The 18.C06 Master Document

Jack D.V. Carson

jdcarson@mit.edu

Contents

1 Linear Algebra

This document assumes you already have a reasonable familiarity with the most basic ideas of linear algebra (i.e. What is a matrix? What is matrix multiplication? Why would one be compelled to multiply a matrix?). From hence, we shall begin with the beginning.

1.1 The Basics

- In linear algebra, we most often want to express some situation in terms of a formula $Ax = b$ for a matrix A, and vectors x, b. Then, once we have expressed them in this form, we wish to solve for the values of x that give b. The simplest way to do this is with **Gaussian Elimination** wherein rows are added to one another or multiplied by scalar constants to give a **diagonal matrix**. This algorithm always goes columnwise. I.e. start from the first column and try and clear everything such that everything below the pivot is 0. This may not always be the fastest approach but it is consistent. From this diagonal matrix, it is simple to **backsubstitute** to get the complete values of x.

$$\left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 11 & 6 & 35 \end{array}\right] \xrightarrow[\;]{\substack{r_2 - r_1 \\ r_3 - 3r_1}} \left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 2 & 3 & 8 \end{array}\right] \xrightarrow{r_3 + r_2} \left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 0 & 1 & 0 \end{array}\right]$$

  Here we have gone from the augmented matrix $[A \mid b]$ to a much nicer $[U \mid b]$. By recalling that matricies are just an abstraction of linear functions, we can look at the last row and say that $x_3 = 0$. We can *propogate* this upwards to say therefore that $-2x_2 + 0 = -8 \rightarrow x_2 = 4$

- If we get a 0 in a pivot position, we say that a matrix is **singular**. In this case $Ax = b$ may not have a solution, or it may have infinite solutions. But it certainly does not have a single solution. All linear equations of this type either have 0, 1, or ∞ solutions.

- It wouldn't hurt us to brush up on matrix multiplication either. It turns out there are many ways to think about this operation.

  1. **Entry-wise:** For each $1 \le i \le m$ and $1 \le jp$ we have

$$C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$$

  2. **Inner product:** C_{ij} is the dot product (also known as "inner product") of the ith row in A and the jth column in B. For instance,

$$C = \begin{bmatrix} — & x_1 & — \\ — & x_1 & — \\ & \vdots & \\ — & x_1 & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} x_1 \cdot b_1 & x_1 \cdot b_2 & \cdots & x_1 \cdot b_p \\ x_2 \cdot b_1 & x_2 \cdot b_2 & \cdots & b_2 \cdot b_p \\ \vdots & \vdots & \ddots & \vdots \\ x_m \cdot b_1 & x_m \cdot b_2 & \cdots & x_m \cdot b_p \end{bmatrix}$$

  3. **Column-wise:** the jth column of a matrix C is the matrix-vector product of A and the jth column of B. For instance,

$$C = A \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ Ab_1 & Ab_2 & \cdots & Ab_p \\ | & | & & | \end{bmatrix}$$

  4. **Outer product:** C is the sum of the product of ith column of A and the ith row of B such as

$$C = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} — & y_1 & — \\ — & y_2 & — \\ & \vdots & \\ — & y_n & — \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} | \\ a_i \\ | \end{bmatrix} \begin{bmatrix} — & y_i & — \end{bmatrix}$$

Properties of Matrix Multiplication

1. **Associative:** $A(BC) = (AB)C$
2. **Distributive:** $A(B + C) = AB + AC \iff (A + B)C = AC + BC$
3. **Non-commutative:** $AB \neq BA$
4. **Identity:** $IA = AI = A$

- Although it can often be lost in the abstraction of mathematics, matricies really, truly are **linear operators**. They transform spaces and vectors to other spaces and other vectors. As an example consider 2023 Recitation 1 Problem 1

Example: 1.1

1. Find a 2×2 matrix such that when you multiply a 2-D vector by it, the result is a reflection of the vector across the origin

2. Find a 3×3 matrix such that when you multiply a 3-D vector by it, it swaps the second and third coordinates.

3. If you have a 4×4 matrix A, find a 4-D vector x such that Ax is the second column of A.

Solution (1.1)**.**

1. To reflect across the origin, $x \mapsto -x$, $y \mapsto -y$. Therefore, $A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$.

2. Here $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} x \\ z \\ y \end{bmatrix} = x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + z \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

   Here A is a **permutation matrix**, as it permutes one or more of the variables. Permutation matricies have a variety of desirable properties such as $A^{-1} = A^T$

3. Since we are only interested in the second column, we want the products $\mathbf{c_i} x_i = 0$ for $i = 1, 3, 4$ to be 0. The only way to guaruntee this is to specify $x_{1,3,4} = 0$ and $x_2 = 1$. Therefore $\mathbf{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T$

Takeaways (1.1)**.**

1. All of the matrix formulations here are useful to recognize, particularly the intuition of part (3). This will be seen many times.

2. One useful way to interpret the result in part (2) is the fact that left-multiplied matricies operate on the rows of that which they multiply. Right-multiplied matricies operate on columns. Because $\mathbf{x}$ has only one column, right-multiplication is nonsensical.

Example: 1.2

1. Find a 3×3 matrix P such that in $B = PA$ is the result of subtracting the second row from the third row of A and then swapping the first and second rows.

2. Find a 4×4 matrix that right multiplies A such that result $C = AQ$ is A after dividing the first column by two, and then adding the first column to the second and third columns

3. Does the order of performing the operations in (1) and (2) matter?

Solution (1.2)**.**

1. $P = \underbrace{\begin{bmatrix} 1 & & \\ & 1 & \\ & -1 & 1 \end{bmatrix}}_{r_3 = r_3 - r_2} \underbrace{\begin{bmatrix} & 1 & \\ 1 & & \\ & & 1 \end{bmatrix}}_{\text{swap } \mathbf{c}_1, \mathbf{c}_2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix}$

2. $Q = \begin{bmatrix} 1/2 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. This multiplication should be obvious by the column-wise definition above.

3. Although it might look like we are dealing with the commutative propety here, we are really dealing with the associative property. In (1). $B = (P_1P_2)A = P_1(P_2A)$. So the order of operations does not matter. In (2) it is the same $C = A(Q_1Q_2) = (AQ_1)Q_2$

Takeaways (1.2)**.**

- Remember that left multiplication always affects rows only, and that right multiplication affects columns only. Complex operations can be formed by chaining linear operators together.

- Matrix multiplication is associative! When doing these chained operations, the order does not matter.

- Remember the forms that these kinds of matricies take. They are not always obvious.

- Before we close up, there are some other noteworthy operations we can perform with matricies that will follow us around. The **transpose** of a matrix A^T "flips" a matrix A such that

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

  and with it we can define certain identities such as $\boxed{(A^T)_{i,j} = A_{j,i}}$ and $\boxed{(AB)^T = B^TA^T}$.Finally, we can define the dot product between two vectors $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T\mathbf{b}$. All three of these facts will show up constantly.

- Last but not least, the **inverse** A^{-1} of a matrix is the matrix such that $AA^{-1} = A^{-1}A = I$. More interestingly we can say that in a situation $Ax = b$, $x = A^{-1}b$. This matrix inverse does not always exists, and is exceedingly impractical to calculate for large matricies, an issue we will deal with thoroughly in optimization. A matrix is said to be **invertible** if and only if it is square and full column rank. That is, every column has a pivot. The inverse, if it exists, is always unique. Like the transpose it is subject to the identity $\boxed{(AB)^{-1} = B^{-1}A^{-1}}$

- We can define more facts that will help us with these two operations. For instance $(A^T)^T$ and $(A+B)^T = A^T + B^T$, which follow directly from the definition of the transpose. More useful is $\boxed{(A^{-1})^T = (A^T)^{-1}}$. All of these identities will help us greatly.

