# Linear Algebra

My notes from Math 54

Matthew Signorotti
UC Berkeley

Summer 2018

# Contents

# Chapter 1

# Matrices and vectors

**Definition 1** (an $m \times n$ matrix)**.** *A $m \times n$ matrix is a grouping of numbers, typically real numbers, each one of which has an associated row (between* 1 *and* $m$*) and an associated column (between* 1 *and* $n$*).*

An example of a $2 \times 3$ matrix is

$$A = \begin{bmatrix} -0.3 & 6 & 5 \\ -2.3 & 0 & 4 \end{bmatrix}.$$

The top left entry, here $-0.3$, is said to occupy row and column 1, or position $(1, 1)$. The bottom right entry, 4, is said to be in position $(2, 3)$. Matrices are typically denoted as capital letters; for instance, this matrix is denoted by $A$. To refer to the entry in row $i$ and column $j$ of some matrix $A$, one could write $a_{ij}$.

**Definition 2** (an $n$-vector)**.** *An $n$-long vector is an ordering of $n$ numbers. The most common format for a vector is the column vector, which is an $n \times 1$ matrix, with $n$ rows and 1 column. A vector may also be a $1 \times n$ matrix, also known as a row vector.*

The $i$th element of a vector $\mathbf{x}$ is $x_i$. Examples of row and column vectors include the following:

$$\begin{bmatrix} 7 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 7 & -1 & 0 & 2 \end{bmatrix}$$

## 1.1 Defining basic matrix operations and relationships

### 1.1.1 Scalar multiplication, addition, and equality

The operations of scalar multiplication, addition, and equality are defined just as one might expect for matrices and vectors. For scalar multiplication ($cA$), one multiplies each element of the matrix $A$, in any order, by the scalar $c$. For addition (or subtraction) of two matrices or vectors of equal dimensions ($A + B$), one adds (subtracts) the two elements in row $i$ and column $j$, for all $i$ and $j$. Lastly, two matrices or vectors $A$ and $B$ are equal exactly when (1) they are of equal dimension and (2) $a_{ij} = b_{ij}$ for all $i$ and $j$.

Without going into excessive formalism, the following equation is true, given the above definitions and the standard properties of real numbers.

$$2\begin{bmatrix}-0.3 & 5\end{bmatrix} - \begin{bmatrix}2 & 0.1\end{bmatrix} = \begin{bmatrix}2(-0.3) - 2 & 2(5) - 0.1\end{bmatrix} = \begin{bmatrix}-2.6 & 9.9\end{bmatrix}$$

Make sure you fully understand why this equality holds. These three operations are some of the most fundamental of matrix algebra, so it is important that you have them down!

### 1.1.2 Matrix multiplication

**Definition 3** (matrix-matrix multiplication)**.** *Consider an $l \times m$ matrix $A$ and $m \times n$ matrix $B$. (The matrix multiplication $AB$ is only defined when $A$'s number of columns matches $B$'s number of rows.) By definition, $AB$ has dimensions $l \times n$, and the element in row $i$ and column $j$ is*

$$\sum_{k=1}^{m} a_{ik} b_{kj},$$

*if $a_{ij}$ is the element corresponding to row $i$ and column $j$ of $A$.*

The same definition holds when $A$ and/or $B$ is a vector. In fact, matrix-vector multiplication ($A\mathbf{x}$, with $\mathbf{x}$ a column vector) is an incredibly common operation.

Here is an example:

$$\begin{bmatrix}1 & 3\\ -1 & 0\end{bmatrix}\begin{bmatrix}1\\ 2\end{bmatrix} = \begin{bmatrix}1(1) + 3(2)\\ -1(1) + 0(2)\end{bmatrix} = \begin{bmatrix}7\\ -1\end{bmatrix}$$

The definition of matrix-matrix multiplication has multiple equivalent definitions, which at times may allow for easier thinking about a problem. You should think of the above main definition, along with the following two definitions, as one and the same.

**Definition 4** (the column-wise view of matrix-matrix multiplication)**.** *In accordance with the above definition, the matrix operation defined by* $AB = \begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_m\end{bmatrix}\begin{bmatrix}\mathbf{b}_1 & \cdots & \mathbf{b}_n\end{bmatrix}$ *can also be viewed as n linear combinations of the columns* $\{\mathbf{a}_i\}$ *according to the weights given in each column* $\mathbf{b}_i$*:*

$$AB = \begin{bmatrix}A\mathbf{b}_1 & \cdots & A\mathbf{b}_n\end{bmatrix} = \begin{bmatrix}\sum_{i=1}^m \mathbf{a}_i b_{i1} & \cdots & \sum_{i=1}^m \mathbf{a}_i b_{in}\end{bmatrix}$$

When $B$ is just an $m \times 1$ column vector $\mathbf{b}$, this column-wise view simplifies nicely:

$$A\mathbf{b} = b_1\mathbf{a_1} + b_2\mathbf{a_2} + \cdots + b_n\mathbf{a_n}.$$

**Definition 5** (the row-wise view of matrix-matrix multiplication)**.** *An equivalent row-wise view of matrix multiplication also exists. For this definition, let* $\hat{\mathbf{a}}_i$ *or* $\hat{\mathbf{b}}_i$ *be the ith* ***row*** *of A or B, unlike* $\mathbf{a}_i$ *or* $\mathbf{b}_i$*, which are the* ***columns*** *of these matrices. Then*

$$AB = \begin{bmatrix}\hat{\mathbf{a}}_1 B \\ \vdots \\ \hat{\mathbf{a}}_l B\end{bmatrix} = \begin{bmatrix}\sum_{i=1}^m a_{1i}\hat{\mathbf{b}}_i \\ \vdots \\ \sum_{i=1}^m a_{li}\hat{\mathbf{b}}_i\end{bmatrix}$$

When $A$ is just a row vector $\mathbf{a}$, we get a very simple expression of $\mathbf{a}B$:

$$\mathbf{a}B = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_n\mathbf{b}_n$$

If these three definitions are confusing, try writing out some arbitrary matrix multiplications to see that they are equivalent.

### 1.1.3 The matrix transpose

**Definition 6** (the transpose of a matrix)**.** *The transpose of an* $m \times n$ *matrix A is an* $n \times m$ *matrix* $A^T$ *such that*

$$A^T_{ij} = A_{ji} \; \forall \; i, j.$$

*(*$\forall$ *means "for all.")*

As an example,

$$\begin{bmatrix}3 & -2 & 1 \\ 7 & 6 & -1\end{bmatrix}^T = \begin{bmatrix}3 & 7 \\ -2 & 6 \\ 1 & -1\end{bmatrix}.$$

## 1.2 Matrix algebra

A number of properties extend from our definitions of matrices and their operations.

### 1.2.1 Associativity

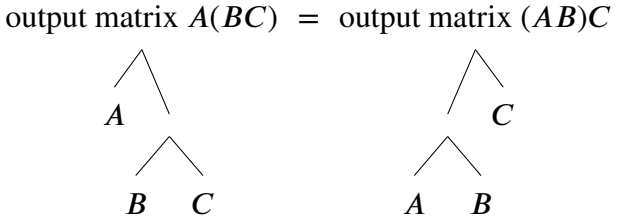
**Theorem 1** (associativity of matrix multiplication)**.** $A(BC) = (AB)C$*, provided that* $A \in \mathbb{R}^{k \times l}$*,* $B \in \mathbb{R}^{l \times m}$*, and* $C \in \mathbb{R}^{m \times n}$ *(the matrices have valid dimensions for the matrix multiplications to work out).*

*Proof.* If we write out $(A(BC))_{ij}$ for any $i$ and $j$ in range, we can rearrange sums and simplify to show that this equals $((AB)C)_{ij}$:

$$\begin{aligned}(A(BC))_{ij} = \sum_{h=1}^{l} a_{ih} bc_{hj} = \sum_{h=1}^{l} a_{ih} \sum_{o=1}^{m} b_{ho} c_{oj} = \sum_{o=1}^{m} \sum_{h=1}^{l} a_{ih} b_{ho} c_{oj} \\ = \sum_{o=1}^{m} \left( \sum_{h=1}^{l} a_{ih} b_{ho} \right) c_{oj} = \sum_{o=1}^{m} ab_{io} c_{oj} = ((AB)C)_{ij}\end{aligned}$$

□

In mathematics, one generally specifies implicitly the order in which arithmetic operations should be performed. That task is the work of the parentheses here. However, we can also describe the matrix multiplication $A(BC)$ through a tree representation:




To compute a result specified by a tree representation, you would start from the bottom of the tree and gradually move up, performing a matrix multiplication wherever two branches join together. Note that shifting between these two representations preserves the in-order traversal of the tree's leaves, in this case $A$, $B$, and $C$. (An **in-order traversal** is the order in which you would visit the tree's leaves, or

matrices, if you traced the bottom outline of the tree with your pencil. If you have a matrix multiplication written in parentheses form, like $(A(BC))D$, the in-order traversal is given by reading off the matrices from left to right.)

Why might a tree representation be important? As we will see, such a representation allows us to state a more general associative property:

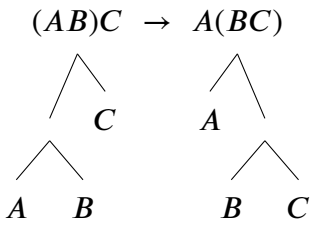
**Theorem 2** (associativity, in full generality)**.** *Any two matrix multiplications involving matrices $A_1$ through $A_n$ and possessing the same in-order traversal are equivalent.*

Before proving this theorem, it is useful to have the following lemma:

**Lemma 1.** *Given any number $i \in \{2, \ldots, n\}$, a tree with in-order traversal $A_1, \ldots, A_n$ can be partitioned such that all leaves $< i$ are descendants of the tree's root's left branch (on the left half of the tree), and all leaves $\geq i$ descend from the right branch.*

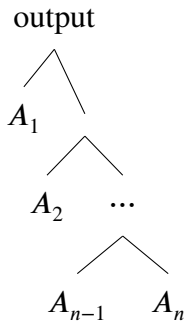
*Proof.* Following is a constructive proof.

1. If one exists, find the top-most junction which has two children and is the left child of its parent. Use the simple associative property first shown to "sink" the junction so that it is right-heavy, i.e. convert from the left tree to the right tree below.

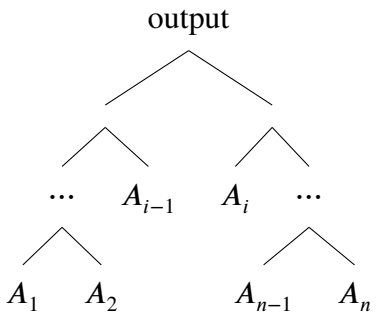



   (a) Repeat this process from top to bottom, until no such junctions exist. At that point, the tree will be transformed so that it descends to the

right:




2. Now, $i-2$ times, apply the associative property to the root junction so that the tree sinks to the left. This will yield the tree




Sure enough, this tree obeys the partition property stated in the lemma. □

Now, we are ready to see why associativity in full generality (the second theorem above) holds. If we have the tree representations of two matrix multiplications, with identical in-order traversals, we can always get from the first representation to the second: For the root node of the second tree, find which nodes descend from the root's left child, and which descend from the root's right child. From the lemma, we know we can partition the first tree so that the same leaves descend from either side of the first tree's root; this will preserve the in-order traversal of both sides. Then, recursively apply this procedure to the first tree's left and right sub-trees, if they are non-trivial (have more than 2 leaves). In the end, the first tree will become such that if you examine any junction, the same leaves will descend from the left and right branches, as from the same junction in the original second tree; the first tree is now equivalent to the second tree. And since we used only the

simple associative property, the matrix multiplications seen throughout this procedure are equivalent, so the original first and second trees are equivalent, assuming only that they have the same in-order traversal.

### 1.2.2 Other properties

Although associativity is a useful and rather amazing property to study, it is far from the only notable one. Here are some other important properties, with proofs omitted for brevity.

