorphism $U \to V$, we say that U and V are **isomorphic** and we write $U \cong V$.

Much as above, isomorphism of F-vector spaces is a formal equivalence relation. Indeed, for F-vector spaces U, V, W, it is easy to see that: $U \cong U$; if $U \cong V$ then $V \cong U$; if $U \cong V$ and $V \cong W$, then $U \cong W$.

$\clubsuit \qquad \heartsuit \qquad \diamondsuit \qquad \spadesuit$

Exercise1. 2.1: Let F be any field. Give an example of an F-vector space V and a subspace U strictly contained in V such that $U \cong V$.

3: Spanning, linear independence, bases, dimension

In this version of the notes, we omit to cover much of the eponymous material of this section. Let us just note a few major points, omiting proofs. The following result is a key to many others.

Lemma 3.1: (Steinitz Exchange Lemma.) *Let V be a vector space. Let r_1, ..., r_m be independent vectors in V and let S be a finite spanning set in V. Then $n \leq |S|$ and the elements of S can be enumerated as $S = \{s_1, ..., s_m\}$ such that, for all $1 \leq k \leq m$, the set $\{r_1, ..., r_k, s_{k+1}, ..., s_m\}$ is a spanning set for V.*

The remaining results in this section are not hard to deduce using theb latest lemma.

Theorem 3.2: *Let V be a vector space, let T be a linearly independent set in V and let S be a finite spanning set in V. Then T is finite and $|T| < |S|$. In particular, V has a finite basis and all the bases of V have the same size.*

When the equivalent conditions in the latest theorem hold, we say that V is **finite-dimensional** and we define the **dimension** of V, denoted $\dim(V)$, to be the size of a basis for V.

Theorem 3.3: *Given a subspace U of a finite-dimensional vector space V, then U is finite-dimensional, any basis for U can be extended to a basis for V and, in particular,* $\dim(U) \leq \dim(V)$*. Furthermore,* $\dim(U) = \dim(V)$ *if and only if $U = V$.*

$\clubsuit \qquad \heartsuit \qquad \diamondsuit \qquad \spadesuit$

Exercise 1.3.1: Let F be a finite field. Show that there exists a prime p and a positive integer n such that $V = p^n$.

Exercise 1.3.2: Let $\mathbb{F}_3$ denote the field with order 3. Let V be a 3-dimensional $\mathbb{F}_3$-vector space. How many subspaces does V have?

4: Coding theory

We shall discuss coding theory over the field $\mathbb{F}_2 = \{0, 1\}$. Similar notions apply, with little change, when $\mathbb{F}_2$ is replaced by another finite field.

Let n be a positive integer. Consider the standard n-dimensional vector space $\mathbb{F}_2^n$ over $\mathbb{F}_2$. We define a **linear code** in $\mathbb{F}_n$ to be a subspace of $\mathbb{F}_2$. Note that, given a liear code C in $\mathbb{F}^n$ and letting $m = \dim_{\mathbb{F}_2}(C)$, then $C \cong \mathbb{F}_2^m$ and, in particular, $|C| = |\mathbb{F}_2^m| = 2^m$.

A **linear coding scheme** over $\mathbb{F}_2$ is a triple (C, e, d) consisting of:

- a code C in a standard vector space $\mathbb{F}^n$,
- an bijective function $e : C \leftarrow \mathbb{F}^m$ called the **encoding function**,
- a function $d : \mathbb{F}^m \rightarrow \mathbb{F}^n$, called the **decoding function**, such that $de = \mathrm{id}_{\mathbb{F}^m}$.

The primary use of coding theory is as follows. We are to transmit a message as a sequence of message words, each *message word* being a binary string with length m. We encode each message word as a binary string of length n, called the corresponding *codeword*. During transmission, the data may be corrupted, and the received binary strong of length n, called the *received word*, may be different from the codeword. We decode the received word, detecting errors if possible, correcting errors if possible.