1.2 Vector Spaces

Definition

A **vector space** V is a set of elements (e.g., vectors in $\mathbb{R}^n$, polynomials, diagonal 2×2 matricies) defined over a "field" F of scalars that are closed under

1. **Vector Addition:** For any vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in V, $\boldsymbol{u} \pm \boldsymbol{v}$ also belongs the vector space V.

2. **Scalar Multiplication:** For any vector $\boldsymbol{v}$ in V and scalar c in F, $c\boldsymbol{v}$ is an element of the vector space

- If $W \subseteq V$ is also a vector space with respect to the operations in V, W is a **subspace** of V. Specifically for any $v, w \in W$, $v + w \in W$ and $cv \in W$. For instance of we define a vector space $\mathbb{R}^2$. W could be $\mathbb{R}^2$ or $W = \{0\}$. A more enlightening sample vector space is a line that passes through the origin (note that a line that does not pass through the origin would not be a vector space since $\alpha v \notin V$ for $\alpha = 0$). Every point on this line is closed under addition and scalar multiplication.

- There are an infinite number of obscure facts to say about spaces and subspaces. More interestingly, is how to show something is a space or a substapce. We can be convinced that for subspaces S_1 and S_2 $S = S_1 \cap S_2$ is also a subspace by showing that S is closed under addition and scalar multiplication.

  Proof. Assuming that this is true, any $v, w \in S$ will also be $v, w \in S_1$ and $v, w \in S_2$ by the definition of an intersection. Then $v + w$ will be in S_1 and S_2 by the deifnition of subspace. Thus $v + w \in S_1 \cap S_2 = S$ which shows that it is closed under addition. Then we can define $v \in S_1 \cap S_2$ and $\alpha \in \mathbb{R}$. Since S_1 is a subspace, $\alpha v \in S_1$ and S_2 such that $\alpha v \in S_1 \cap S_2 = S$, showing that it is closed under scalar multiplication.

Common vector spaces

- The **column space** of an $m \times n$ matrix A, denoted $C(A)$ is the set of linear combinations of columns of A, also known as the **span** of A. Although the idea may seem a bit abstract, soon we will be very interested in what columnspace a vector is in. We can define useful facts such as

  1. Formally, $C(A) = \{Ax \mid x \in \mathbb{R}^n\}$
  2. $Ax = b$ has a solution if and only if $b \in C(A)$
  3. If $m = n$, then A is invertible if and only if $C(A) = \mathbb{R}^n$

- The **null space** of A, denoted $N(A)$ is the set of vectors x such that $Ax = \mathbf{0}$. We can define similar facts

  1. $N(A) = \{x \in \mathbb{R}^n \mid Ax = \mathbf{0}\}$
  2. If B is square and invetible, $N(A) = N(BA)$
  3. If $A \in \mathbb{R}^{n \times n}$, then $C(A) = \mathbb{R}^n$ is equivalent to $N(A) = \{\mathbf{0}\}$

Computing basis of a null space

- Before we start with computation, it helps to briefly examine our matrix. If we are looking at an invertible matrix, then its null space will clearly be $\{\mathbf{0}\}$, since there is no non-trivial x that will give $\mathbf{0}$. Even if A is not invertible, if every column has a pivot, then there are no variables that can move freely. This can happen easily for an "overdetermined" system where a matrix has more rows than columns. However, the most interesting case in which A is "underdetermined" such that it has many more columns than rows, and thus many free variables.

- Our goal for underdetermined systems will be to transform some matrix A
$$A \rightsquigarrow U = \begin{bmatrix} U_r & F \\ m-r \text{ rows of } 0's & \cdots \end{bmatrix}$$

- Now how do we implement this scary looking transformation? In reality it is quite simple. First transform A into an upper-triangular matrix U with gaussian elimination.

$$A = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 4 & 5 & -3 \\ 1 & 6 & 7 & -7 \end{bmatrix} \rightsquigarrow \begin{bmatrix} \boxed{1} & 2 & 3 & 1 \\ & \boxed{2} & 2 & -4 \\ & & & \end{bmatrix} \tag{1}$$

Let pivot columns $U_r = \begin{bmatrix} 1 & 2 \\ & 2 \end{bmatrix}$ and free columns $F = \begin{bmatrix} 3 & 1 \\ 2 & -4 \end{bmatrix}$ such that $U = \begin{bmatrix} U_r & F \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$.

- Then we can start with the actual linear algebra. To find $N(A)$ we want $Ax = \mathbf{0} \iff Ux = \mathbf{0}$. In order to compute this we define two vectors $\mathbf{p}$ for the coefficients of x by which the pivot columns of U are multiplied, and a vector f for the values of x that multiply the free columns. With this we can say,

$$U = \begin{pmatrix} U_r & F \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} U_r\mathbf{p} + F\mathbf{f} \\ \mathbf{0} \end{pmatrix} = \mathbf{0} \tag{2}$$
$$U_r\mathbf{p} + F\mathbf{f} = \mathbf{0} \tag{3}$$
$$\boxed{U_r\mathbf{p} = -F\mathbf{f}} \tag{4}$$

U_r is guarunteed to be invertible, so we can say $\mathbf{p} = U_r^{-1}(-F\mathbf{f})$ is guarunteed to be uniquely determined for any choice of $\mathbf{f}$. Then $\begin{pmatrix} \mathbf{p} \\ \mathbf{f} \end{pmatrix}$ is a basis vector of our nullspace. To show this, let's expand the boxed formula above for our example.

$$\begin{bmatrix} 1 & 2 \\ & 2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = -\begin{bmatrix} 3 & 1 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{5}$$

- For an n-dimensional null space, we can just make up whatever n linearly independent $\mathbf{f}$ vectors we might fancy. For simplicity, say $\mathbf{f} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ such that

$$\begin{bmatrix} 1 & 2 \\ & 2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -3 \\ -2 \end{bmatrix}}_{-F\mathbf{f}} \longrightarrow \mathbf{p} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \text{ and } x = \begin{bmatrix} -1 & -1 & 1 & 0 \end{bmatrix}^T \tag{6}$$

Then we can do this again for $\mathbf{f} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ to complete the basis for $N(A)$.

- In the case that the pivot columns are not adjacent, you *can* interlace the p and f elements systematically.