1. Although associativity holds, matrix multiplication is often **not** commutative ($AB$ does not necessarily equal $BA$), by a simple counterexample.
2. Right-distributive multiplication: $A(B + C) = AB + AC$.
3. Left-distributive multiplication: $(A + B)C = AC + BC$.
4. For any $c \in \mathbb{R}$, $c(AB) = (cA)B = A(cB)$.
5. $I_m A = A = AI_n$ (where $A$ is $m \times n$ and $I_m$ is the square identity matrix, with ones down the diagonal and zeroes everywhere else).
6. $(AB)^T = B^T A^T$.
7. $(A + B)^T = A^T + B^T$.

# Chapter 2

# Systems of linear equations

A **linear equation** is an equation of the form

$$c_1 x_1 + c_2 x_2 \cdots c_n x_n = m,$$

where $x_1, \ldots x_n$ are unknown values and $c_1, \ldots c_n$ and $m$ are known constants. Note that whereas before we were mostly focused on the rules and properties of matrix operations, we are now discussing unknown values, which are useful concepts in discussing questions of a form similar to "For what values of **x** does this equality hold?" Additionally, the above expression left of the equals sign is called a **linear combination**: a sum of terms, in this case denoted $x_1$ through $x_n$, and each multiplied by some constant $c_i$.

Naturally, a **system of linear equations** is then a group of linear equations, like above, whose solution(s) are all the combinations $x_1, \ldots x_n$ which satisfy all equations in the system. Many real-world problems involve systems of linear equations. Some common, fundamental questions concerning linear equations include whether a solution exists (the existence problem), what such a solution might be, and if a solution is unique (the uniqueness problem); all of these questions we will soon consider in greater depth.

## 2.1 Matrix equations = systems of linear equations

Perhaps the most convenient form to write a system of linear equations is the **matrix equation**, $A\mathbf{x} = \mathbf{b}$. Keeping in mind the above definitions of matrix multiplication, now would be a great time to convince yourself that the following system of equations and two matrix equations are all equivalent. (The third style below

is called an **augmented matrix** and is shorthand for the matrix equation in the middle.)

$$\begin{cases} x_1 + 2x_2 = 0 \\ -5x_1 - x_2 = 3 \end{cases} \qquad \begin{bmatrix} 1 & 2 \\ -5 & -1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \qquad \left[\begin{array}{cc|c} 1 & 2 & 0 \\ -5 & -1 & 3 \end{array}\right]$$

Given any system of equations, one can populate the $i$th row of $A$ with the coefficients of the $i$th equation's left-hand side and fill $\mathbf{b}$ with the constants on the right-hand side of the equations. Then, after considering $\mathbf{x}$ to be the vector of unknowns $x_i$, the problem becomes solving $A\mathbf{x} = \mathbf{b}$: for which $\mathbf{x}$ does $A\mathbf{x} = \mathbf{b}$?

## 2.2 Solving systems of linear equations: Gaussian elimination

An algorithm known as **Gaussian elimination** or **row reduction** addresses the common problem of solving a system of linear equations, written in matrix form as $A\mathbf{x} = \mathbf{b}$. Gaussian elimination depends on three "operations" performed on systems of linear equations:

1. **Interchanging** two equations, i.e. switching the order in which they are presented
2. **Scaling** both sides of an equation by a real number $c \neq 0$
3. **Replacing** one equation with the sum of that equation and a scalar multiple of another

These operations preserve the solution set of the original system of equations. In matrix equation terminology, all intermediate matrix equations arising from the above three operations will be **row equivalent**, sharing the same solution set. Why is this?

### 2.2.1 Proof of correctness

**Theorem 3.** *Any system of equations reached through the three basic operations above (interchange, scaling, and replacement) will have the same solution set as the original system of equations.*

*Proof.* Denote $S_1$ as the system of equations before the operation, and $S_2$ as the system after. I will show that solutions to $S_1$ (before the operation) also solve $S_2$ (after the operation), and that each operation can reverse itself, so solutions to $S_2$ must also solve $S_1$, by the same logic. Then, because solving $S_1$ implies solving $S_2$ and vice versa, it follows that both systems have the same solution set, and the above operations preserve solution sets. Here is such analysis for each operation:

1. **Interchange**: This row operation is primarily for achieving a convention called upper-triangular form, soon to be explained. Switching the order equations are presented is obviously reversible, and the solution set is preserved because the equations themselves are unchanged.

2. **Scaling** by a nonzero constant: It is axiomatic that two equals scaled by the same real number will be equal. Multiplying by a *nonzero* constant $c$ is reversible through multiplication by $1/c$.

3. **Replacement**: One of Euclid's axioms in his "Common Notions" observes that "If equals are added to equals, then the wholes are equal." So if $\mathbf{x}$ satisfies both equations before a replacement (addition of $c$ times one equation to another), then $\mathbf{x}$ satisfies both equations after the replacement. Additionally, replacement can undo itself: if $c$ times one equation was added to another, add $-c$ times that same equation to the latter equation to re-obtain the original system.

That completes the proof. For future reference, note that this analysis also applies to systems of nonlinear equations; the above three operations preserve the solution set of any system of equations. □

### 2.2.2 The algorithm

Essentially, Gaussian elimination aims to use the three operations of interchange, scaling, and replacement to bring a system of *linear* equations to a form whose solution is obvious. Here, I will represent the matrix equation $A\mathbf{x} = \mathbf{b}$ as an augmented matrix.

1. Bring the matrix to **echelon** or **upper-triangular form**. A matrix is in echelon form exactly when a "staircase" of 0s goes up and to the left across the

matrix, hence the name "upper triangular" form. An example is

$$\left[\begin{array}{cccc|c} 1 & -6 & 4 & 0 & 0 \\ 0 & 3 & -1 & -1 & 4 \\ 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

- *Algorithm*: In the $i$th iteration, choose a row $j$ between row $i$ and row $n$ which has the leftmost **pivot** (the first nonzero entry in a row). If necessary, swap rows $i$ and $j$, promoting row $j$ as $i$th row ($i < j$). Next, using replacement, eliminate all nonzero entries below the $i$th row's pivot. Repeat for all rows $i = 1, \ldots, m$.
  - Often a row with the largest-magnitude numbers is promoted, for reasons involving computer representation of numbers and numerical stability.

2. Now, we have upper-triangular form. For each pivot from right to left, use replacement to create 0s above that pivot.

3. Scale each row to have a pivot of 1. This step may be performed simultaneously with the previous step.

### 2.2.3 Output and interpreting solutions

By the end of step 3, we have reached a state called **reduced echelon form**, in which the solution set can be easily described. Reduced echelon form is echelon form with 0s above all pivots and only pivots equal to 1. For example, the reduced echelon form for the above example is

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & -1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

Columns 1, 2, and 4 contain pivots of 1, but since column 3 has no pivot, its entries can be anything so long as upper-triangular form is respected.

At this point, we are ready to interpret the solution set. We have an equation for every pivot variable in terms of constants and possibly **free variables**, which are

non-pivot variables. In the above example, because matrix equations and systems of linear equations are equivalent, the final matrix equation is equivalent to

$$\begin{aligned} x_1 &= 2 - 2x_3 \\ x_2 &= 1/3 + x_3/3 \\ x_4 &= -3 \end{aligned}$$

Here I have subtracted the free variable, $x_3$, to the right side while leaving pivot variables on the left, to emphasize that any value of the free variable, $x_3$, yields a distinct solution.

We can also write our findings in **parametric vector form**, which essentially expresses the same equations with vectors:

$$\mathbf{x} = \begin{bmatrix} 2 - 2x_3 \\ 1/3 + x_3/3 \\ x_3 \\ -3 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 2 \\ 1/3 \\ 0 \\ -3 \end{bmatrix} + x_3 \begin{bmatrix} -2 \\ 1/3 \\ 1 \\ 0 \end{bmatrix}$$

### 2.2.4 The existence and uniqueness of solutions

By simple example, the algorithm can sometimes encounter an impossible equation, such as $0 = 1$. Obviously, no $\mathbf{x}$ would satisfy such an equation; the solution set would be empty, and the system of equations would be called **inconsistent**. But in the above example, we were lucky enough not to encounter an inconsistent equation. One row of all 0s came up in $A$, but it corresponded to the entry 0 in $\mathbf{b}$; the row represented the trivially satisfied equation $0 = 0$. When the equations are all consistent, as so, there are two possibilities:

- We have **no free variables** (equivalently, every column in $A$ contains a pivot). Each equation either is $0 = 0$ or assigns a pivot variable to a constant. There exists one unique solution.
- There is **at least one free variable**. Since a free variable can be any real number, an infinite number of solutions exist.

Indeed, only the three cases — no solution (an inconsistent system), one solution, and an infinite number of solutions — can arise. The existence of two distinct solutions to $A\mathbf{x} = \mathbf{b}$ actually implies an infinite number of solutions:

*Proof.* Say that for $\mathbf{u} \neq \mathbf{v}$, both $A\mathbf{u} = \mathbf{b}$ and $A\mathbf{v} = \mathbf{b}$. We can choose an infinite number of two real numbers $m$ and $n$ such that $m+n = 1$. Then by the distributive and scalar-multiplication properties of matrix multiplication,

$$A(m\mathbf{u} + n\mathbf{v}) = mA\mathbf{u} + nA\mathbf{v} = (m+n)\mathbf{b} = \mathbf{b},$$

so any such $m\mathbf{u} + n\mathbf{v}$ is a solution. Also, by requiring $m, n \geq 0$, the same logic applies to the matrix inequalities $A\mathbf{x} \leq \mathbf{b}$ and $A\mathbf{x} \geq \mathbf{b}$. □

## 2.3 The (P)LU decomposition: an algorithm

The **PLU decomposition** factors an $m \times n$ matrix $A$ into $PA = LU$, with $L$ an $m \times m$ lower-triangular matrix, $U$ an $m \times n$ upper-triangular matrix, and $P$ an $m \times m$ **permutation matrix**. A permutation matrix has a single 1 in every row and column, and 0s elsewhere; an example is

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

We call the special case in which $P = I$ an **LU decomposition**. The (P)LU decomposition provides a more computationally efficient way to solve $A\mathbf{x} = \mathbf{b}$, by reformulating the problem as $LU\mathbf{x} = P\mathbf{b}$. To solve the new problem, we just need to solve $L\mathbf{y} = P\mathbf{b}$, and if such a $\mathbf{y}$ exists, then find $\mathbf{x}$ for which $U\mathbf{x} = \mathbf{y}$. This is easier because the matrix equations involved have only triangular matrices.

Using the simple Gaussian elimination algorithm we have already learned, we can perform a $PLU$ decomposition for any matrix. The decomposition can be done as follows.

1. Initialize $P = L = I_{m \times m}$ and $U = A$. Right now, $PA = LU$ is satisfied trivially, because $IA = IA$.

2. Following the first step of Gaussian elimination, row reduce $U$ to upper-triangular form. Since Gaussian elimination will always succeed in reducing $U$, we have a working algorithm, if only we can specify a way at each step to change $P$ and $L$ that maintains $PA = LU$, preserves $P$ as a permutation matrix, and keeps $L$ lower-triangular. I shall specify such instructions now.

   (a) In iteration $i$ of Gaussian elimination, you might add $c$ times $U$'s row $i$ to row $j$, but only when $i < j$.

i. The row-wise view of matrix multiplication shows that row $h$ of $L$ contains the weights of a linearly combination of the rows of $U$ which will yield row $h$ of $LU$. Thus, we can compensate for adding $c$ times row $i$ to row $j$ of $U$: for every nonzero entry $l_{hj} \neq 0$ in $L$'s $j$th column, add $-cl_{hj}$ to $l_{hi}$ (row $h$, column $i$).

ii. Now, the value of $LU$ will remain constant, maintaining $PA = LU$. And since we only edited left of $L$'s main diagonal ($i < j$), $L$ will remain lower-triangular.

