Linear Algebra

Vectors

Matrices

Matrix Multiplication

Matrix Norms

Matrix Rank

Matrix Spaces

Systems of Linear Equations

Singular Value Decomposition

Quadratic Form

Vectors

Interpretations of Vectors

- **Algebraic vectors** ($\boldsymbol{v}$, $\vec{v}$): an ordered list of numbers, e.g.,

$$\boldsymbol{v} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

  - Vectors can be written as rows (seen above) or columns (seen below), but differ only at the level of notation, some notation is more useful with certain conventions.
  - The order of elements in a vector matters:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \neq \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

- **Dimensionality**: the number of elements in a vector, where each new element provides new information, or geometrically, a new direction.
- **Euclidean (geometric, spatial) vectors**: a line in geometric space that indicates the magnitude and direction from a starting point (tail) to an end point (head).
  - Geometric vectors can start at any point in space, but often represented as starting from the origin—such vectors are in standard position.
  - Coordinates are not the same as vectors, but they do indicate where the head of a vector will land if it is in standard position.

Vector Addition and Subtraction

  - Algebraically, dimensionality of vectors must be equal. When they are, then addition or subtraction vectors is done on the corresponding elements of each vector, e.g.,

$$\begin{bmatrix} 1 \\ 0 \\ 4 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \\ -6 \\ 11 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ -2 \\ 16 \end{bmatrix}$$

  - Geometrically, addition can be thought of translating the tail of one vector to the head of the other—resulting in a new vector.
  - Geometric interpretations of subtraction can be thought of in two ways:
    1. Multiplying one vector by -1, then applying vector addition method above.
    2. Placing both vectors in standard position, with the resulting vector between the two heads being the answer.

Vector Scalar Multiplication

- **Scalar**: typically denoted with lower case Greek letters (e.g., α, λ) indicating an element of a field (typically real numbers) used in scalar multiplication of vectors.
- Algebraically, scalar multiplication is the multiplication of each element of a vector by a particular scalar, e.g.,

$$\lambda \boldsymbol{v} \rightarrow 7 \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -7 \\ 0 \\ 7 \end{bmatrix}$$

- Geometrically, scalar-multiplication can be thought of as the extension ($\lambda > 1$) or compression ($\lambda \in (0, 1)$) of a vector.
  - When $\lambda < 0$, then it can be thought of inverting its direction with respect to the origin.

The Dot Product

- **Dot (scalar, inner) product**: an algebraic operation that takes two equal-length sequences of numbers (usually coordinate vectors), and returns a single number.
  - The result of a dot product is a scalar↑, so often it is represented as such.
    - It can also be represented as multiplication between two vectors ($\boldsymbol{a} \cdot \boldsymbol{b}$).
    - However, it is commonly represented as $\boldsymbol{a}^T \boldsymbol{b}$ – transpose↓ will be explained in more detail when dealing with matrix multiplication↓.
  - Algebraically: $\sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{b}_i$ – where Σ denotes summation and n is the dimension of the vector space, e.g.,

$$\begin{bmatrix} 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 4 \\ -2 \\ 1 \end{bmatrix} = (1 \cdot 4) + (3 \cdot -2) + (-5 \cdot -1) = 3$$

Properties of the Dot Product

- Note: the following properties hold as long as $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are real vectors.
- ✓ Distributive: $\boldsymbol{a}^T(\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a}^T \boldsymbol{b} + \boldsymbol{a}^T \boldsymbol{c}$ – vector multiplication distributes over vector addition.
- ✘ Associative: $\boldsymbol{a}^T(\boldsymbol{b}^T \boldsymbol{c}) \neq (\boldsymbol{a}^T \boldsymbol{b})\boldsymbol{c}$ – in general the associative property does not hold, as the dot product would most likely produce different scalars.
  - Additionally, $\boldsymbol{a}$ could have a different dimensionality than $\boldsymbol{b}$ and $\boldsymbol{c}$. I.e., even if $\boldsymbol{b}$ and $\boldsymbol{c}$ had the same dimensionality ($\boldsymbol{a}^T(\boldsymbol{b}^T \boldsymbol{c})$ would be valid vector scalar multiplication) then $\boldsymbol{a}^T \boldsymbol{b}$ would be invalid.
- ✓ Commutative: $\boldsymbol{a}^T \boldsymbol{b} = \boldsymbol{b}^T \boldsymbol{a}$ – the order of the vectors does not matter.