Computing the basis of a columnspace

- Let's revisit our matrix A from the previous example.

$$A = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 4 & 5 & -3 \\ 1 & 6 & 7 & -7 \end{bmatrix} \rightsquigarrow U = \begin{bmatrix} 1 & 2 & 3 & 1 \\ & 2 & 2 & -4 \\ & & & 0 \end{bmatrix} \tag{7}$$

In U we can easily identify that c_1 and c_2 contain our pivots. Therefore, our basis for $C(A)$ is simply columns c_1 and c_2 of A itself. Don't forget, though, that $C(A) \neq C(U)$. Rather,

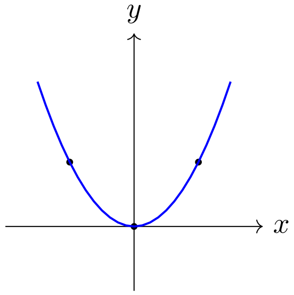
$$C(A) = \text{span}\left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \right\} \tag{8}$$

Polynomial fitting

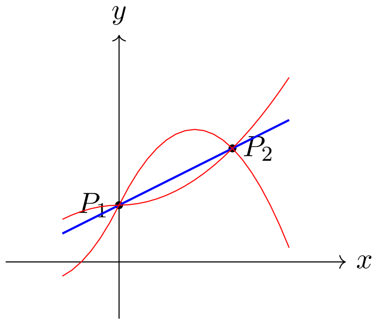
- The topic of optimization begins with polynomial fitting. This is a task we will revisit extensively in this course. A simple way to fit a polynomial is with a **Vandermode Matrix**. A degree-2 polynomial follows a form with linear coefficients $p(x) = c_0 + c_1 x + c_2 x^2$, leading to the intuition

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}}_{A} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

- This actually gives us a great way to visualize overdetermined and underdetermined systems.



$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} \rightarrow \underbrace{A \in \mathbb{R}^{3\times 3}}_{\text{1-sol}} \rightarrow N(A) = \{\mathbf{0}\}$$

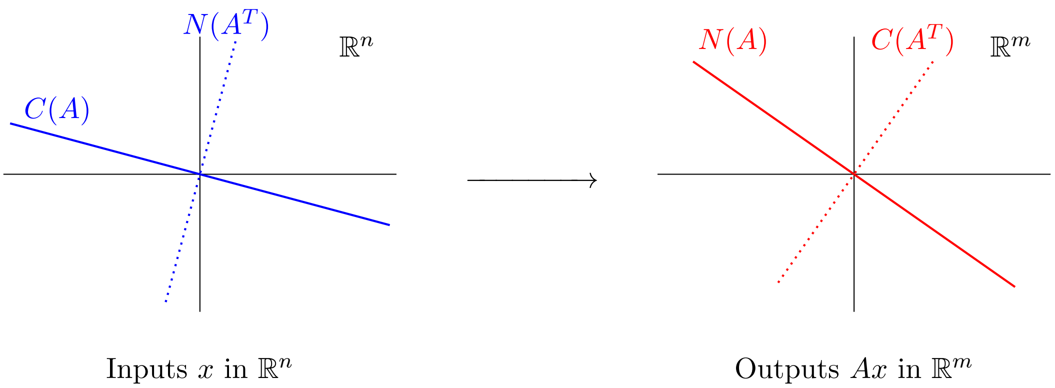


$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \rightarrow \underbrace{A \in \mathbb{R}^{2\times 3}}_{\text{underdetermined}} \rightarrow N(A) \neq \{\mathbf{0}\}$$

- As we can see with the figures, if we have two points, we can define exaclty one line. If we have 3 points, we can define exactly 1 parabola. This is a formalization of the intuitive fact that we can draw an infinte number of parabolas to fit two points.

Four Fundemental Subspaces

- We can model almost everything in the $Ax = b$ mathematical ecoysstem in terms of transformations between four subspaces, $C(A),\ N(A),\ C(A)^{\perp} = N(A^T),\ N(A)^{\perp} = C(A^T)$. Wonderfully, there is an amazing visual for this as well.



Example: 1.3

1. If the zero vector is in $C(A)$, then the columns of A are linearly dependent.
2. The columns of a matrix are a basis for the column space
3. Define the row space of A as the span of the row vectors. If A is square, then the row space equals the column space.
4. The row space of A is equal to the column space of A^T.
5. If the row space of A equals the column space, then $A^T = A$.
6. A 4×4 permutation matrix has $C(P) = \mathbb{R}^4$.
7. For $v \in N(A)$, if x is a solution to $Ax = b$, so is $x + v$

Solution (1.3)**.**

1. **False.** $A = I$ is a counterexample. The zero vector is in the column span of every matrix!
2. **False.** This is true only if the columns are linearly independent. It is clearly not true for underdetermined sysystems where there are more columns than rows.
3. **False.** As a counterexample $A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ where $C(A) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $R(A) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

4. **True.** This is simply the definition of the row space, and the set of rows of A is identical to the set of columns of A^T.
5. **False:** Consider $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. Both $C(A) = \mathbb{R}^2$ and $R(A) = \mathbb{R}^2$. However, $A \neq A^T$.
6. **True:** A permutation space simply permutes the order of the variables. If it were to not be in $\mathbb{R}^4$, it would somehow lose one of the dimensions.
7. **True:** If $v \in N(A)$, $Av = \mathbf{0}$. Therefore $A(x+v) = Ax + av = b$ which is defined in $C(A)$

Takeaways (1.3)**.**

1. We will see the row space $C(A^T)$ in more detail soon. This problem examines simple properties that it has. It is essential to remember that $C(A) \neq C(A^T)$, even if they have the same dimension.
2. Always look for simple matricies of size 2×2 or smaller tos how counterexamples. Most false properties will collapse even by this size.

Example: 1.4

$AB = 0$ (the zero matrix) for matricies A and B. If the null space of ___ is $\{=, \subseteq, \supseteq\}$ the column space of ___?

Solution (1.4)**.** Recalling our "Column-Wise" definition for matrix multiplication

$$AB = \begin{bmatrix} | & | & & | \\ Ab_1 & Ab_2 & \cdots & Ab_p \\ | & | & & | \end{bmatrix}$$

$AB = 0$ only if every column of B is in the null space of A. Therefore the $N(A)$ must contain any possible linear combination of B and possibly more. Therefore $N(B) \supseteq C(A)$

Example: 1.5

Show that if A is full column rank, then A^TA is invertible.

Solution (1.5)**.**

We know easily that A^TA will be square. What we really want to show here is that $N(A) = N(A^TA)$. The trouble is that in order to prove they are *equal*, we must show that **both** any $x \in N(A)$ is in $N(A^TA$ (i.e. $N(A) \subseteq N(A^TA)$) and that any $x \in N(A^TA)$ is in $N(A)$ (i.e. $N(A^TA) \subseteq N(A)$)

First, $x \in N(A) \to Ax = 0 \to A^T Ax = 0 \to x \in N(A^T A)$. Then $x \in N(A^T A) \to A^T Ax = 0 \to x^T A^T Ax = (Ax)^T (Ax) = \|Ax\|$, which is only 0 for $x \in N(A)$.

Takeaways (1.5)**.**

– Remember that to prove two subspaces are equal, you must show that they are **both** both contained inside one another.

– You can always multiply both sides by a vector or matrix that will help you get something into a nice form. Particularly if you have a transpose, look for the ability to convert it into a norm for a matrix, or a dot product for a vector.