(b) Also, at the beginning of iteration $i$ of Gaussian elimination, you may need to swap row $j$ to row $i$ of $U$. Go ahead, but we will edit both $P$ and $L$ to maintain $PA = LU$ as well as the permutation and lower-triangular properties of these matrices.

i. At the beginning of iteration $i$, only up to column $i - 1$ of $L$ can contain nonzero entries, with the exception of the main diagonal. This is because nonzero entries only appear in $L$ during row replacement, and if the replacement happens during iteration $i$, only column $i$ is changed.

ii. Say now that we swap rows $i$ and $j$ of $U$ and swap entries $l_{ih}$ and $l_{jh}$ of $L$, for $h = 1, \ldots, i - 1$. Overall, this will create the effect of swapping rows $i$ and $j$ of $LU$. To account for swapping the rows in $LU$, swap rows $i$ and $j$ of $P$ as well, thereby maintaining $PA = LU$. Rest assured that swapping rows of a permutation matrix does always give another permutation matrix.

A. *Note*: Computer programs will often swap a row with high-magnitude entries toward the top, for reasons involving numerical stability and how decimal numbers are represented in computers.

(c) If you scale row $i$ of $U$ by $c \neq 0$, scale column $j$ of $L$ by $1/c$ for $LU$ to remain constant.

You can hopefully see that this algorithm will always converge to a $PLU$ decomposition. We only ever change entries left of $L$'s diagonal; we only alter $P$ by switching its rows, yielding another permutation matrix; and Gaussian elimination guarantees that $U$ will reach upper-triangular form.

An interesting question is, when does $P = I$ in the final solution — a true "$LU$" decomposition, without a $P$ matrix? Using the above algorithm, $P$ can

equal $I$ if row reducing $A$ does not require a row interchange, but if an interchange is performed, $P \neq I$.

### 2.3.1 A numerical example

Let

$$A = \begin{bmatrix} 6 & -2 & 0 & 3 \\ 0 & 0 & -9 & 0 \\ 4 & -2 & 3 & 0 \end{bmatrix}.$$

Initialize

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & -2 & 0 & 3 \\ 0 & 0 & -9 & 0 \\ 4 & -2 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & -2 & 0 & 3 \\ 0 & 0 & -9 & 0 \\ 4 & -2 & 3 & 0 \end{bmatrix}.$$

Now, we simply row-reduce $U$, the right-most matrix above, while updating the other matrices as previously described. We first add $-2/3$ times row 1 to row 3, and then swap rows 2 and 3:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & -2 & 0 & 3 \\ 0 & 0 & -9 & 0 \\ 4 & -2 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2/3 & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & -2 & 0 & 3 \\ 0 & 0 & -9 & 0 \\ 0 & -2/3 & 3 & -2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 6 & -2 & 0 & 3 \\ 0 & 0 & -9 & 0 \\ 4 & -2 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & -2 & 0 & 3 \\ 0 & -2/3 & 3 & -2 \\ 0 & 0 & -9 & 0 \end{bmatrix}$$

# Chapter 3

# Determinants of square matrices

**Definition 7** (the determinant)**.** *For a square matrix $A = \left[a_{ij}\right]$ with at least 2 rows, the determinant of A is defined as*

$$\det A = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det A_{1j},$$

*where $A_{ij}$ is A without the ith row and jth column, and $a_{ij}$ is the entry of A in the ith row and jth column.*

One main use of the determinant is as one of the conditions of the invertible matrix theorem, to be covered later.

**Theorem 4.** *The "**cofactor expansion**" of A along any of A's rows and columns is equal to the determinant of A. In particular, the cofactor expansion along the ith row is*

$$\sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det A_{ij} = \det A,$$

*and the cofactor expansion along the jth column is*

$$\sum_{i=1}^{n} (-1)^{i+j} a_{ij} \det A_{ij} = \det A.$$

*The original definition uses a cofactor expansion along the first row, but any row or column works.*

## 3.1 Some determinant properties

1. The determinant of an upper- or lower-triangular matrix $A$ is the product of $A$'s diagonal entries. For lower-triangular matrices, this can be seen by taking the determinant as cofactor expansions along only the top row. For upper-triangular matrices, take cofactor expansions along the bottom row.

2. $\det AB = \det A \det B$. Justification is omitted for brevity.

3. $\det A^T = \det A$, because as we have established, row and column expansions are two approaches which yield the same determinant, and doing cofactor expansions along the columns of $A^T$ is equivalent to doing cofactor expansions along the rows of $A$.

## 3.2 More efficient computation of determinants

The above method is very computationally demanding, and the problem complexity quickly much more quickly that the problem size. Fortunately, there is a more efficient method, and it depends primarily on the following three facts:

1. Interchanging rows negates a matrix's determinant.

2. Scaling a row by a constant multiplies the determinant by that constant.

3. Replacement (adding a multiple of one row to another) does not change the determinant.

This method entails reducing a square matrix to echelon form using the three operations of Gaussian elimination. At each operation, write an expression for the determinant of the matrix's current form, in terms of the determinant of the original matrix. Once echelon form is achieved, calculate the determinant of the current matrix to be the product of the diagonals, according to property 1 above. This calculated determinant equals the expression for the current matrix's determinant, which you've written in terms of the original matrix's determinant. Using this fact, solve for the original determinant.

# Chapter 4

# Linear transformations

## 4.1 The matrix multiplication of a linear transformation

**Definition 8.** *A function $T : \mathbb{R}^n \to \mathbb{R}^m$ is linear if the following properties hold for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $c \in \mathbb{R}$:*

$$T(\mathbf{x} + \mathbf{y}) = T(\mathbf{x}) + T(\mathbf{y}) \tag{4.1}$$

$$T(c\mathbf{x}) = cT(\mathbf{x}) \tag{4.2}$$

Recall that $\mathbb{R}^n$, the input of $T$, is the set of all real- or decimal number-valued vectors of length $n$; likewise, $T$'s output is vectors of length $m$. Equivalently, one can consider $T$ in this definition to contain $m$ different linear functions from $\mathbb{R}^n$ to $\mathbb{R}$.

Next, let $\mathbf{e}_i$ be a vector containing all zeroes except 1 as the $i$th entry. By consequence of these properties, for a linear transformation $T$ and any vector $\mathbf{x} \in \mathbb{R}^n$,

$$T(\mathbf{x}) = T(x_1\mathbf{e}_1 + \cdots + x_n\mathbf{e}_n) = x_1T(\mathbf{e}_1) + \cdots + x_nT(\mathbf{e}_n).$$

This observation leads us to the following finding:

**Theorem 5.** *For all $i = 1, \ldots, n$, a* linear *transformation $T : \mathbb{R}^n \to \mathbb{R}^m$ maps $\mathbf{e}_i \in \mathbb{R}^n$ to the vector $T(\mathbf{e}_i) \in \mathbb{R}^m$. Due to the linearity of $T$, $T(\mathbf{x})$ can be computed as*

$$x_1T(\mathbf{e}_1) + \cdots + x_nT(\mathbf{e}_n),$$

*provided that one knows $T(\mathbf{e}_i)$. This form happens to match the form of the following matrix multiplication:*

$$T(\mathbf{x}) = A\mathbf{x}, \textit{with } A = \begin{bmatrix} T(\mathbf{e_1}) & T(\mathbf{e_2}) & \cdots & T(\mathbf{e_n}) \end{bmatrix}$$

*$A\mathbf{x}$ therefore implements the linear transformation $T(\mathbf{x})$.*

Additionally, this matrix $A$ is unique: If a different matrix, $B$, correctly implemented the transformation, then one could choose the $i$th column where the two matrices differed and multiply the two matrices by $\mathbf{e}_i$. Since the $i$th columns of $A$ and $B$ would be different, $A\mathbf{e}_i$ and $B\mathbf{e}_i$ must map to different results. Because $T$ is a function, it certainly should not map to different results.

This theorem allows for easy computation with matrix multiplication of often seemingly complex transformations. For instance, consider rotation by $\theta$, in two dimensions, about $(0, 0)$. Determining the output of this transformation may seem complicated at first, but the transformation is actually linear for vectors in two dimensions. We can easily implement a rotation matrix if we simply use trigonometry to find the rotated versions of $\mathbf{e_1} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and $\mathbf{e_2} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$.

## 4.2 Composition of linear transformations

Consider a composition of linear functions $T_A \circ T_B$, defined so that $T_A \circ T_B(\mathbf{x}) = T_A(T_B(\mathbf{x}))$. Suppose also that we have matrices $A$ and $B$ implementing $T_A$ and $T_B$, as in the previous section. Through a simple proof, one can show that the composition of linear functions $T_A \circ T_B$ is itself linear. Using this linearity and our previous findings regarding linear transformations, the matrix of the linear transformation $T_A \circ T_B$ must be

$$\begin{aligned} \begin{bmatrix} T_A(T_B(\mathbf{e_1})) & T_A(T_B(\mathbf{e_2})) & \cdots & T_A(T_B(\mathbf{e_n})) \end{bmatrix} &= \begin{bmatrix} T_A(B\mathbf{e_1}) & \cdots & T_A(B\mathbf{e_n}) \end{bmatrix} \\ &= \begin{bmatrix} T_A(\mathbf{b_1}) & \cdots & T_A(\mathbf{b_n}) \end{bmatrix} = \begin{bmatrix} A\mathbf{b_1} & \cdots & A\mathbf{b_n} \end{bmatrix} = AB. \end{aligned}$$

As you see, the matrix of $T_A \circ T_B$ turns out to be $AB$, the product of each composed transformation's matrix. We can extend this logic to an arbitrarily long transformation $T_{A_1} \circ T_{A_2} \circ \cdots \circ T_{A_n}$, whose transformation matrix must be $A_1 A_2 \cdots A_n$.

## 4.3 A linear transformation's inverse and the invertible matrix theorem

Let $T$ be a linear transformation, and $A$ be its corresponding matrix. Does an inverse transformation exist for $T$, and what is its matrix? In this section, we shall concern ourselves only with transformations $T$ from $\mathbb{R}^m$ to $\mathbb{R}^m$, corresponding to square $A$. For the more general case of transformations from $\mathbb{R}^n$ to $\mathbb{R}^m$, read about **pseudoinverse matrices**, a topic typically requiring some knowledge of the singular value decomposition, which is covered later in this book.

First, let us acquire a bit of intuition. Imagine a linear transformation $T$ defined so that $T(\mathbf{y}) = \mathbf{0}$. It makes sense that no inverse transformation would correspond to $T$. You would have no idea what $\mathbf{x}$ yielded $\mathbf{0}$ as an output — it could've been $\mathbf{y}$, or $0.1\mathbf{y}$, or even $10000\mathbf{y}$! In fact, in any case that $T$ is not one-to-one (one output corresponds to one input and vice versa), an inverse function would be meaningless, because one output would have to map to multiple inputs. However, it turns out that a linear inverse transformation does exist for many linear transformations, and we get the nice commutative property between $A$ and its inverse transformation matrix $A^{-1}$, so that

$$AA^{-1} = A^{-1}A = I.$$

If this reasoning feels loose, you're in luck: formal conditions will be solidified now in the invertible matrix theorem. Let us build up to the invertible matrix theorem with a lemma.

**Lemma 2** (preliminary to the invertible matrix theorem)**.** *If a linear transformation $T : \mathbb{R}^m \to \mathbb{R}^m$ is one-to-one and spans $\mathbb{R}^m$, it has an inverse transformation $T^{-1}$ which is likewise linear. If the transformation neither is one-to-one nor spans $\mathbb{R}^m$, no inverse transformation exists such that $T^{-1}(T(\mathbf{x})) = \mathbf{x}$ for all $\mathbf{x}$.*

*Proof.* First consider the negative case. If $T$ is not one-to-one, it maps multiple $\mathbf{x}_1, \ldots, \mathbf{x}_\infty$ to the same $\mathbf{y}$ ($T(\mathbf{x}_i) = \mathbf{y}$). Plugging in any of these $\mathbf{x}_i$, we get

$$T^{-1}(T(\mathbf{x}_i)) = T^{-1}(\mathbf{y}),$$

but $T^{-1}$ is a function and can only map back to one $\mathbf{x}_i$. Hence, for all but at most one $\mathbf{x}_i$, $T^{-1}(T(\mathbf{x}_i)) \neq \mathbf{x}_i$.