More precisely, the message word $w \in \mathbb{F}^m$ is encoded as a codeword $e(w) \in \mathbb{F}^n$. The received word $r \in \mathbb{F}^n$ is then decoded as $d(r) \in \mathbb{F}^m$. The condition $de = \mathrm{id}$ ensures that, if $r = e(w)$, then $d(r) = w$. Of course, if r is not a codeword, then there must have been an error of transmission. The main aim of coding theory is to optimize the task of detecting or even correcting small errors of transmission.

We define the **weight** $\mathrm{wt}(x)$ of a binary string x to be the number of nonzero digits of x. Thus, for example $\mathrm{wt}(100010110) = 4$.

We view $\mathbb{F}_2^n$ as a metric space where the distance $d(x, y)$ between two elements $x, y \in \mathbb{F}_2$ is

$$d(x, y) = \mathrm{wt}(x - y) \ .$$

Note that, since $1 + 1 = 0$ in $\mathbb{F}_2$, we have $x - y = x + y$. Thus, writing $x = x_1 ... x_n$ and $y = y_1 ... y_n$ as binary strongs, with each $x_i, y_i \in \mathbb{F}_2$, then

$$d(x, y) = |\{i : x_i \neq y_i\}| \ .$$

For instance, $d(1011010, 1010110) = 2$.

Proposition 4.1: *Let C be a linear code over $\mathbb{F}_2$ and let k be the minimum weight of a nonzero element of C. Then k is the minimum distance between any two distinct elements of C. Furthermore:*

(1) *If the number of errors of transmittion in the received word is less than k, then we can detect any error of transmission.*

(2) *If the number of errors of transmission in the received word is less than $k/2$, then we can correct any error of transmission.*

Proof: Let c be a codeword of minimum weight, and let x and y be codewords such that $d(x, y)$ is minimum. Since $\mathrm{wt}(c) = d(c, 0)$ and 0 is a codeword, we have $\mathrm{wt}(c) \geq d(x, y)$. But $x - y$ is a codeword, so $d(x, y) = \mathrm{wt}(x - y) \geq \mathrm{wt}(c)$.

Part (1) is obvious. For part (2), we note that, to correct a received word r with less than $k/2$ errors of transmission, we can replace r with the unique nearest codeword. □

The crucial features of linear coding scheme are the rate m/n, which one prefers to be small for the sake of efficiency, and the minimum weight of a codeword k, which determines the error-detection and error-correction capabilities. Note that those parameters are determined entirely by the code C, presuming that the decoding function operates in the manner described in the proof of the latest proposition.

♣ ♡ ♢ ♠

We shall examine coding schemes of the following form. The scheme is determined by a matrix $A \in \mathrm{Mat}_{n-m,m}(\mathbb{F}_2)$, we mean, A is an $(n-m) \times m$ matrix over $\mathbb{F}_2$. We define the **generating matrix** G and the **Hamming matrix** H to be

$$G = \begin{bmatrix} I_m \\ A \end{bmatrix} \in \mathrm{Mat}_{n,m}(\mathbb{F}_2) \ , \qquad H = \begin{bmatrix} A & I_{n-m} \end{bmatrix} \in \mathrm{Mat}_{n-m,n}(\mathbb{F}_2) \ .$$

The matrix G determines the encoding function. For a message word $w \in \mathbb{F}^m$, the corresponding codeword is $e(w) = Gw \in \mathbb{F}_n$. The matrix H can be used as a check to see whether a received word $r \in \mathbb{F}_n$ is a codeword. Indeed, we have the following little proposition, whose proof we defer to Exercise 1.4.B

Proposition 4.2: *With the notation above $Hr = 0$ if and only if r is a codeword.*

The matrix H can also be used to evaluate the decoding function, as explained below.

For such coding schemes, let us describe a method for writing out a table for decoding by hand, and then a method for using the table carry out the encoding and decoding. To construct the decoding table, we carry out the following steps:

Step 1: Write out the message words w in the top row of the table.

Step 2: In the next row, beneath each message word w, write out the corresponding codeword Gw.