1.3 Orthogonality

- Let us recall that for $x, y \in \mathbb{R}^n$, i.e. two "n-component vectors", we can define the dot product (a.k.a. the inner product) as $x^T y = \begin{bmatrix} x_1 & x_2 & \cdots \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix} = x_1 y_1 + x_2 y_2 + \cdots$. From this we can define the L^2 **norm** or length of x as $\|x\| = \sqrt{x^T x} = \sqrt{x_1^2 + x_2^2 + \cdots}$.
- Also we can recall the key property that a matrix transpose moves the matrix to the other side of a dot product. For example $u \cdot (Av) = u^T Av = (A^T u)^T v = (A^T u) \cdot v$
- In the two most common cases, orthogonal, and parallel, we will find nice properties. For $x \perp y$, $x^T y = 0$; for $x \parallel y$, $x^T y = \|x\| \|y\|$.
- In our original basis, we made no promises about what vectors defined the basis of the column space. This means that in order to solve $Ax = b$, we need to solve the whole system naively, which is computationally expensive. If we were to have a nicer basis, we could optimize some of that computation and inelegance. Imagine we have an **orthonormal basis** for a matrix. That is, the basis vectors of the matrix are orthogonal, and have magnitude 1. We will work on the details of how to compute one of these bases soon. For basis vectors $\mathbf{q}_1, \mathbf{q}_2$, we could define $\mathbf{b} = c_1 \mathbf{q}_1 + c_2 \mathbf{q}_2$ and $\|\mathbf{b}\| = b^T b = (c_1 \mathbf{q}_1 + c_2 \mathbf{q}_2)^T (c_1 \mathbf{q}_1 + c_2 \mathbf{q}_2) = c_1^2 + c_2^2$, or, the Pythagorean Theorem.
- In matrix form $Q = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 \end{bmatrix}$, we have desirable properties such as $Q^T Q = I$, assuming that Q is square, meaning that the transpose of Q is it's inverse! If Q is **not square**, we can say a lot less about it. It does not follow $Q^T = Q^{-1}$ and $Q^T Q \neq QQ^T$.
- We can also define orthogonal subspaces S_1 and S_2 of V if every vector in S_1 is orthogonal to every vector in S_2, or $x^T y = 0 \; \forall \; x \in S_1, y \in S_2$. Also, $\dim S_1 + \dim S_2 \leq \dim V$.
- Finally, we can define orthogonal complements of subspaces $S^\perp$ for the subspace containing *all* vectors in V that are othogonal to *every* vector in S. Therefore, $S^\perp$ is the largest subspace that is orthogonal to S. It is formally defined as $S^\perp = \{w \in V : w^T x = 0 \text{ for any } x \in S\}$

Example: 1.5

Denote a subspace $V = \left\{ \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} : 2v_1 + 3v_2 + 5v_3 = 0 \right\}$. Find $V^{\perp}$ (give a basis). Can you relate V and $V^{\perp}$ to column and/or null spaces of some matrix.

Solution (1.5)**.**
As we defined $V^{\perp} = \{w \colon w \cdot v = 0 \text{ for any } v \in V\}$. For $v = \begin{bmatrix} \mathrm{v}_1 & v_2 & v_3 \end{bmatrix}^T$,w = $\begin{bmatrix} 2 & 3 & 5 \end{bmatrix}^T$ such that $v \cdot w = 0$ as given. Therefore, $V^{\perp} = \text{span}\left\{ \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} \right\}$. Alternatively $A \equiv \begin{bmatrix} 2 & 3 & 5 \end{bmatrix}$ such that $Av = 0 \implies V = N(A)$ and $V^{\perp} = C(A^T)$.

Takeaways (1.5)**.**

- Remember the formal definition of the orthogonal subspace: it is just the subspace for which the dot product with *anything* is 0.
- This orthogonal subspace is highly related to the nullspace. Always be looking for the nullspace and columnspace connctions.

Example: 1.6

Suppose we have a subspace $\mathcal{S}$ with orthogonal (but not necessarily orthonormal) basis $\{v_1, \cdots v_k\}$

$$v = \sum_{i=1}^{k} \alpha_i v_i$$

by the definition of a basis, for some cosntants $\alpha_1, \ldots \alpha_k$. Determine every α_j in terms of $v_i, \ldots v_k$. Will this work if the basis is not orthogonal? If we put the vectors v_i as columns of a marix $V = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix}$, what special form does $V^T V$ have?

Solution (1.6)**.**
In order to find α_j specifically, left multiply by v_j^T such that

$$v_j^T v = \sum_{i=1}^{k} v_j^T(\alpha_i v_i) = \sum_{i=1}^{k} \alpha_i v_j^T v^i = \alpha_j \|v_j\|$$

The genius of muliplying by v_j^T is that for all terms $i \neq j$, $v_j^T v_i = 0$ such that we are able to get rid of that nasty summation. This leads easily to $\alpha_j = \frac{v_j^T v}{\|v_j\|^2}$. Then if we take the matrix V, we will end with just

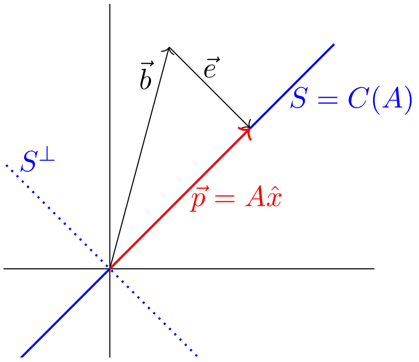
$$V^TV = \begin{bmatrix} \|v_1\| & & & \\ & \|v_2\| & & \\ & & \ddots & \\ & & & \|v_k\| \end{bmatrix}$$

since V forms a linear multiple of the basis vectors.

Takeaways (1.6)**.**

- The trick of left-multipling by v_j^T is tricky as hell here yet. A good problem solving technique is to look for what you're trying to isolate and then the conditions that can help you simplify the opaque linear algebra expressions.
- For the basis vectors, it's important to realize this property of v^TV that we will revisit on the section on symmetry later.

Projections



If we imagine some n-dimensional subspace $S \subseteq \mathbb{R}^n$, we can define a vector $\mathbf{b}$ that exists outside the subspace. We can also define a vector $\mathbf{p}$, the orthogonal projection of $\mathbf{b}$ onto S. Here we could say

$$\mathbf{b} = \underbrace{\mathbf{p}}_{\in S} + \underbrace{\mathbf{e}}_{\in S^\perp} = P\mathbf{b} + (I - P)\mathbf{b}$$

For our "projection matrix" P. Now, let's imagine we define $S = C(A)$ such that $S^\perp = N(A^T)$. Then, $\mathbf{p} = A\hat{x} \in C(A)$, and $\hat{x}$ is the closeset solution to $\mathbf{b}$. Therefore, $\mathbf{e} = \mathbf{b} - A\hat{x} \in N(A^T) \to A^T(b - A\hat{x}) \to A^Tb = A^TA\hat{x}$. From this, we can define three **normal equations**.

$$\boxed{\begin{array}{l} \hat{x} = (A^TA)^{-1}A^Tb \\ \mathbf{p} = A\hat{x} = P\mathbf{b} \\ P = A(A^TA)^{-1}A^T \end{array}}$$

- We can also define a handful of neat properties about the projection matrix P.

  1. $P^2 = P$ Proof. $A(A^TA)^{-1}(A^TA)(A^TA)^{-1}A^T = P$
  2. $P^T = P$ Proof. $\left((A^TA)^{-1}A^Tb\right)^T = (A^T)^T\left((A^TA)^{-1}\right)^TA^T = P$
  3. $C(P) = C(A)$
  4. $N(P) = C(A)^\perp = N(A^T)$
  5. $I - P$ is the projection matrix onto $N(A^T)$

- Our projection vector $\mathbf{p}$ is going to minimize the norm $\|b - Ax\|$ as should be clear in the geometry above. Notably, it won't necessarily come upwith an *exact solution*. Rather it is able to find the closest solution when there is no absolute solution. This is the beginning of our forees into optimization. As an example, let's examine the simplest optimization problem: least squares.