Now, think of the positive case, in which $T$ is one-to-one and onto. Because of the one-to-one correspondence of inputs and outputs, it is theoretically possible to make a function mapping outputs to inputs. But would this transformation, $T^{-1}$,

be linear, and thus implemented as a matrix multiplication? Sure enough, $T^{-1}$ satisfies both linearity properties:

Property 1:

$$T(\mathbf{y}) = \mathbf{x} \text{ and } T(c\mathbf{y}) = c\mathbf{x}$$
$$T^{-1}(c\mathbf{x}) = c\mathbf{y} = cT^{-1}(\mathbf{x})$$

Property 2:

$$T(\mathbf{y}_1) = \mathbf{x}_1 \text{ and } T(\mathbf{y}_2) = \mathbf{x}_2$$
$$T(\mathbf{y}_1 + \mathbf{y}_2) = \mathbf{x}_1 + \mathbf{x}_2$$
$$T^{-1}(\mathbf{x}_1 + \mathbf{x}_2) = \mathbf{y}_1 + \mathbf{y}_2 = T^{-1}(\mathbf{x}_1) + T^{-1}(\mathbf{x}_2)$$

$T^{-1}$ is also linear, and can be modelled with a matrix $A^{-1}$. □

**Theorem 6** (the invertible matrix theorem)**.** *The following statements regarding a linear transformation $T$ and its corresponding square matrix $A$ are either all true or all false.*

- *$T$ has a corresponding linear inverse transformation $T^{-1}$.*
  - *$A$ has an corresponding inverse matrix $A^{-1}$.*

    *There is a matrix $B$ such that $BA = AB = I$.*
  - *$A^T$ has an inverse.*
- *$T$ is one-to-one.*
  - *Row reducing $A$ yields a pivot in every column and every row (no free variables will be possible in Gaussian elimination).*
  - *$A\mathbf{x} = \mathbf{0}$ has only the trivial solution of $\mathbf{x} = \mathbf{0}$.*
- *$T$ is **onto**, in that every vector in $\mathbb{R}^m$ is reachable by some input to $T$.*
  - *In terms of $A$, its columns span $\mathbb{R}^m$.*

    *The columns of $A$ are linearly independent.*

    · *The **rank** of $A$ is $m$.*

    *$A$'s rows span $\mathbb{R}^m$.*

- det $A \neq 0$.

*Proof.* First, consider if $T$ is one-to-one. We would like to first prove the true case, in which this fact implies all of the other conditions are also true. Then, we will consider if $T$ is not one-to-one, and prove the other conditions are all false.

If $T$ is one-to-one, then there can be no free variables when row reducing $A$ in $A\mathbf{x} = \mathbf{b}$, or else the null space would be nonempty ($A\mathbf{x} = \mathbf{0}$ would have nontrivial solutions). Therefore, a pivot is in every column and every row of $A$, and $A\mathbf{x}$ can reach any vector in $\mathbb{R}^m$ (equivalently, $T$ is onto). Since we have already established $A\mathbf{x} = \sum_i \mathbf{a}_i x_i = \mathbf{0}$ has only the trivial solution, the columns are linearly independent by definition ($A\mathbf{x} = \mathbf{0}$ has only the trivial solution $\mathbf{x} = \mathbf{0}$). Next, from this we also know no column of $A$ can be described as a linear combination of the other columns. For reasons established later, the rank of $A$ is the number of pivots, $m$. Also, $A$ row reduces to $I$, and row reduction is actually just linear combination of the rows. Some linear combination of the rows, or $A^T$'s columns, can lead to every $\mathbf{e}_i$, so the rows span $\mathbb{R}^m$ because matrix multiplication, in this case by $A^T$, is linear. Moreover, from the discussion in the determinant section of a more efficient method of calculating the determinant, one can see that if a matrix row reduces to $I$, its determinant will be nonzero. Lastly, the preceding lemma indicates that an inverse transformation $T^{-1}$ and matrix $A^{-1}$ do exist when $T$ is one-to-one and onto.

If $T$ is not one-to-one, then by definition multiple inputs map to the same output. A free variable must exist after row reducing $A$, and $A\mathbf{x} = \mathbf{0}$ has nontrivial solutions; $A$'s columns are linearly dependent. Since a pivot is not in every row, $T$ cannot be onto, and $A$'s columns do not span $\mathbb{R}^m$. For reasons established later, $A$'s rank is the number of pivots, and is less than $m$. Since $A$ row reduces to have a 0 on its diagonal, $\det A = 0$. Also, some nontrivial combination of the rows of $A$ (columns of $A^T$) yielded the $\mathbf{0}$ vector, so $A^T$ is not full rank and has no inverse. Lastly, as mentioned earlier, it also makes no sense to have an inverse transformation or inverse matrix since $T$ maps multiple vectors to the same output, and the inverse would have to arbitrarily choose one of these inputs to map to. By similar logic, no inverse matrix $A^{-1}$ could really exist such that $A^{-1}A = I$. Returning to the example of multiple $\mathbf{x}$ mapping to the same $\mathbf{y}$, suppose an inverse did exist — by the associativity of matrix multiplication, it would have to be true that $A^{-1}A\mathbf{x} = (A^{-1}A)\mathbf{x} = \mathbf{x} = A^{-1}(A\mathbf{x}) = A^{-1}\mathbf{y}$. Yet $A^{-1}$ can only map $\mathbf{y}$ to one $\mathbf{x}$, even though $\mathbf{x}$ could have been any of an infinite number of vectors that map to $\mathbf{y}$. □

### 4.3.1 Computing matrix inverses

As we have discovered, if a transformation $T$ is invertible, its inverse will be linear and can be implemented as a matrix in the same way we have discussed for other linear transformations. We want to implement the transformation through matrix multiplication, so we do what we might do for any linear transformation: find what it maps each $\mathbf{e}_i$ to. Since $T^{-1}$ is an inverse of $T$, we are essentially solving the equation $T(\mathbf{x}_i) = A\mathbf{x}_i = \mathbf{e}_i$, with $\mathbf{x}_i$ to be the $i$th column of $A^{-1}$.

In cases like these, in which we want to compute $A\mathbf{x} = \mathbf{b}$ for $i > 1$ different $\mathbf{b}$ but the same $A$ every time, we employ a more efficient form of Gaussian elimination called **Gauss-Jordan elimination**. See a search engine for more info in this algorithm; it is almost identical to Gaussian elimination except you have multiple $\mathbf{b}$ column vectors on the right.

# Chapter 5

# Vector spaces

**Definition 9** (a vector space)**.** *A vector space is a nonempty set of "vectors" over which are defined the operations of addition and scalar multiplication by a real number. (These operations can be technically defined in any way, as long as the following hold.) A vector space is defined by the following 10 axioms for any* $\mathbf{u}, \mathbf{v} \in V$*:*

1. $\mathbf{u} + \mathbf{v}$ *is in* $V$.
2. *Commutative addition*
3. *Associative addition*
4. *There is a zero vector* $\mathbf{0} \in V$ *which provides an additive identity operation:* $\mathbf{x} + \mathbf{0} = \mathbf{x}$.
5. *For each* $\mathbf{u}$*, there is an additive inverse element* $-\mathbf{u}$.
6. $c\mathbf{u} \in V$ *for all real numbers* $c$.
7. *Right-distributive scalar multiplication:* $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$.
8. *Left-distributive scalar multiplication:* $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$.
9. $c(d\mathbf{u}) = (cd)\mathbf{u}$.
10. $1\mathbf{u} = \mathbf{u}$.

Common examples of vector spaces include not just $\mathbb{R}$ and $\mathbb{R}^m$, but

- the set of continuous functions $\{f : \mathbb{R} \to \mathbb{R}\}$,
- the set of integrable functions $\{f : [a, b] \to \mathbb{R}\}$,
- the set of polynomials with real coefficients, and
- the solution sets to differential equations.

## 5.1 Subspaces

**Definition 10** (a subspace of a vector space such as $\mathbb{R}^n$)**.** *A subspace is any set $H$ in a vector space $V$ satisfying three properties:*

1. $\mathbf{0} \in H$.
2. *For all* $\mathbf{u}, \mathbf{v} \in H$, $\mathbf{u} + \mathbf{v}$ *is in* $H$.
3. *For all* $\mathbf{u} \in H, c \in \mathbb{R}$, $c\mathbf{u}$ *is also in* $H$.

A subspace is described by a **basis**, which is a set of vectors which are linearly independent and span the subspace. I.e., a linear combination of basis vectors can reach any vector in a subspace. The **dimension** of a subspace is the number of vectors in the subspace's basis, which turns out to always be the same. Similarly, the dimension of the column space of $A$ is called the **rank** of $A$.

Examples of subspaces include the **column space** of a matrix $A$, which is all vectors reachable as $A\mathbf{x}$; and the **null space** of $A$, the set of all solutions to $A\mathbf{x} = \mathbf{0}$.

**Theorem 7.** *Any basis of a subspace $H \in \mathbb{R}^l$ will contain the same number of vectors as the dimension of $H$,* $\dim H$*. The dimension can be found as the number of any set of vectors which are linearly independent and span $H$, and therefore .*

*Proof.* Suppose the counterexample: Two bases exist, $A$ and $B$, with different numbers of vectors. Specifically, $A$ has fewer vectors, $n$, while $B$ has $m$ vectors, with $m > n$. However, both are supposedly linearly independent and spanning. I seek to show that this is impossible, and the longer basis cannot possibly be linearly independent, as the matrix equation $B\mathbf{x} = \mathbf{0}$ must have a nontrivial solution.

Note that since $A$, the smaller basis, is spanning, $B$ can be written as follows:

$$B = \begin{bmatrix} \sum_{i=1}^{n} c_{1i}\mathbf{a}_i & \cdots & \sum_{i=1}^{n} c_{mi}\mathbf{a}_i \end{bmatrix}$$

We can then expand $B\mathbf{x}$ as

$$B\mathbf{x} = \sum_{i=1}^{n} c_{1i}x_1\mathbf{a}_i + \cdots + \sum_{i=1}^{n} c_{mi}x_m\mathbf{a}_i = \sum_{j=1}^{m}\sum_{i=1}^{n} c_{ji}x_j\mathbf{a}_i$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ji}x_j\mathbf{a}_i = \sum_{i=1}^{n}\mathbf{a}_i\sum_{j=1}^{m} c_{ji}x_j = \sum_{i=1}^{n}\mathbf{a}_i(\mathbf{c}_i^T\mathbf{x}).$$

Next, choose $\mathbf{x}$ so that all $\mathbf{c}_i^T\mathbf{x} = 0$. This can be done equivalently by solving the matrix equation $C\mathbf{x} = \mathbf{0}$, or:

$$\left[\begin{array}{c|c} \mathbf{c}_1 & 0 \\ \vdots & \vdots \\ \mathbf{c}_n & 0 \end{array}\right]$$

Since $m > n$, this matrix is wide, not tall, and therefore must row reduce to have free variables. The number of solutions $\mathbf{x}$ is infinite. Additionally, we need not worry about inconsistent equations arising while solving this matrix equation, since the right-hand-side will always consist of only 0s. Once an $\mathbf{x}$ is chosen, we can be sure that it non-trivially satisfies $B\mathbf{x} = \mathbf{0}$, from the previous expansion of $B\mathbf{x}$. The longer "basis" is actually linearly dependent and not a basis at all. □

**Theorem 8.** *The columns of A which correspond to pivot columns in A's row-reduced form a basis of the column space of A.*

*Proof.* Consider any $\mathbf{b}$ be in the column space of $A$, so that $A\mathbf{x} = \mathbf{b}$ has a solution. If row reducing $A$ yields free variable(s), one can set those free variables to be 0, and the solution $\mathbf{x}$ becomes a linear combination of only $A$'s pivot columns. Since this is true of any $\mathbf{b} \in \text{Col } A$, the columns corresponding to pivots span Col $A$.