Vector Length

- **Vector norm (magnitude, length)**: denoted with double vertical bars $\|\boldsymbol{v}\|$, indicating length of a vector in euclidean space. Not to be confused with absolute value $|x|$ of a scalar's "norm." However, sometimes the notation $|\boldsymbol{v}|$ is used.
- Calculating $\|\boldsymbol{v}\|$ is done using the Euclidean norm:

$$\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

  - This is a consequence of the Pythagorean theorem, since the basis vectors$^{\downarrow}$ $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ are orthogonal$^{\downarrow}$ unit vectors$^{\downarrow}$.
- Thus, the norm can easily be found by taking the square root of the dot product of the vector with itself:

$$\|\boldsymbol{v}\| = \sqrt{\boldsymbol{v}^T v}$$

Geometric Interpretation of the Dot Product

- The dot product of two Euclidean vectors$^{\uparrow}$ $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined by:

$$\lambda = \boldsymbol{a}^T \boldsymbol{b} = \|\boldsymbol{a}\| \, \|\boldsymbol{b}\| \cos(\theta)$$

  where θ is the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$.
- **Features based on θ**:
  - When $\cos\theta > 0$ ($\theta < 90° -$ acute) then $\lambda > 0$ $(+)$
  - When $\cos\theta < 0$ ($\theta > 90° -$ obtuse) then $\lambda < 0$ $(-)$
  - When $\cos\theta = 0$ ($\theta = 90° -$ perpendicular) then $\lambda = 0$
    - This represents a special case where the vectors are said to be orthogonal.
    - **Orthogonality**: the generalization of the notion of perpendicularity to the linear algebra of bilinear forms.
  - When $\cos\theta = 1$ then the vectors are codirectional:

$$\boldsymbol{a}^T \boldsymbol{b} = \|\boldsymbol{a}\| \, \|\boldsymbol{b}\|$$

    - Thus, the dot product with a vector $\boldsymbol{v}$ with itself is

$$\boldsymbol{v}^T \boldsymbol{v} = \|\boldsymbol{v}\|^2$$

    - Which gives us the norm$^{\uparrow}$ as defined above, i.e., $\|\boldsymbol{v}\| = \sqrt{\boldsymbol{v}^T v}$
    - If $\cos\theta = -1$, then really vectors are still codirectional, but point in opposite directions with respect to the origin.

Properties of Vectors

- Matrices↓ are properly defined later. However, vectors are technically single row or column matrices, so many of the following operations also work on vectors, thus use of matrices often appears in this section.

Vector Hadamard Multiplication

- **Hadamard (element-wise) product**: a binary operation (only takes two operands) that matrices of the same dimensions and produces another matrix of the same dimension as the operands, e.g., vector Hadamard multiplication:

$$\begin{bmatrix}1\\0\\4\\5\end{bmatrix} + \begin{bmatrix}2\\3\\-6\\11\end{bmatrix} = \begin{bmatrix}2\\0\\-24\\55\end{bmatrix}$$

Outer Product

- Recall that the dot (scalar, inner) product↑ produces a 1×1 matrix, or rather a scalar, hence "scalar" product.
  - Note: the typical notation used for dot products is $\boldsymbol{v}^T \boldsymbol{w}$ — part of reasoning for the "inner" product.
- **Outer product**: an $n \times m$ matrix that results from the product of two vectors with dimensions n and m.

$$\boldsymbol{v}\boldsymbol{w}^T = n \times m$$

- The subtle change in notation matters in contrast to the dot product, as both represent distinct operations (assuming they are column vectors).
- The outer product allows for the multiplication of vectors with different dimensionality
- Can be thought of in two different ways:
  - The row perspective:

$$\begin{bmatrix}1\\0\\4\\2\end{bmatrix}\begin{bmatrix}a & b & c\end{bmatrix} = \begin{bmatrix}1a & 1b & 1c\\0a & 0b & 0c\\4a & 4b & 4c\\2a & 2b & 2c\end{bmatrix}$$