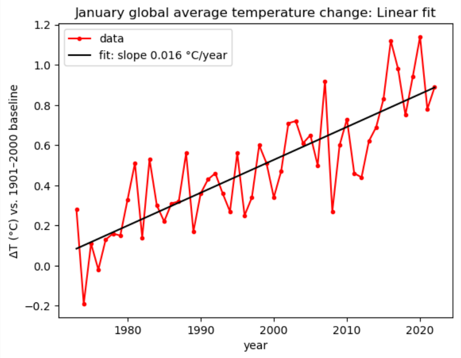


Figure from Prof. Johnson's Outstanding 18.065 Notes

Given a time series of temperature data, we want to calculate the linear regression with our matrix methods. We are going to construct another Vandermonde matrix: this time with a highly overdetermined system.

$$\begin{bmatrix} 1 & y_1 - y_0 \\ 1 & y_2 - y_0 \\ \vdots & \vdots \\ 1 & y_m - y_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \\ \vdots \\ \Delta T_m \end{bmatrix}$$

To find our solution, we must simply minimize the orthogonal projection with the normal equations

$$\operatorname*{argmin}_{x \in \mathbb{R}^2} \|b - Ax\| = (A^T A)^{-1} A^T b$$

1.4 Factorization

Before we get to factorization, we should define some simple concepts that have already appeared. For instance, the **rank** of a matrix is simply the number of pivots. Simply, $\operatorname{rank} A = \dim C(A) = \dim C(A^T)$, which should be intuitive. We can also define the **rank-nullity theorem** as $\dim N(A) = n - r$ for r pivots and n columns. Although we do not yet have the tools to introduce them all now, in this document we will examine **four critical factorizations**

$$A = LU \tag{9}$$

$$A = QR \tag{10}$$

$$A = U\Sigma V^T \tag{11}$$

$$S = Q\Lambda Q^T \tag{12}$$

LU Factorization

It turns out that matricies can be described in many ways. Some of these ways can help you solve difficult problems. The first of these is the ***LU*-factorization**, which relates A to its upper-triangular form U. In the process, it tells us about the inverse of a matrix.

The crux of the factorization is just Gaussian elimiation of $A \rightsquigarrow U$, but storing the coefficients of each row operation in a left-multipliying matrix L

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 5 & 1 \\ -3 & 1 & -1 \end{bmatrix} \xrightarrow[r_3+3r_1]{r_2-2r_1} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 7 & -1 \end{bmatrix} \xrightarrow{r_3-7r_2} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -8 \end{bmatrix} = U \tag{13}$$

$$\implies L = \begin{bmatrix} 1 & & \\ +2 & 1 & \\ -3 & +7 & 1 \end{bmatrix} \longrightarrow A = LU \tag{14}$$

And as it turns out it is *much* easier to calculate the inverse of a triangular matrix than a dense matrix. We can rely on $A^{-1} = (LU)^{-1} = U^{-1}L^{-1}$, and then on the kind fact that we can find each of those inverses through back substitution of U and forward-substitution of L.

In reality, it's worth noting that it's not *really* $A = LU$, but rather $PA = LU$. We must apply a permutation matrix to A in order to account for the fact that the pivots of A may not be perfectly ordered. We may at some point be forced to do a row swap, and hence apply a permutation at the end of the computation.

CR Factorization

This is not one of our "critical factorizations". However, it is useful for us. We learned a moment ago how to compute a basis for $C(A)$. We can use that to write $A = CR$, where $C \in \mathbb{R}^{m\times r}$ consists of *any* basis for C. The truth of this should be intuitive by how we have constructed the principle of a columnspace. We can take for example

$$A = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 2 & 5 & -3 \\ 1 & 2 & 7 & -7 \end{bmatrix} \rightsquigarrow U = \begin{bmatrix} 1 & 2 & 0 & 7 \\ & & 1 & 2 \\ & & & \end{bmatrix}$$

where clearly $\mathbf{c}_1$ and $\mathbf{c}_3$ are the pivot rows such that $C(A) = \{\mathbf{c_1}, \mathbf{c}_3\}$. We can from this express a *specific case* of CR-Factorization as

$$A = \underbrace{\begin{bmatrix} 1 & 3 \\ 1 & 5 \\ 1 & 7 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} 1 & 2 & 0 & 7 \\ 0 & 0 & 1 & 2 \end{bmatrix}}_{R}$$

where, here, C is the *specific* column space basis corresponding to the pivot columns of the original matrix, and R is the reduced row eschelon form of A, ommiting the trivial last row. This does not generalize to all solutions for the factorization.

QR Factorization

- As we showed earlier, our first basis is rarely a good choice. In order to solve for any solutions we need to do a full elimination. An orthonormal basis is the cleanest basis. But how can we go from our first basis $C(A) = \begin{bmatrix} \mathbf{a_1} & \cdots & \mathbf{a_n} \end{bmatrix} \longrightarrow \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_n \end{bmatrix} = Q$? We can use an algorithm relying on the projection techniques above called **Gram-Schmidt** orthogonalization.

- We define our first orthonormal vector with respect to the first normalized first basis vector $\mathbf{q}_1 = \frac{a_1}{\|a_1\|}$. From this, $\mathbf{q}_2 = \frac{\overbrace{(I - \mathbf{q}_1\mathbf{q}_1^T)}^{\text{proj}_\perp q_1}\mathbf{a}_2}{\|\text{''}\|} = \frac{\mathbf{a}_2 - \mathbf{q}_1(\mathbf{q}_1^T\mathbf{a}_2)}{\|\text{''}\|}$. This pattern continues $\mathbf{q}_3 = \frac{\mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^T\mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^T\mathbf{a}_3)}{\|\text{''}\|}$. The denominator is just the norm of the numerator. This gives us a number of orthonormal vectors, but we currently have no way to actually express A in terms of these vectors. We will have to work backwards.

$$\mathbf{q}_1 = \frac{\mathbf{a}_1}{\underbrace{\|a_1\|}_{r_{11}}} \qquad\qquad \mathbf{a}_1 = \mathbf{q}_1 r_{11}$$

$$\mathbf{q}_2 = \frac{\mathbf{a}_2 - \mathbf{q}_1 \overbrace{\mathbf{q}_1^T\mathbf{a}_2}^{r_{12}}}{\underbrace{\|\text{''}\|}_{r_{22}}} \qquad\qquad \mathbf{a}_2 = \mathbf{q}_1 r_{12} + \mathbf{q}_2 r_{22}$$

$$\mathbf{q}_3 = \frac{\mathbf{a}_3 - \mathbf{q}_1 \overbrace{q_1^T\mathbf{a}_3}^{r_{13}} - \mathbf{q}_2 \overbrace{\mathbf{q}_2^T\mathbf{a}_3}^{r_{23}}}{\underbrace{\|\text{''}\|}_{r_{33}}} \qquad\qquad \mathbf{a}_3 = \mathbf{q}_1 r_{13} + \mathbf{q}_2 r_{23} + \mathbf{q}_3 r_{33}$$

and the pattern so continues, giving us

$$\underbrace{\begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix}}_{\substack{A \\ m\times n}} = \underbrace{\begin{bmatrix} | & | & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \\ | & | & | \end{bmatrix}}_{\substack{Q \\ m\times n}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \end{bmatrix}}_{\substack{R \\ n\times n}}$$

- It is assumed here that A is **full column rank**. Hence, it will either be square or "tall". R is upper-triangular and invertible. R is upper triangular because, as we saw in the Gram-Schmidt algortihm, we construct our basis columns $\mathbf{q}_i$ of Q in the first place based on only $\mathbf{a}_{j\leq i}$
- For the purposes of 18.C06 as a class, it is much more important to understand the *shapes* and *behavior* of this factorization than work through the computation of the basis and R itself. THe full equations earlier were mostly expressed long-form to give a sense of the pattern how this relates to the projections into each orthogonal vector.