Additionally, row reducing a matrix with just the pivot columns reveals a pivot in every column. Since there are no free variables, the null space is empty aside from $\mathbf{0}$, and these "pivot columns" are linearly independent. Both properties of a basis, the spanning and independence properties, are satisfied. □

**Theorem 9** (the rank theorem)**.** *Consider a matrix A with n columns. Then*

$$\dim \text{Col } A + \dim \text{Nul } A = n.$$

*Proof.* We have already proven that dim Col $A$, also denoted as rank $A$, is the number of pivot columns in $A$. Now, I seek to show that the number of free variables in a matrix $A$'s row-reduced version, $n - \dim \text{Col } A$, is the dim Nul $A$.

Typically, solving for $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{0}$ leads to something like this:

$$\begin{aligned} x_1 &= -2x_3 \\ x_2 &= x_3/2 + x_5 \\ x_4 &= 0 \end{aligned} \qquad \mathbf{x} = \begin{bmatrix} -2x_3 \\ x_3/2 + x_5 \\ x_3 \\ 0 \\ x_5 \end{bmatrix} = x_3 \begin{bmatrix} -2 \\ 1/2 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

It seems perfectly reasonable that for $n$ free variables, we will always get $n$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ in parametric vector form, used above to the right. Now, notice that each vector is multiplied by a free variable $x_i$, and the $i$th entry of that vector is always 1, while the $i$th entry of all the other vectors is always 0. Since each vector contains at least one entry which is nonzero for that vector and zero for all others, the linear dependence equation $\sum_i c_i \mathbf{v}_i = \mathbf{0}$ can only have the trivial solution $\mathbf{c} = \mathbf{0}$. If it had a nontrivial solution, one of the entries corresponding to a free variable would have to be nonzero in $\sum_i c_i \mathbf{v}_i$. □

## 5.2 Coordinate systems

**Definition 11** (coordinates of a vector)**.** *Consider an $m$-dimensional basis $\mathcal{B}$ of a subspace of $\mathbb{R}^n$. Then the $\mathcal{B}$-coordinates of a vector $\mathbf{x}$ in that subspace are the vector $\mathbf{b}$ such that*

$$\mathbf{x} = \begin{bmatrix} \mathcal{B}_1 & \cdots & \mathcal{B}_m \end{bmatrix} \mathbf{b}.$$

Ideally, this concept could prove useful in data compression. You can imagine a scenario in which you have to track data in the form of vectors in $\mathbb{R}^n$, but all data points are part of a $p$-dimensional subspace, with $p << n$. One can track the $\mathcal{B}$-coordinates of the data and have to only keep $p$ numbers per data point, rather than $n$.

Unfortunately, the real world is riddled with noise, so data vectors are typically not perfect linear combinations of only a small handful of basis vectors. Similar methods such as principal component analysis address the inevitability of noisy data.

### 5.2.1 Change of bases

Suppose you want to change from $\mathcal{B}$-coordinates to $\mathcal{C}$-coordinates. In other words, you have two bases, $\mathcal{B}$ and $\mathcal{C}$, of a subspace of $\mathbb{R}^n$, and a vector $\left[\mathbf{x}\right]_{\mathcal{B}}$ such that

$$\begin{bmatrix}\mathbf{b}_1 & \cdots & \mathbf{b}_m\end{bmatrix}\left[\mathbf{x}\right]_{\mathcal{B}} = \mathbf{x} = \begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_m\end{bmatrix}\left[\mathbf{x}\right]_{\mathcal{C}}.$$

You would like to find $\left[\mathbf{x}\right]_{\mathcal{C}}$ in the above expression; all the rest is known. Certainly you could solve this Gaussian elimination problem each time you had to find $\left[\mathbf{x}\right]_{\mathcal{C}}$, but there turns out to be an easier way.

**Theorem 10.** *Change of coordinates is a linear transformation, so it can be implemented in one matrix multiplication.*

*Proof.* Changing from $\mathcal{B}$-coordinates to coordinates in $\mathbb{R}^n$ consists of the matrix multiplication $\mathbf{x} = \begin{bmatrix}\mathbf{b}_1 & \cdots & \mathbf{b}_m\end{bmatrix}\left[\mathbf{x}\right]_{\mathcal{B}}$. Since matrix multiplication is linear, this transformation is linear.

Now, the harder part: showing that converting from $\mathbf{x}$ to $\left[\mathbf{x}\right]_{\mathcal{C}}$ is also linear. Once this has been shown, we can say the entire transformation, which is the composition of two linear transformations, is itself linear.

Does this second transformation satisfy the two properties of linearity for any $c \in \mathbb{R}$ and $\mathbf{x}, \mathbf{y}$ in the subspace? yes, it does satisfy the first property:

$$c\mathbf{x} = c\begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_m\end{bmatrix}\left[\mathbf{x}\right]_{\mathcal{C}} = \begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_m\end{bmatrix}\left(c\left[\mathbf{x}\right]_{\mathcal{C}}\right) \implies \left[c\mathbf{x}\right]_{\mathcal{C}} = c\left[\mathbf{x}\right]_{\mathcal{C}}$$

I.e., if your input is $c\mathbf{x}$, your output will be $c$ times the output corresponding to $\mathbf{x}$. Additionally, this transformation satisfies the second linear property:

$$\begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_m\end{bmatrix}\left(\left[\mathbf{x}\right]_{\mathcal{C}} + \left[\mathbf{y}\right]_{\mathcal{C}}\right) = \begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_m\end{bmatrix}\left[\mathbf{x}\right]_{\mathcal{C}} + \begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_m\end{bmatrix}\left[\mathbf{y}\right]_{\mathcal{C}} = \mathbf{x} + \mathbf{y}$$

If your input is $\mathbf{x} + \mathbf{y}$, the one corresponding output is $\left[\mathbf{x}\right]_{\mathcal{C}} + \left[\mathbf{y}\right]_{\mathcal{C}}$, the sum of the individual outputs corresponding to $\mathbf{x}$ and $\mathbf{y}$. □

Since changing coordinates is a linear transformation, we can resort to the standard method of finding how the transformation responds to all the $\mathbf{e}_i$ vectors. This comes down to $n$ Gaussian elimination problems, the $i$th of which will look like

$$B\mathbf{e}_i = C\mathbf{x}_i,$$

with $\mathbf{x}_i$ the $i$th column of the transformation matrix. To simplify things, we can use Gauss-Jordan elimination to perform all $n$ Gaussian elimination problems at once.

## 5.3 Eigenvectors, eigenvalues, and eigenspaces

**Definition 12.** *An **eigenvector** $\mathbf{x} \in \mathbb{R}^n$ of a transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ an $n \times n$ matrix $A$ is defined by two properties:*

*(1)* $\mathbf{x} \neq \mathbf{0}$.

*(2)* $A\mathbf{x} = \lambda\mathbf{x}$ *for some* $\lambda \in R$.

*$\lambda$ is the **eigenvalue** corresponding to the eigenvector $\boldsymbol{x}$. The span of eigenvectors corresponding to an eigenvalue form a subspace known as an **eigenspace**.*

To find eigenvalues and their corresponding eigenvectors of a matrix $A$, we find $\lambda$ such that the equation $A\mathbf{x} = \lambda I\mathbf{x}$ is satisfied by some nontrivial $\mathbf{x}$. Alternately, we can solve for which $\lambda$ $(A - \lambda I)\mathbf{x} = \mathbf{0}$ has a nonempty solution set.

By the invertible matrix theorem, this technique is equivalent to finding which $\lambda$ give $\det(A-\lambda I) = 0$. Using this method, one will often translate the problem into solving for the roots $\lambda$ of a so-called "characteristic polynomial." The **algebraic multiplicity** of $a$ is the highest degree on $\lambda$ in the characteristic polynomial. The **geometric multiplicity** is the dimension of the eigenspace, and is, for unproven reasons, at least 1 and at most the algebraic multiplicity.

**Theorem 11** (linear independence of eigenvectors of different eigenvalues)**.** *Any set of nonzero eigenvectors of an $n \times n$ matrix $A$ which correspond to different eigenvalues (different eigenspaces) are linearly independent.*

*Proof.* Suppose the opposite case, in which the set of eigenvectors $S$, corresponding to unique eigenvalues, is dependent. In theory, we could check the linear independence of all possible subsets (every set in the power set) of $S$, and determine a positive integer $m \geq 2$ which is the minimum number of vectors to form a linearly dependent subset. Denote the indices of the vectors in this smallest dependent grouping as $D$. Then for some nonntrivial set $\{c_i | i \in D\}$,

$$\sum_{i \in D} c_i \mathbf{v}_i = \mathbf{0}.$$

We can also multiply both sides by $A$:

$$A \sum_{i \in D} c_i \mathbf{v}_i = \sum_{i \in D} c_i \lambda_i \mathbf{v}_i = \mathbf{0}$$

Next, choose any index $j \in D$, and multiply the first equation by $\lambda_j$:

$$\lambda_j \sum_{i \in D} c_i \mathbf{v}_i = \sum_{i \in D} c_i \lambda_j \mathbf{v}_i = \mathbf{0}$$

Finally, we can subtract the above two equations:

$$\sum_{i \in D} c_i \lambda_i \mathbf{v}_i - \sum_{i \in D} c_i \lambda_j \mathbf{v}_i = \sum_{i \in D} c_i (\lambda_i - \lambda_j) \mathbf{v}_i = \sum_{i \in D, \neq j} c_i (\lambda_i - \lambda_j) \mathbf{v}_i = \mathbf{0}$$

In the summation above, $\lambda_j - \lambda_j = 0$, so we can sum over only the $m-1$ indices in $D$ such that $i \neq j$. Because we're considering eigenvectors of different eigenvalues, $\lambda_i - \lambda_j \neq 0$ for all $i \neq j$. We also know that since all the eigenvectors are nonzero, the original linearly dependent combination had at least two $c_i$ nonzero, and so at least one $c_i$ above is nonzero. In conclusion, $c_i(\lambda_i - \lambda_j) \neq 0$ for some $i$, and we've achieved a nontrivial linearly dependent combination of $m - 1$ vectors.

This finding contradicts the original premise that $m$ is the minimum number of eigenvectors in $S$ to form a linearly dependent set. Therefore, the original premise that some number $m \geq 2$ of vectors in $S$ is the minimum number of dependent vectors implies that it is false. And one single eigenvector cannot possibly form a linearly dependent set, so we have a contradiction to the possibility of any number of eigenvectors corresponding to distinct eigenvalues being linearly dependent. □

This proves that eigenvalues corresponding to different eigenvalues of a matrix $A$ are independent. However, sometimes the eigenspace corresponding to one eigenvalue of $A$ is multi-dimensional, in that multiple basis eigenvectors span the solution set to $A\mathbf{x} = \lambda\mathbf{x} \iff (A - \lambda I)\mathbf{x} = \mathbf{0}$. In this case, one must simply choose a basis for the solution space of that equation, and the resulting set will, by definition of a basis, be linearly independent and span all eigenvectors corresponding to some $\lambda$. This trivial addendum, coupled with the above proof, establishes that any and all sets of eigenvectors corresponding to an $n \times n$ matrix $A$ are certainly independent, provided that one chooses a (linearly independent, spanning) basis for each eigenspace of $A$.

In practice, eigenvalues and eigenvectors come up fairly often. For instance, many physical processes in engineering can be "linearized" through matrix multiplication. A simple problem is to determine what input will amplify or dampen some input vector. Equivalently, what eigenvector of $A$ will appropriately be dampened in magnitude ($\lambda \in (-1, 1)$) or amplified ($\lambda > 1$) by the system?