Step 3: Having completed a row of the decoding table, if not all of the possible received words have yet appeared in the table, choose a received word r of minimal weight such that r has not yet appeared. Write r in the first column of the next row. Complete the row by entering $r + c$ in the column containing codeword c. Repear this step until all 2^n possible received words appear.

Step 4: Add a new column on the right such that, for each received word r, the entry in the rightmost column of the rowe containing r is Hr. We call Hr the **syndrome** of r. All the received words in a row have the same syndrome.

The following is the decoding table for $A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$. We have

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} , \qquad H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} .$$

000	001	010	011	100	101	110	111	syndrome
000000	001101	010011	011110	100110	101011	110101	111000	000
100000	101101	110011	111110	000110	001011	010101	011000	110
010000	011101	000011	001110	110110	111011	100101	101000	011
001000	000101	011011	010110	101110	100011	111101	110000	101
000100	001001	010111	011010	100010	101111	110001	111100	100
000010	001111	010001	011100	100100	101001	110111	111010	010
000001	001100	010010	011111	100111	101010	110100	111001	001
100001	101100	110010	111111	000111	001010	010100	011001	111

Encoding message words is very easy. As we have already said, for a message word w, the encoding of w is $e(w) = Gw$. Thus, for the example above, the sequence of message words

$$100, 101, 000, 111$$

is encoded as

$$100110, 101011, 000000, 111000\ .$$

To decode a received word r, we could just scan the whole table to find r. Then the decoded word $d(r)$ is the message word in the same column as r. An algorithm that is rather faster to implement by computer is to calculate the syndrome Hr, then find the received word in the same row as the syndrome, whereupon the decoded word $d(r)$ is again the message word in the same column as r. Thus, returning again to the example above, the sequence of received words

$$100111, 101010, 010000, 011010$$

has corresponding syndromes

$$001, 001, 011, 111$$

and correponding message words

$$100, 101, 000, 010\ .$$

None of those 4 syndromes are 000, so none of those received words are codewords.

So, had someone wished to communicate the message 100, 101, 000, 111, had they encoded it as 100110, 101011, 00000, 111000, had 5 inversions of digits taken place in the ether, and had we received 100111, 101010, 010000, 011010, then our decoding would be 100, 101, 000, 010, but we would at least know that the data had been corrupted.

Exercise 1.4.A: *Consider the linear coding scheme with generating matrix*

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}\ .$$

(a) *Write down the Hamming matrix H for the coding scheme.*

(b) *Write down a decoding table in the manner described above, including the column of syndromes.*

(c) *Encode the message words* 100, 011, 110, 000*.*

(d) *For the received words* 10010, 01001, 00011, 11111*, write down the syndromes, then write down the decoded words.*

(e) *What is the rate of the code?*

(f) *If a single codeword is transmitted, what is the maximum number of errors of transmission (the maximum number of inversions of binary digits) such that any error can be detected? And what is the maximum number of errors of transmission (the maximum number of inversions of binary digits) such that any error can be corrected?*

Exercise 1.4.A: Prove Proposition 4.2. (Hint: this can be done using the rank-nullity formula, or by a direct combinatorial argument.)

Solutions

Solution 1.4.A: Part (a). We have $H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$.

Part (b). The decoding table:

000	001	010	011	100	101	110	111	syndrome
00000	00101	01011	01110	10010	10111	11001	11100	00
00001	00100	01010	01111	10011	10110	11000	11101	01
00010	00111	01001	01100	10000	10101	11011	11110	10
01000	01101	00011	00110	11010	11111	10001	10100	11

Part (c). Respectively, 100, 011, 110, 000 have encodings 10010, 01110, 11001, 00000.

Part (d). Respectively, 10010, 01001, 00011, 11111 have syndromes 00, 10, 11, 11 and decodings 100, 010, 010, 101.

Part (e). The rate is 3/5.

Part (f). The minimum weight of a nonzero codeword is 2. So up to 1 error of transmission can always be detected, and up to 0 errors of transmission can always be corrected.