Singular Value Decomposition

- Ah, finally. The "factorization to rule them all", the **SVD**, which holds the deepest insight we can gain into the behavior of a matrix from it's representation in the four essential subspaces. For a rank-r matrix $A \in \mathbb{R}^{m\times n}$, we express it in it's "compact form" as

$$A = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_r \end{bmatrix}}_{\text{orthonormal basis for } C(A)} \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}}_{\text{scale factors } \sigma_k > 0} \underbrace{\begin{bmatrix} v_1 & v_2 & \cdots & v_r \end{bmatrix}^T}_{\text{orthonormal basis for } C(A^T)} = \hat{U}\hat{\Sigma}\hat{V}^T$$

- We call the vectors constituting matrix $\hat{U}$ the **left-singular** vectors, the vectors of $\hat{V}^T$, the **right-singular** vectors, and the scalar constants of $\hat{\Sigma}$ the **singular values**. These singular values are in sorted order, from greatest to smallest. This is essential.
- In general, it is very difficult to compute this factorization. For now we just won't worry about it. Furthermore, we won't worry *why* it exists either until we discuss eigenvalues and eigenvectors.
- So how does this work? Let's consider it's mathematical "mechanism of action" so to speak.
  1. $Ax \rightarrow U\Sigma(V^T x)$. This first projects x onto $C(A^T)$. If we recall the geometry of our fundemental subspaces,we see that this operation will project x orthogonal to $N(A)$, which is desirable.
  2. In $C(A^T)$, we scale up the components of $\hat{V}^T x$ using our singular-value matrix $\hat{\Sigma}$. This gives us our desired magnitudes.
  3. We project into our final $C(A)$ using $\hat{U}$ to give our final vector Ax
- If we look at our expression for the SVD earlier, it might appear how we cold express it in terms of a summation. For rank-r matrix A, consider

$$A = \hat{U}\hat{\Sigma}\hat{V}^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

- Imagine that we do not take the sum to completion. Instead, we only consider the first terms $k \leq r$. If we recall that the singular values are in sorted order, the **Eckart-Young Theorem** tells us that the best approximation for A of rank k is $A \approx \sum_{i=1}^{k} \sigma_i u_i v_i^T$. The output of this is called the **truncated SVD**.
- If we consider the shapes of our matricies $\hat{U}\hat{\Sigma}\hat{V}^T$, we see that $\hat{U}$, $\hat{V}^T$ are very rarely square. This means that they cannot be invertible, which could be inconvienent. In order to solve this, let's construct the **full SVD**. Currently the columns of $\hat{U}$ are an orthonormal basis for $C(A)$. Let's append a basis of $N(A^T)$ such that $U = \begin{bmatrix} U_r & U_\perp \end{bmatrix}$, and a basis of $N(A)$ to V such that $V = \begin{bmatrix} V_r & V_\perp . \end{bmatrix}$. We don't actually want any component of these vectors in our output, so we will modify $\hat{\Sigma}$ with the appropriate shape to 0 the components in the complementary spaces.

$$A = \underbrace{\begin{bmatrix} u_r & \cdots & u_m \end{bmatrix}}_{\substack{U \\ m\times m}} \underbrace{\begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & 0 & \cdots \\ & & 0 & 0 & \cdots \\ & & \vdots & \vdots & \ddots \end{bmatrix}}_{\substack{\Sigma \\ m\times n}} \underbrace{\begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}^T}_{\substack{V^T \\ n\times n}}$$

for Σ with $n - r$ extra columns of 0's and $m - r$ extra rows of 0's compared to $\hat{\Sigma}$.

Norms and Transformations

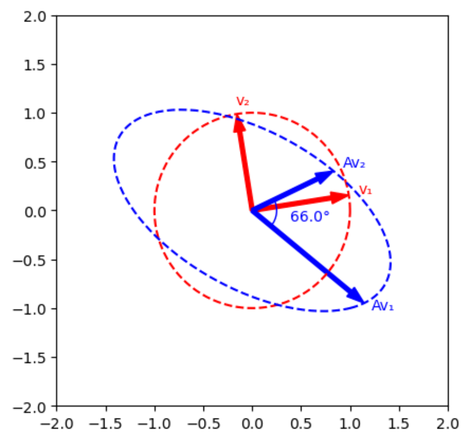


Figure from Prof. Johnson's 18.C06 Notebooks

A useful way to think of linear transformations on an orthonormal basis is an operator that transforms a unit circle into an ellipsoid, as the figure shows. This visualization is also the key to connecting the SVD to the eigenvectors of a matrix, which is perhaps the deepest connection in all of linear algebra.

We also can define here the **operator norm**, which is the largest amount by which A can stretch its orthonormal basis. For our case of SVD, the operator norm is always just the largest singular value.

This leads naturally to our formal definition of a **norm**. Given a vector space V a **norm** is a map $\vec{v} \in V \mapsto \|\vec{v}\|\mathbb{R}$ satisfying

1. **Non-negative:** $\|\vec{v}\| \geq 0$. $(= 0 \text{ iff } \vec{v} = \vec{0})$
2. **Scaling:** $\|\alpha\vec{v}\| = \alpha\|\vec{v}\|$
3. **Triangle Inequality:** $\|\vec{v_1} + \vec{v_2}\| \leq \|\vec{v_1}\| + \|\vec{v_2}\|$

Example: 1.7

Using this definition let's prove our lemma about the operator norm

$$\|Ax\|_0 = \max_{x\in\mathbb{R}^n} \frac{\|Ax\|}{\|x\|} = \sigma_{\texttt{max}}$$

.

Solution (1.7)**.**

1. First let's first prove $\|Ax\| \geq 0$. $\|Ax\| = (Ax)^T(Ax) = x^TA^TA^T = x^TV\Sigma U^TU\Sigma V^Tx = v^TV\Sigma^2V^Tx = (V^Tx)^T\Sigma^2(V^Tx)$. Let's define $c = V^Tx$. Then, we can decompose the system down to a sum of vector dot products such that $c^T\Sigma^2 c = \sum_{i=1}^{r} \sigma_i^2 c_i^2$.
2. By the triangle equality, $\|x\| \geq \|VV^Tx\|$ since VV^T is just a projection matrix that cannot increase the norm of x. From this we can say $\|VV^Tx\| = (VV^Tx)^T(VV^Tx) = x^TVV^TVV^Tx = x^TVV^Tx = (V^Tx)^T(V^Tx)$ which, by the c we defined earlier equals c^Tc. If we express it simarly as a sum, $\|x\| \geq c^Tc = \sum_{j=1}^{r} c_j^2$.

3. Putting the last two steps together, $\dfrac{\|Ax\|}{\|x\|} = \sqrt{\dfrac{\sum \sigma_i^2 c_i^2}{\sum c_i^2}} = \sqrt{\sigma_{\texttt{max}}^2} = \sigma_{\texttt{max}}$

Takeaways (1.7)**.**

- The decomposition of Σ into the sum of singular values is a very tricky lifesaver.
- You can get a lot out of expressing the 2-norm in terms of $x^T x$. Things often cancel which will lead you with an interesting expression.