### 5.3.1 Diagonalization

**Definition 13** (diagonalization of a matrix $A$)**.** *To **diagonalize** a square, $n \times n$ matrix $A$ is to write it in the form $A = PDP^{-1}$, with $D$ a diagonal matrix.*

One significance of diagonalization is that it allows a much more computationally efficient way to compute large matrix powers, such as $A^{100}$. Each multiplication, a $P$ and $P^{-1}$ matrix will cancel out, and you just need to compute $A^{100} = PD^{100}P^{-1}$. Calculating $A^{100}$ manually is much more computationally complex and subject to round-off inaccuracies stemming from the limitations of floating point numbers.

A natural question arises: when can we diagonalize a square matrix $A$?

**Theorem 12.** *Suppose $A \in \mathbb{R}^{n\times n}$ has (linearly independent) eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, arranged in an arbitrary order, with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. Additionally, let $P = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}$ and $D = diag(\lambda_1, \ldots, \lambda_n)$. Then $A$ is diagonalizable, because this choice of $P$ and $D$ diagonalizes $A$.*

*Proof.* First, I show that $AP = PD$:

$$\text{Left-hand side } = A\begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix} = \begin{bmatrix}A\mathbf{v}_1 & \cdots & A\mathbf{v}_n\end{bmatrix} = \begin{bmatrix}\lambda_1\mathbf{v}_1 & \cdots & \lambda_n\mathbf{v}_n\end{bmatrix}$$
$$\text{Right-hand side } = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix} diag(\lambda_1, \ldots, \lambda_n) = \begin{bmatrix}\lambda_1\mathbf{v}_1 & \cdots & \lambda_n\mathbf{v}_n\end{bmatrix}$$

By the invertible matrix theorem, $P$ is invertible because it contains distinct eigenvectors, which will always be linearly independent. Therefore, we can compute $P^{-1}$ and right-multiply the above equation to obtain

$$A = PDP^{-1}.$$

□

#### Implications for what forms $P$ and $D$ can take

The above forms for $P$ and $D$ are not necessarily unique. One can order the eigenvectors in $P$ and eigenvalues in $D$ in a different order, ensuring that the $i$th column of $P$ corresponds to the $i$th of $D$. With a different ordering of columns, $P$ will still be invertible, and $AP = PD$, so this choice of $P$ and $D$ would also work in implying $A = PDP^{-1}$. Additionally, by the above proof, any pair of a column in $P$ and a diagonal value in $D$, which I shall denote $\mathbf{p}$ and $\lambda$, must satisfy $A\mathbf{p} = \lambda\mathbf{p}$ if the diagonalization is valid, i.e. $A = PDP^{-1}$ and $AP = PD$. So the columns of $p$ in a diagonalization must be eigenvectors of $A$, and the corresponding diagonal entries of $D$ must be those eigenvectors' eigenvalues.

**Diagonalization theorems (continued)**

**Theorem 13.** *An $n \times n$ matrix $A$ which does not have $n$ linearly independent eigenvectors cannot be diagonalized.*

*Proof.* Suppose instead that $A$ can be diagonalized, despite having $< n$ linearly independent eigenvectors. Then we can show a contradiction:

$$A = PDP^{-1}$$
$$AP = PD = Pdiag(d_1, \ldots, d_n)$$
$$\begin{bmatrix} A\mathbf{p}_1 & \cdots & A\mathbf{p}_n \end{bmatrix} = \begin{bmatrix} d_1\mathbf{p}_1 & \cdots & d_n\mathbf{p}_n \end{bmatrix}$$

Since $P$ is invertible, by the invertible matrix theorem, all its columns $\{\mathbf{p}_i\}$ must be linearly independent. Therefore, we have found $n$ linearly independent eigenvectors of $A$, which contradict the original assumption that $A$ has fewer than $n$ yet is diagonalizable. □

**Theorem 14.** *If and only if a matrix $A \in \mathbb{R}^{n \times n}$ has $n$ linearly independent eigenvectors can it be diagonalized as $A = PDP^{-1}$.*

*Proof.* This theorem shortly summarizes and follows from the previous two theorems. □

## 5.4 The inner product

**Definition 14** (an inner product over a vector space $V$)**.** *An inner product over $V$ is a function $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ which satisfies the following for all $\mathbf{u}, \mathbf{v} \in V$ and $c \in \mathbb{R}$:*

1. $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$.
2. $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$.
3. $\langle c\mathbf{u}, \mathbf{v} \rangle = c\langle \mathbf{u}, \mathbf{v} \rangle$.
4. $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$.
5. $\langle \mathbf{u}, \mathbf{u} \rangle = 0$ *if and only if* $\mathbf{u} = \mathbf{0}$.

*When one associates an inner product with a vector space $V$, $V$ is sometimes called an* ***inner product space****.*

**Definition 15** (the dot product)**.** *The most common inner product, the* ***dot product****, is defined over* $\mathbb{R}^n$ *as*

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}.$$

*The inner product is often written as* $\mathbf{u} \cdot \mathbf{v}$ *instead of* $\langle \mathbf{u}, \mathbf{v} \rangle$.

**Definition 16** (the $p$-norm)**.** *The p-norm is a function of a vector in* $\mathbb{R}^n$ *defined as follows:*

$$\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^{n} x_i^p}$$

You may recognize that the 2-norm is actually Euclidean distance in 2- or 3-dimensional space, as given by the Pythagorean theorem. When someone writes just $\|\mathbf{x}\|$, not $\|\mathbf{x}\|_p$, they typically mean the 2-norm; people drop the 2 because it is the most common norm.

### 5.4.1 The Cauchy-Schwarz inequality

**Theorem 15** (the Cauchy-Schwarz inequality)**.** *For any vectors* $\mathbf{u}, \mathbf{v}$ *of an inner product space, it must be true, by consequence of the properties of the inner product, that*

$$\langle \mathbf{u}, \mathbf{v} \rangle^2 \leq \langle \mathbf{u}, \mathbf{u} \rangle \cdot \langle \mathbf{v}, \mathbf{v} \rangle.$$

*Proof.* Say you give me any two vectors $\mathbf{u}, \mathbf{v}$ in an inner product space. In the trivial case in which $\mathbf{v} = \mathbf{0}$, the above is satisfied as an equality. But what if $\mathbf{v} \neq \mathbf{0}$? Define $\mathbf{z} = \mathbf{u} - \langle \mathbf{u}, \mathbf{v} \rangle \mathbf{v} / \langle \mathbf{v}, \mathbf{v} \rangle$. Then $\langle \mathbf{z}, \mathbf{v} \rangle = 0$, because

$$\langle \mathbf{z}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle - \frac{\langle \mathbf{u}, \mathbf{v} \rangle \langle \mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} = \langle \mathbf{u}, \mathbf{v} \rangle - \langle \mathbf{u}, \mathbf{v} \rangle = 0.$$

Since $\mathbf{u} = \mathbf{z} + \langle \mathbf{u}, \mathbf{v} \rangle \mathbf{v} / \langle \mathbf{v}, \mathbf{v} \rangle$,

$$\begin{aligned}
\langle \mathbf{u}, \mathbf{u} \rangle &= \langle \mathbf{z} + \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{v}, \mathbf{z} + \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{v} \rangle \\
&= \langle \mathbf{z}, \mathbf{z} \rangle + 2 \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \langle \mathbf{z}, \mathbf{v} \rangle + \left( \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \right)^2 \langle \mathbf{v}, \mathbf{v} \rangle \\
&= \langle \mathbf{z}, \mathbf{z} \rangle + \left( \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \right)^2 \langle \mathbf{v}, \mathbf{v} \rangle = \langle \mathbf{z}, \mathbf{z} \rangle + \frac{\langle \mathbf{u}, \mathbf{v} \rangle^2}{\langle \mathbf{v}, \mathbf{v} \rangle}
\end{aligned}$$

Recall our assumption that $\langle \mathbf{v}, \mathbf{v} \rangle > 0$, and also note that $\langle \mathbf{z}, \mathbf{z} \rangle \geq 0$.

$$\langle \mathbf{u}, \mathbf{u} \rangle \geq \frac{\langle \mathbf{u}, \mathbf{v} \rangle^2}{\langle \mathbf{v}, \mathbf{v} \rangle}$$
$$\langle \mathbf{u}, \mathbf{u} \rangle \langle \mathbf{v}, \mathbf{v} \rangle \geq \langle \mathbf{u}, \mathbf{v} \rangle^2$$

□

# Chapter 6

# Orthogonality and least squares

**Definition 17** (orthogonality)**.** *Consider two vectors* $\mathbf{u}, \mathbf{v}$ *in any inner product space.* $\mathbf{u}$ *and* $\mathbf{v}$ *are orthogonal if*

$$\langle \mathbf{u}, \mathbf{v} \rangle = 0.$$

This definition naturally extends to the notion of an **orthogonal set**: a set of vectors, every pairing of which is orthogonal.

**Theorem 16.** *If a vector* $\mathbf{y}$ *in an inner product space* $W$ *is a linear combination of orthogonal vectors* $\mathbf{u}_1, \ldots, \mathbf{u}_n \in W$*, in that*

$$\mathbf{y} = c_1\mathbf{u}_1 + \cdots + c_n\mathbf{u}_n,$$

*then* $c_i = \frac{\langle \mathbf{y}, \mathbf{u}_i \rangle}{\langle \mathbf{u}_i, \mathbf{u}_i \rangle}$.

*Proof.* The key assumption is that all these vectors are in an inner product space, for example $\mathbb{R}^n$ with the dot product as an inner product. The following logic follows.

$$\langle \mathbf{y}, \mathbf{u}_i \rangle = \sum_{j=1}^{n} c_j \langle \mathbf{u}_j, \mathbf{u}_i \rangle = c_i \langle \mathbf{u}_i, \mathbf{u}_i \rangle$$

$$c_i = \frac{\langle \mathbf{y}, \mathbf{u}_i \rangle}{\langle \mathbf{u}_i, \mathbf{u}_i \rangle}$$

□

This theorem indicates that an orthogonal set of non-zero vectors is linearly independent.

## 6.1 The orthogonal decomposition theorem

**Theorem 17** (the orthogonal decomposition theorem)**.** *Consider $W$, a subspace of an inner product space $I$ such as $\mathbb{R}^n$, with the orthogonal basis $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$. Every $\mathbf{y} \in I$ can be expressed uniquely in the form $\hat{\mathbf{y}} + \mathbf{z}$, with $\hat{\mathbf{y}} \in W$ and $\mathbf{z}$ orthogonal to every vector in $W$. $\hat{\mathbf{y}}$ will be equal to*

$$\hat{\mathbf{y}} = \sum_{i=1}^{p} \frac{\langle \mathbf{y}, \mathbf{u}_i \rangle}{\langle \mathbf{u}_i, \mathbf{u}_i \rangle} \mathbf{u}_i.$$

*This definition of $\hat{\mathbf{y}}$ is the **projection** of $\mathbf{y}$ onto $W$,* $\text{proj}_W \mathbf{y}$.

*Proof.* Suppose there exist $\hat{\mathbf{y}}$ and $\mathbf{z}$ such that $\mathbf{y} = \hat{\mathbf{y}} + \mathbf{z}$. Then $\hat{\mathbf{y}} = \sum_i c_i \mathbf{u}_i$ since $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ form a (spanning) basis. Because $\langle \mathbf{u}_i, \mathbf{u}_{j \neq i} \rangle = 0$ and $\mathbf{z}$, for all $i$

$$\langle \mathbf{y}, \mathbf{u}_i \rangle = c_i \langle \mathbf{u}_i, \mathbf{u}_i \rangle \implies c_i = \frac{\langle \mathbf{y}, \mathbf{u}_i \rangle}{\langle \mathbf{u}_i, \mathbf{u}_i \rangle}.$$

This can be the only form for $\hat{\mathbf{y}}$, and this form also exists for all $\mathbf{y}$.