- So why is any of that useful? Let's take a look at **error propogation** when solving linear systems. Suppose shape $A = m \times m$, rank $A = r$ and $x = A^{-1}b$. If we make a little error Δb in b (roundoff error, measurement noise, c), what is the error Δx?

$$A^{-1}(b + \Delta b) = A^{-1}b + A^{-1}\Delta b = x + \Delta x$$

This equation really doesn't tell us much. We want to know how Δb scales with $\|b\|$ and how Δx scales with $\|x\|$.

$$\frac{\|\Delta b\|}{\|b\|} = \frac{\|Ax\|}{\|x\|}\frac{\|\Delta x\|}{\|\Delta b\|} \leq \|A\|\|A^{-1}\| = \frac{\sigma_{\texttt{MAX}}}{\sigma_{\texttt{MIN}}} \equiv K(A)$$

where we are defining $K(A)$ as the **condition number** of A. This tells us how close A is to being singular.

- For example, the least singular matrix is I. $A = U\Sigma V^T = III$. All $\sigma = 1$. $K(I) = 1$. This also applies for any orthogonal matrix or multiple of Q or I.

Penrose Pseudo-inverse

With the SVD factorization, we can also describe the **pseudo-inverse** A^+ of A. This is a contrived generalization on our established idea of a matrix inverse that allows non-square matricies to be "inverted" such that $A^+A = I \;\forall\; A \in \mathbb{R}^{m\times n}$.

$$A^+ = \hat{V}\hat{\Sigma}^{-1}\hat{U}^T = \begin{cases} A^{-1} & \text{if } m = n \text{ (square)} \\ (A^TA)^{-1}A^T & \text{if } n = r \text{ (tall)} \\ A^T(AA^T)^{-1} & \text{if } m = r \text{ (wide)} \end{cases}$$

Example: 1.8

Prove using SVD that A^TA is invertible if A has full column rank and $C(A^TA) = C(A^T)$.

Solution (1.8)**.**

Say $A \in \mathbb{R}^{m\times n}$ and rank $A = r$ so that $A = \hat{U}\hat{\Sigma}\hat{V}^T \to A^TA = \hat{V}\hat{\Sigma}\hat{U}^T\hat{U}\hat{\Sigma}\hat{V}^T = \hat{V}_{n\times r}\hat{\Sigma}^2_{r\times r}\hat{V}^T_{r\times n}$. Since A is full column-rank, $n = r$ and rank $A^TA = n$. Knowing $A^T = \hat{V}\hat{\Sigma}\hat{U}^T$, we see that

$\hat{V}$ spans $C(A^T)$. We see the same in $A^TA = \hat{V}\hat{\Sigma^2}\hat{V}^T$ showing that $C(A^TA) = C(A)$

Takeaways (1.8)**.**

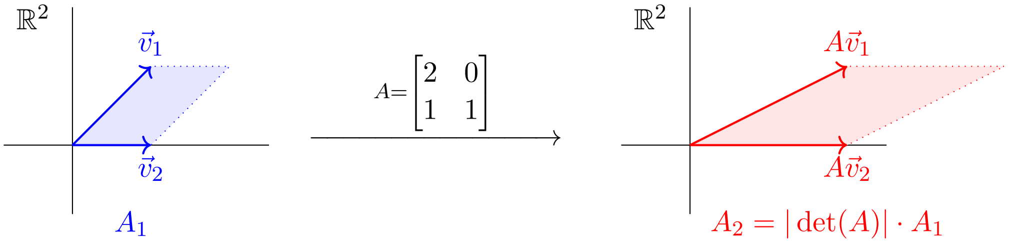
- The SVD transforms the vector into the column space spanned by the first matrix. This isn't always an obvious subspace.
- Consider writing out the shapes of the matricies explicitly during algebra.

1.5 Eigenproblems

- We have finally arrived at the biggest idea in linear algebra: eigenvalues and eigenvectors. These eigenproblems, so called, are the web with which we can tie together almost everything in linear algebra. They will also lead us to our final factorization $S = Q\Lambda Q^T$. For now, unfortunately, I must be coy, and insist we visit the determinant.

Determinant

- You may be surprised the determinant has not been so far mentioned. In reality, it's just not that useful for modern applications. But why don't we care about determinants anymore? While $\det A = ad - bc$ in $\mathbb{R}^2$, it turns out that past $\mathbb{R}^4$, the calculation of a determinant becomes so clumsy that we either must abstract it to computer computation or, if we are conservative about our compute, find ways to go around it to begin with.
- Geometrically, the determinant represnts the factor by which the area of parallelogram between two vectors is multiplied as the result of a vector a transformation. The following figure may elucidate this obscure geometric meaning.



- Yet, we are not geometers here. The most cherished property of the determinant is its part in deriving the eigenvalues and vectors. For that, we must understand its algebra. Symbolically, we can name many determinant identities that are useful. Here is a curation.

  1. **Zeros:** $\det A = 0$ if A is singular.
  2. **Normalization:** $\det I = 1$.
  3. **Swaps:** If you switch rows of A, flip the sign of $\det A$.
  4. **Multiplication:** $\det(AB) = \det A \det B$
  5. **Inverse:** $\det A^{-1} = \frac{1}{\det A}$

6. **Transpose:** $\det A^T = \det A$
7. **Linearity:** $\det \begin{bmatrix} \alpha a & \alpha b \\ c & d \end{bmatrix} = \alpha \det \begin{bmatrix} a & b \\ c & d \end{bmatrix} \longrightarrow \det(\alpha A) = \alpha^n \det A$ for $A \in \mathbb{R}^{n \times n}$.
8. **Row operations:** Adding a multiple of one row αr_i to another r_k does not affect the determinant. *Note this does not imply that* $\det A$ *is invariant under row operations (i.e. permutation, scalar multiplication, &c).*
9. **Triangular matricies:** For upper triangular matrix $U = \begin{bmatrix} d_1 & * & * \\ & d_2 & * \\ & & \ddots \end{bmatrix}$, $\det U = \prod_{i=1}^{k} d_i$.

Eigenvalues

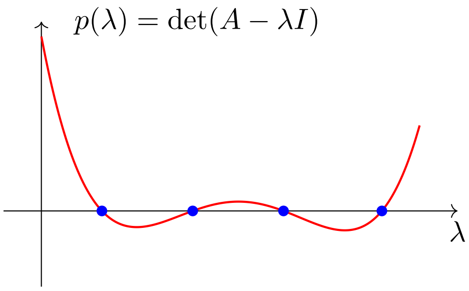
- The central idea of eigenvalues and eigenvectors is that we can find some vectors for which a linear operator A acts like a scalar multiplier λ on x. That is, only the *magnitude* of x is affected. It's direction stays the same. We express this as

$$\boxed{Ax = \lambda x}$$

  When this is true, we say x is our **eigenvector**, and λ is our **eigenvalue.**

- Normally, for a matrix $A \in \mathbb{R}^{n \times n}$, we will find n corresponding eigenvalues and eigenvectors $\lambda_1,\ \lambda_2,\ \ldots\ , \lambda_n$. A matrix that satisfiees this condition is called **diagonalizable**. Very few matricies will not. And we will address their special case, called "defective matricies" in due time.

- We will try to construct a basis of eigenvectors X such that we can take any vector x and expand it in the basis $x = c_1 x_1 + c_2 x_2 + \cdots = Xc$. For each eigenvector x_k, A acts as the scalar λ_k. In order to find the vectors, we much find the eigenvalues.



- Our defining case $Ax = \lambda_k x \implies (A - \lambda_k I)x = 0$, meaning that for λ_k, $A - \lambda_k I$ is singular, or $p(\lambda) = \det(A - \lambda_k I) = 0$. Our determinant will expand into some **characteristic polynomial** $p(\lambda)$. If A is square, the roots of $p(\lambda)$ are the eigenvalues. For example if we have

$$A = \begin{bmatrix} 1 & 1 \\ -2 & 4 \end{bmatrix}$$

  It's eigenvalues can be calculated by simply

$$\begin{aligned} \det(A - \lambda I) = \det \begin{bmatrix} 1-\lambda & 1 \\ -2 & 4-\lambda \end{bmatrix} &= (1-\lambda)(4-\lambda) - (1)(-2) \\ &= \lambda^2 - 5\lambda + 6 = (\lambda - 2)(\lambda - 3) \\ &\implies \lambda_{1,2} = \{2,\ 3\}. \end{aligned}$$

- Since we are dealing with polynomials here, we must prepare for our eigenvalues to be complex. In the case we have $\lambda_i \in \mathbb{C}$ for a real matrix, the complex eigenvalues will form a **complex conjugage pair**. We will see this again soon.

Eigenvectors

- Once we have the eigenvalues, the eigenvectors are just a basis for the $N(A - \lambda I)$. In our case from above

$$A - 2I = \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix} \text{ such that } (A - 2I)x_1 = \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

revealing that $x_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. We repeat the same operation for $\lambda_2 = 3$ to discover $x_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Diagonalization

- So why have we done all this to begin with? Really the purpose of constructing this eigenvector basis is to give us much greatear flexibility with our matrix operations. For an arbitrary matrix A, taking A^4 is a nightmare by naive methods. We can use our eigensolutions to give us a much deeper insight into these kinds of problems. For example, using our same A as above,

$$A^n \begin{bmatrix} 2 \\ 3 \end{bmatrix} = A^n \underbrace{\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right)}_{x_1 + x_2} = 2^n \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 3^n \begin{bmatrix} 1 \\ 2 \end{bmatrix} \approx 3^n \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ for } n \gg 1$$

as n grows, the largest eigenvector will **dominate** all smaller terms.

- This is a pretty slick expression. However you may be skeptical about how contrived the vector $\begin{bmatrix} 2 & 3 \end{bmatrix}^T$ is. Recalling diagonalization above, we can generalize this into a marvelous form. Recall, we are generally doing all of this to construct a basis of eigenvectors X. With this we can say something like

$$\begin{aligned} AX = \begin{bmatrix} Ax_1 & Ax_2 & \cdots & Ax_n \end{bmatrix} &= \begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \cdots & \lambda_n x_n \end{bmatrix} \\ &= X \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} = X\Lambda \\ & \boxed{A\vec{y} = X\Lambda X^{-1}\vec{y}} \end{aligned}$$

  - $\boldsymbol{X^{-1}}\mathbf{y} \equiv \mathbf{c}$ projects y into the X basis.
  - $\boldsymbol{\Lambda}$ multiplies each coefficient c_k by it's appropriate λ_k.
  - $\boldsymbol{X}$ adds up the eigenvectors and their coefficients.

- With this, we can now make useful expressions such as $\vec{y} = \vec{x}_1 c_1 + \vec{x}_2 c_2 + \cdots$ or similarly that $A\vec{y} = \lambda_1 \vec{x}_1 c_1 + \cdots$.

- Now why did we expend all of that effort? We can connect this to our example of matrix powers above in a *much* more elegant way. Here, $A^n = X\Lambda X^{-1} X\Lambda X^{-1} X\Lambda X^{-1} \cdots = \boxed{X\Lambda^n X^{-1}}$. Since Λ is a diagonal matrix, $\Lambda^n = \text{diag}\left(\lambda_1^n,\ \lambda_2^n,\ \cdots\ \lambda_r^n\right)$ which is just scalar multiplication.

- And, as we saw earlier with eigenvalue domination, for large n,

$$\Lambda^n = \lambda_1^n \begin{bmatrix} 1 & & \\ & (\lambda_2/\lambda_1)^n & \\ & & \ddots \end{bmatrix} \approx \begin{bmatrix} 1 & & \\ & 0 & \\ & & \ddots \end{bmatrix}$$

as the lower-order terms get exponentially small.

Determinant and Trace

Before going any further, let's quickly define two facts that we will exploit thoroughly. We define the **trace** operation $\operatorname{tr} A$ as the sum of it's diagonal enteries for any square A. It obeys a handful of nice properties, particularly that $\operatorname{tr}(AB) = \operatorname{tr}(BA)$. Therefore, $\operatorname{tr} A = \operatorname{tr}(X\Lambda X^{-1}) = \operatorname{tr}(XX^{-1}\Lambda) = \operatorname{tr}\Lambda$. Simply put,

$$\operatorname{tr} A = \sum_i \lambda_i$$

We can also uncover a hideen identity of the determinant now. As we established earlier, $\det(AB) = \det A \det B$. Therefore $\det A = \det X \det \Lambda (\det X)^{-1} = \det \Lambda$. Alternatively,

$$\det A = \prod_i \lambda_i$$

These results are *very* useful. Last but not least, these operations also define a shortcut formula for our characteristic polynomial

$$p(\lambda) = \lambda^2 - \operatorname{tr} A + \det A$$

- We would say that a matrix A is **similar** to B if $A = SBS^{-1}$ for some invertible S. Should this be true we can say $\det A = \det B,\ \operatorname{tr} A = \operatorname{tr} B,\ \det(A - \lambda I) = \det(SBS^{-1} - \lambda SS^{-1}) = \det(S(B - \lambda I)S^{-1}) = \det(B - \lambda I)$. In short, the eigenvalues are the same, but the eigenvectors are not.

2 Applied Linear Algebra

2.1 Tikhonov Regularization

2.2 Statistics

- For some black box distribution that generates samples x_k, we have m data points and sample mean

$$\mu = \frac{1}{m} \sum_{k=1}^{m} x_k.$$

As the limit $m \to \infty$ the sample mean μ approach the true mean. Until then, we say the **sample variance** S^2

$$\operatorname{Var}(x) = S^2 = \frac{1}{m-1} \sum_{k=1}^{m} (x_k - \mu)^2$$

- We can define the statistical mean in linear algebra terms by establishing $\vec{o} = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}^T$. By the definition of the dot product we know $\sum_{i=1}^{n} x_i = \vec{o}^T x$ and that $m = o^T o$. Substituting these into our summation, $\mu = \frac{o^T x}{o^T o}$. We can rewrite the variance in a similar way

$$\text{Var}(x) = \frac{\|\vec{x} - \mu\vec{o}\|^2}{m-1} = \frac{\|(I - \frac{oo^T}{o^T o})\vec{x}\|^2}{m-1}$$

- One may also be interested in evaluating the correlation between two sets of data, as measured by the **covariance**. For two variables x and y,

$$\text{Covar}(x, y) = \frac{1}{m-1}\sum_{k=1}^{m}(x_k - \mu_x)(y_k - \mu_y = \frac{(Px)^T(Py)}{m-1} = \frac{x^T Py}{m-1}$$

for projection matrix $P = I - \frac{oo^T}{o^T o}$.