Additionally, $\mathbf{z} = \mathbf{y} - \hat{\mathbf{y}}$ is indeed orthogonal to all of $W$ in all cases. One can manually take the inner product of $\mathbf{z}$ with any $\mathbf{u}_i$ to establish this result. □

## 6.2 Finding orthogonal bases: the Gram-Schmidt process

Many times, one wants to obtain an orthogonal basis for an inner product space. Luckily, the Gram-Schmidt process helps solve this problem, if you already have a non-orthogonal basis $\{\mathbf{v}_1, \dots, \mathbf{v}_q\}$ of the inner product space.

**Lemma 3.** *Say you have a set of independent vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_q\}$. Then subtracting from $\mathbf{v}_j$ a linear combination of all $\mathbf{v}_i$ such that $i \neq j$ will yield a linearly independent set:*

$$\{\mathbf{w}_1, \dots, \mathbf{w}_q\} = \{\mathbf{v}_1, \dots, \mathbf{v}_j - \sum_{i \neq j} c_i \mathbf{v}_i, \dots, \mathbf{v}_q\}$$

*Proof.* Suppose the contradiction, that the new set is not linearly independent. Then for some constants $d_i$, at least one of which is nonzero,

$$\sum_{i=1}^{q} d_i \mathbf{w}_i = d_j \mathbf{v}_j + \sum_{i \neq j} (d_i - d_j c_i) \mathbf{v}_i = \mathbf{0}.$$

In this scenario, either $d_j = 0$ or $d_j \neq 0$. In the first case, we have a linear combination of $\mathbf{v}_i$ equal to $\mathbf{0}$: $\sum_{i\neq j}(d_i - d_j c_i)\mathbf{v}_i = \sum_{i\neq j} d_i\mathbf{v}_i = \mathbf{0}$, and at least one of these $d_i \neq 0$ by assumption that this is a nontrivial solution to the above equation.

In the second case that $d_j \neq 0$, we can rearrange the equation to write $\mathbf{v}_j = \sum_{i\neq j}(d_j c_i - d_i)/d_j\mathbf{v}_i$. $\mathbf{v}_j \neq 0$ since the original set is linearly independent; therefore this linear combination is nontrivial and can be rearranged to show that some linear combination of $\mathbf{v}_i$'s is $\mathbf{0}$. But the $\mathbf{v}_i$'s are linearly independent, so this is impossible. □

**Theorem 18.** *Suppose you have a basis of q vectors* $\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$*, not necessarily orthogonal to each other. Then subtracting from one of these vectors* $\mathbf{v}_j$ *its projection onto some other subset of the vectors S will yield an equivalent (linearly independent) basis, with the same spanning set, but with all* $\mathbf{v}_{i\in S}$ *orthogonal to* $\mathbf{v}_j$*.*

*Proof.* A projection is, by definition, a linear combination of the vectors being projected onto. By the above lemma, the new set must then still be linearly independent.

Additionally, note the form of the new vector, $\mathbf{u}_j = \mathbf{v}_j - \text{proj}_{\{\text{all } \mathbf{v}_i | i\in S\}}\mathbf{v}_j = \mathbf{v}_j - \sum_{i\in S}\langle\mathbf{v}_j, \mathbf{v}_i\rangle\mathbf{v}_i/\langle\mathbf{v}_i, \mathbf{v}_i\rangle$. This vector must be orthogonal to $\mathbf{v}_i$ for all $i \in S$: if you expand out the inner product $\langle\mathbf{u}_j, \mathbf{v}_{i\in S}\rangle$, you will see that it goes to 0. This result was also indicated in the derivation of the orthogonal decomposition theorem.

And now, the final question to prove. Is the span of $\{\mathbf{v}_1, \ldots, \mathbf{u}_j, \ldots, \mathbf{v}_q\}$ the same as the span of $\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$? If we can prove that, then the set is not only linearly independent, but also spans the same set as $\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$; it is a new basis, with the special property that the $j$th element is orthogonal to all the others. To prove this, first I will show that any vector reachable through a linear combination of $\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$ (in the span of these vectors) is reachable through a linear combination of $\{\mathbf{v}_1, \ldots, \mathbf{u}_j, \ldots, \mathbf{v}_q\}$. Then, I will prove any vector reachable through the latter basis is be reachable through the former. With these two directions established, the spanning sets will be determined equal.

Suppose $\mathbf{y} = \sum_{i=1}^{q} c_i\mathbf{v}_i$. I would like coordinates $d_i$ to write $\mathbf{y}$ in the new basis, i.e. constants such that

$$\begin{aligned}\mathbf{y} = \sum_{i\neq j} d_i\mathbf{v}_i + d_j\mathbf{u}_j &= \sum_{i\neq j} d_i\mathbf{v}_i + d_j\left(\mathbf{v}_j - \sum_{i\in S}\frac{\langle\mathbf{v}_j, \mathbf{v}_i\rangle}{\langle\mathbf{v}_i, \mathbf{v}_i\rangle}\mathbf{v}_i\right)\\ &= d_j\mathbf{v}_j + \sum_{i\in S}\left(d_i - d_j\frac{\langle\mathbf{v}_j, \mathbf{v}_i\rangle}{\langle\mathbf{v}_i, \mathbf{v}_i\rangle}\right)\mathbf{v}_i + \sum_{i\neq j, \notin S} d_i\mathbf{v}_i.\end{aligned}$$

This can be achieved by setting $d_i$'s such that the coefficient on each $\mathbf{v}_i$ equals $c_i$. For starters, $d_j = c_j$, and following that, $d_{i \in S} = c_i + d_j \langle \mathbf{v}_j, \mathbf{v}_i \rangle / \langle \mathbf{v}_i, \mathbf{v}_i \rangle$; for the rest of the $\mathbf{v}_i$'s, set $d_i = c_i$.

Now, what about the reverse direction, writing coordinates in the old basis ($c_i$'s) from coordinates in the new basis ($d_i$'s)? Again, let's utilize the bottom-most above expression. For each for $\mathbf{y}$ in terms of $d_i$. It reveals that we can set $c_q = d_q$, $c_{i \in S} = d_i - d_j \langle \mathbf{v}_j, \mathbf{v}_i \rangle / \langle \mathbf{v}_i, \mathbf{v}_i \rangle$, and $c_{i \neq j, \notin S} = d_i$. □

Finally, we have established the necessary theorem to realize the Gram-Schmidt method for "orthogonalizing" a basis, or converting from any basis $\mathcal{C}$ to an orthogonal basis:

1. Start with the first vector from $\mathcal{C} = \{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$. Call this vector $\mathbf{u}_1$; it by itself forms an orthogonal basis.

    - You can actually work with the vectors in any order. For the sake of clarity, I will go from vector $\mathbf{v}_1$ up to $\mathbf{v}_q$.

2. **Orthogonalize the $i$th vector**: One by one, replace in $\mathcal{C}$ the $i$th vector $\mathbf{v}_i$ with $\mathbf{v}_i$ minus its projection onto all $\mathbf{u}_j$ such that $j < i$ (in math notation, $\mathbf{v}_i - \text{proj}_{\mathbf{u}_1, \ldots, \mathbf{u}_{i-1}} \mathbf{v}_i$).

    - Now, the basis looks like $\mathcal{C} = \{\mathbf{u}_1, \ldots, \mathbf{u}_i, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_q\}$.
    - The new vector $\mathbf{u}_i$ cannot be $\mathbf{0}$. The above lemma indicates the new set must be linearly independent.
    - The above theorem shows that $\{\mathbf{u}_1, \ldots, \mathbf{u}_{i-1}, \mathbf{v}_i\}$ has the same span as $\{\mathbf{u}_1, \ldots, \mathbf{u}_i\}$. Therefore, $\mathcal{C}$ has the same span before and after this step.
        - Since this step always preserves linear independence and the span of $\mathcal{C}$, the new bases discovered will be equivalent.
    - However, the above theorem also establishes that the new vector, $\mathbf{u}_i$, will be orthogonal to all the preceding orthogonalized vectors, $\mathbf{u}_1$ through $\mathbf{u}_{i-1}$. Since $\{\mathbf{u}_1, \ldots, \mathbf{u}_i\}$ forms an orthogonal set, the final set of vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_q\}$ will be completely orthogonal, yet still an equivalent basis as noted before.

3. You are done! The basis $\mathcal{C}$ is equivalent to the original basis, yet orthogonal.

- Note that based on the order you orthogonalize the vectors (steps 1 and 2), this process may yield different bases. Still, you can be sure the vectors the algorithm produces are orthogonal and linearly independent, and span the same set as the original basis.

## 6.3 Best approximation and least-squares

### 6.3.1 The best approximation theorem

**Theorem 19** (the best approximation theorem)**.** *Let $W$ be a subspace of $\mathbb{R}^n$, and $\mathbf{y}$ be any vector in $\mathbb{R}^n$. Then letting $\mathbf{v} = \hat{\mathbf{y}}$, according to the definition of $\hat{\mathbf{y}}$ above, minimizes $\|\mathbf{y} - \mathbf{v}\|_2$. $\hat{\mathbf{y}}$ is referred to as the "best approximation."*

*Proof.* One can rewrite $\|\mathbf{y} - \mathbf{v}\|$ as

$$\sqrt{\sum_{i=1}^{n}(y_i - v_i)^2}.$$

Since $\sqrt{x}$ is a strictly increasing function, we can reduce the problem to choosing $\mathbf{v}$ which minimizes the loss function

$$l(\mathbf{v}) = \sum_{i=1}^{n}(y_i - v_i)^2.$$

For reasons outside of this class, this turns out to be the sum of convex functions and thus convex, and so any $\mathbf{v}$ such that $\nabla l(\mathbf{v}) = \mathbf{0}$ will minimize the loss function. One can plug in $\hat{\mathbf{y}}$ and see that it satisfies $\nabla l(\mathbf{v}) = \mathbf{0}$. □

### 6.3.2 Least-squares problems

A **least-squares problem** concerns finding a vector $\mathbf{x} \in \mathbb{R}^m$ such that $\|\mathbf{b} - A\mathbf{x}\|_2$ is minimized, with $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. (The name "least-squares" arises from the fact that $\|\mathbf{b} - A\mathbf{x}\|$ is the square root of a sum of squares.) In general, this problem arises when $A\mathbf{x} = \mathbf{b}$ has no exact solution, and part of the problem's popularity is that it has an elegant, easy to compute solution, as we will soon find.

Perhaps you see a way to solve this problem. The vectors produced by the matrix multiplication $A\mathbf{x}$ are defined to be the column space of $A$, which is a subspace

of $\mathbb{R}^n$. By the best approximation theorem above, the vector $\hat{\mathbf{b}} = \text{proj}_{\text{Col}A}\mathbf{b}$ is the vector in this subspace, Col$A$, which minimizes $\|\mathbf{b} - \hat{\mathbf{b}}\|$. Since by definition of a column space, $A\hat{\mathbf{x}} = \hat{\mathbf{b}}$ has at least solution, we can solve this equation for $\hat{\mathbf{x}}$, and be done. But there is an issue: I defined the projection $\text{proj}_W\mathbf{x}$ in terms of an arbitrary orthogonal basis of $W$, and we don't necessarily have an orthogonal basis of Col$A$, at least not unless we perform Gram-Schmidt.

Is there an easier method than the tedious Gram-Schmidt process to find a suitable $\hat{\mathbf{x}}$? The orthogonal decomposition theorem reveals that given any vector $\mathbf{b}$ in $\mathbb{R}^n$, there exists only one vector $\hat{\mathbf{b}} \in$ ColA, $\text{proj}_{\text{Col}A}\mathbf{b}$, for which $\mathbf{b} - \hat{\mathbf{b}}$ is orthogonal to all of Col$A$. For that $\hat{\mathbf{b}}$,

$$A^T(\mathbf{b} - \hat{\mathbf{b}}) = \mathbf{0} \iff A^T\hat{\mathbf{b}} = A^T\mathbf{b}.$$

The above equations must be true of only that one solution $\hat{\mathbf{b}}$ — they are satisfied iff $\mathbf{b} - \hat{\mathbf{b}}$ is orthogonal to every column of $A$ and thus every vector in Col$A$.

As noted originally, we actually want $\hat{\mathbf{x}}$ such that $A\hat{\mathbf{x}} = \hat{\mathbf{b}}$. The derivation for this problem is almost identical:

$$A^T(\mathbf{b} - A\hat{\mathbf{x}}) = \mathbf{0} \iff A^TA\hat{\mathbf{x}} = A^T\mathbf{b}$$

Here we see what is essentially a classic matrix equation, which must have at least one solution, because $\hat{\mathbf{b}}$ exists, by the orthogonal decomposition theorem.

# Chapter 7

# Advanced matrix applications

## 7.1 Symmetric matrices

**Definition 18** (a symmetric matrix)**.** *A symmetric matrix is a matrix $A$ such that $A^T = A$.*

For example, this matrix is symmetric:

$$\begin{bmatrix} -3 & 4 & 3 \\ 4 & 6 & 5 \\ 3 & 5 & 7 \end{bmatrix}$$

Notice the defining symmetry about the matrix's diagonal. Based on the above definition, it is apparent that symmetric matrices are always square, or else $A^T$ and $A$ would be of different dimension.

**Theorem 20.** *Consider the vector space $\mathbb{R}^n$. If $A$ is symmetric, any two eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ corresponding to different eigenvalues of $A$ are orthogonal.*

*Proof.* We'd like to show that under the above assumptions, $\mathbf{v}_1^T \mathbf{v}_2 = 0$:

$$\begin{aligned} \lambda_1 \mathbf{v}_1^T \mathbf{v}_2 &= (A\mathbf{v}_1)^T \mathbf{v}_2 \\ &= \mathbf{v}_1^T A^T \mathbf{v}_2 \\ &= \mathbf{v}_1^T A \mathbf{v}_2 \\ &= \lambda_2 \mathbf{v}_1^T \mathbf{v}_2 \end{aligned}$$

By assumption that $\lambda_1 \neq \lambda_2$, we must have that $\mathbf{v}_1^T \mathbf{v}_2 = \mathbf{v}_1 \cdot \mathbf{v}_2 = 0$. □

**Theorem 21.** *An $n \times n$ symmetric matrix $A$ has $n$ linearly independent eigenvectors. Therefore, it is diagonalizable in that $A = PDP^{-1}$. It is also **orthogonally diagonalizable**, because by normalizing the orthogonal eigenvectors in $P$, we get that $P^{-1} = P^T$ and $A = PDP^T$.*

*Proof.* From the prior theorem, we already know any two eigenvectors of different eigenvalues will be orthogonal. Additionally, if an eigenspace has more than one dimension, then we can find an orthogonal basis for it so that every single eigenvector is orthogonal to all others.

First, I will consider the case that $A$ is symmetric, and show that it then must have $n$ eigenvalues. □

**Theorem 22.** *An $n \times n$ symmetric matrix $A$ has the following four properties:*

1. *$A$ has $n$ eigenvalues, if you count $i$ eigenvalues for a dimension-$i$ eigenspace corresponding to a certain one eigenvalue.*
2. *The dimension of the eigenspace for each eigenvalue $c$ equals the multiplicity of $c$ by the characteristic (polynomial) equation $\det(A - \lambda I) = 0$.*
    - *Multiplicity is the number of times $(\lambda - c)$ can be factored out of the polynomial equation resulting from $\det(A - \lambda I) = 0$.*
3. *The first theorem in this section: in other words, the eigenspaces of $A$ are mutually orthogonal.*
4. *$A$ is orthogonally diagonalizable, as the second theorem in this section states.*

### 7.1.1 The Cholesky decomposition

The Cholesky decomposition factors a symmetric, positive semidefinite $n \times n$ matrix $A$ into the form $A = LL^T$, with $L$ a lower-triangular matrix.

## 7.2 Quadratic forms

**Definition 19.** *A quadratic form is a function from $\mathbb{R}^n$ to $\mathbb{R}$ which can be defined as*

$$Q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x}$$

*for some matrix $A \in \mathbb{R}^{n \times n}$.*

Let's examine this function more closely. For the sake of clarity, how can we rewrite a formula for $Q(\mathbf{x})$ without matrix notation? $Q$ does, after all, return a single real number.

$$Q(\mathbf{x}) = \sum_{i=1}^{n} x_i (A\mathbf{x})_i = \sum_{i=1}^{n} x_i \left( \sum_{j=1}^{n} a_{ij} x_j \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

As we can see, $Q(\mathbf{x})$ can be interpreted as the sum, for each entry $a_{ij}$ of $A$, of $a_{ij}$ times the $i$th and $j$th variables of $\mathbf{x}$.

**Theorem 23.** *A quadratic form $Q$ is positive definite, i.e. $Q(\mathbf{x}) > 0$ always, iff $A$ has strictly positive eigenvalues. It is positive semidefinite ($Q(\mathbf{x}) \geq 0$) iff all its eigenvalues are nonnegative. Similar but opposite results apply to determine if $Q$ is negative definite or semidefinite.*

## 7.3 The singular value decomposition (SVD)

Not all matrices are diagonalizable. However, every matrix can be written in a special form called the **singular value decomposition**. Why is that?

**Definition 20** (the singular value decomposition)**.** *A singular value decomposition of an $m \times n$ matrix $A$ is $A = U\Sigma V^T$. Here, $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthonormal matrices, so that $U^T U = V^T V = I$. $\Sigma$ a diagonal matrix in $\mathbb{R}^{m \times n}$, whose diagonal entries are positive and arranged down $\Sigma$'s diagonal in decreasing order.*

**Theorem 24.** *Every $m \times n$ matrix $A$ can be written in SVD form, $A = U\Sigma V^T$. (This theorem does not claim this decomposition is unique, and it is not!)*

*Proof.* Assume only that such a decomposition exists. Then by the transpose and associativity properties of matrix multiplication,

$$AA^T = U\Sigma V^T \left(U\Sigma V^T\right)^T = U\Sigma \left(V^T V\right) \Sigma^T U^T = U \left(\Sigma\Sigma^T\right) U^T,$$

with $\Sigma\Sigma^T$ above still a diagonal matrix. The same logic applies for $A^T A$: if a singular value decomposition exists, then

$$A^T A = \left(U\Sigma V^T\right)^T U\Sigma V^T = V\Sigma^T \left(U^T U\right) \Sigma V^T = V \left(\Sigma^T \Sigma\right) V^T.$$

$\Sigma^T\Sigma$ and $\Sigma\Sigma^T$, by definition of the SVD, are $n \times n$ and $m \times m$ diagonal matrices, respectively. They are the same, except that one of them may be larger than the other and padded with zeroes, since $\Sigma$ can have nonzero elements only along its diagonal.

Now, let us continue with the assumption that the SVD exists. Recall that every symmetric square matrix $M$ can be diagonalized in the form $M = PDP^T$, with $D$ diagonal, but $D$ can only contain along its diagonal the set of eigenvalues of $M$, and $P$ then must contain the corresponding normalized eigenvectors of $M$, so that $PP^T = I$. Therefore, $\Sigma\Sigma^T$ and $\Sigma^T\Sigma$ must contain the eigenvalues of $AA^T$ and $A^TA$; by implication, both $AA^T$ and $A^TA$ have the same nonzero eigenvalues since $\Sigma\Sigma^T$ and $\Sigma^T\Sigma$ have the same nonzero entries. These findings, coupled with the SVD's definition that imposes the entries of $\Sigma$ are nonnegative and decreasing, implies $\Sigma$ must contain along its diagonal the positive square roots of these eigenvalues of $AA^T$ or $A^TA$ (these square roots are called **singular values**), in decreasing order. Likewise, $U$ must contain the corresponding orthonormal eigenvectors of $AA^T$, in the corresponding order, as must $V$ for $A^TA$.

Thus we have determined exactly what form $\Sigma$ will take, and narrowed down what forms $U$ and $V$ can take, provided that an SVD does exist. At last, let us stop assuming that the SVD exists, and try to prove that it actually does always exist. If we can find matrices $U$, $\Sigma$, and $V$ meeting the conventions laid out in the definition of the SVD, and such that $AV = U\Sigma$, then by the orthonormality of $V$, $VV^T = I$ and $AVV^T = A = U\Sigma V^T$. By the same logic, we can also prove the existence of the SVD by finding such matrices satisfying $A^TU = V\Sigma^T$. I will explain the first approach, but the second approach is essentially the same, and may be more computationally efficient, depending on which of $m$ and $n$ is larger.

Since $AA^T \in \mathbb{R}^{m\times m}$ is symmetric, is has $m$ eigenvalue-eigenvector pairs, with all pairs of eigenvectors orthogonal. One could arrange these pairs in order of descending eigenvalue: $(\lambda_1, \mathbf{u}_1), \ldots, (\lambda_m, \mathbf{u}_m)$, and then define $U = \begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_m\end{bmatrix}$ and populate the main diagonal of $\Sigma$ with $\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_{min(m,n)}}$. These choices are inspired by my original consideration of what forms $U$, $\Sigma$, and $V$ can take if an SVD does exist.

Lastly, let us consider solving for $V$ in the equation

$$AV = \begin{bmatrix}\sqrt{\lambda_1}\mathbf{u}_1 & \cdots & \sqrt{\lambda_{min(m,n)}}\mathbf{u}_{min(m,n)} & \cdots & \sqrt{\lambda_n}\mathbf{u}_n\end{bmatrix}.$$

Some of the above $\mathbf{u}_i$ are multiplied by 0, and others by nonzero constants. Let's say we solve for $V$ using standard row reduction or other techniques. A solution will always exist because $\mathbf{u}_i$ is an eigenvector of $AA^T$, and thus in the span of $A$.

Furthermore, conveniently, whatever $\mathbf{v}_i$ we find which correspond to $\mathbf{u}_i$ multiplied by nonzero constants are pairwise orthogonal because for $i \neq j$, both $i$ and $j$ corresponding to nonzero singular values,

$$\mathbf{u}_i^T \mathbf{u}_j = 0 \implies 0 = (A\mathbf{v}_i)^T A\mathbf{v}_j = \mathbf{v}_i^T A^T A \mathbf{v}_j = \lambda_j \mathbf{v}_i^T \mathbf{v}_j.$$

If only $j$, but not $i$, corresponds to a nonzero singular value ($\lambda_j \neq 0$), the above still holds. But what about the third and final case, in which both $i$ and $j$ correspond to singular values of 0? In that case, $\mathbf{v}_i$ is a solution to the equation $A\mathbf{v}_i = 0\mathbf{u}_i = \mathbf{0}$, and it is even possible that we found $\mathbf{v}_i = \mathbf{0}$, which clearly is not a normal vector. So how do we find these $\mathbf{v}_i$? Well, there are $l$ eigenvectors of $A^T A$ in $V$ corresponding to singular values of 0. Eigenvectors corresponding to the same eigenvalue are, by convention, always independent, and so the dimension of $Null(A)$ is at least $l$. It is also at most $l$ because we have already found $n - l$ orthogonal (linearly independent), nonzero vectors which $A$ maps to, so rank $A \geq n - l$ and recall the rank theorem: rank $A$ + dim $Null(A) = n$. In conclusion, for these last $l$ problems $A\mathbf{v}_i = \mathbf{0}$, simply find an orthonormal basis $\{\mathbf{v}_i\}$ for $Null(A)$, perhaps by Gram-Schmidt.

And we are done. I have proposed choices for $U$ and $\Sigma$, based on the diagonalization of $AA^T$, which ensure $\Sigma$ is a diagonal, decreasing matrix and $U$ is orthonormal, due to properties of the eigenvectors of a symmetric matrix. Then, I have shown that we can simply solve $A\mathbf{v}_i = \sqrt{\lambda_i}\mathbf{u}_i$, or for $i$ such that $\lambda_i = 0$, simply find an orthogonal basis of $Null(A)$, and the resulting $V$ will be orthonormal. The properties of $U$, $\Sigma$, and $V$ which define the SVD are upheld, and it is true that $AV = U\Sigma$ and thus $A = U\Sigma V^T$. □