  - The column perspective:

$$\begin{bmatrix}1\\0\\4\\2\end{bmatrix}\begin{bmatrix}a & b & c\end{bmatrix} = \begin{bmatrix}a1 & b1 & c1\\a0 & b0 & c0\\a4 & b4 & c4\\a2 & b2 & c2\end{bmatrix}$$

Cross Product

- **Cross (vector, directed area) product**: denoted by the symbol $\times$, indicating a binary operation on two vectors in three-dimensional space $\mathbb{R}^3$.
  - Given two linearly independent↓ vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, then the cross product $\boldsymbol{a} \times \boldsymbol{b}$ produces a new vector that is orthogonal to both $\boldsymbol{a}$ and $\boldsymbol{b}$, or normal to the plane containing them.
  - The direction of the vector is given by the right-hand rule ($\boldsymbol{a}$ = pointer, $\boldsymbol{b}$ = index, thumb = direction).
  - The magnitude of the vector represents the area of the parallelogram that the vectors span.
- The cross product can be defined by the formula:

$$\boldsymbol{a} \times \boldsymbol{b} = \|\boldsymbol{a}\| \, \|\boldsymbol{b}\| \sin(\theta) \, \boldsymbol{v}$$

  - Notice this is similar to the geometric interpretations of the dot product↑ — the contrast between the two leads to an intuitive interpretation:
    - $\cos(\theta)$ in the dot product is used to measure how "parallel" the two vectors are, i.e., they are codirectional when $\theta = 1$, allowing for calculation of the norm↑.
    - $\sin(\theta)$ in the cross product is used to measure how "perpendicular" two vectors are, i.e., they are orthogonal when $\theta = 1$. There are multiple directions of the orthogonal vector, so calculation of the signed area returns vector $\boldsymbol{v}$ that describes both magnitude and direction as described above.
    - The intuition described here will be more clear when the determinant↓ is discussed in more detail.
- An algebraic example of vector e.g.,

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2c & - & 3b \\ 3a & - & 1c \\ 1b & - & 2a \end{bmatrix}$$

Primer on Complex Numbers

- **Complex number**: a number that can be expressed in the form $a + bi$, where a and b are real numbers, and i (j) is a symbol called the imaginary unit that satisfies the equation $i = \sqrt{-1}$.
- This imaginary unit allows for the imaginary set ($a + bi \in \mathbb{C}$) to be described.
- The combination of a real part and an imaginary part ($a + bi$) gives imaginary numbers both a direction and magnitude on the 2-D plane created between the two parts—thus they can be described as 2-D vectors in this space.

- Multiplication of complex numbers can be done by factoring the imaginary unit, e.g.,

$$z = a + bj$$
$$w = c + dj$$

$$\begin{aligned} zw &= (a + bj)(c + dj) \\ &= ac + adj + cbj + bdj^2 \\ &= ac + adj + cbj - bd \qquad (j^2 = -1) \end{aligned}$$

- Computing the dot product with complex vectors is the same, just including the factoring mentioned above when necessary, e.g.,

$$\begin{bmatrix} 1+3j \\ -2j \\ 4 \\ 5 \end{bmatrix}^T \begin{bmatrix} 6+2j \\ 8 \\ 3j \\ -5 \end{bmatrix}$$
$$= (1+3j)(6+2j) + -16j + 12j + 25$$
$$= 6 + 2j + 18j - 6 - 16j + 12j + 25$$
$$= 25 + 16j$$

Conjugate Transpose

- **Conjugate (Hermitian) transpose** M^H, M^*: the n-by-m matrix obtained by taking the transpose$^\downarrow$ and then taking the complex conjugate of each entry.
- **Complex conjugate**: the number with both equal real and imaginary parts equal in magnitude but opposite in sign.

$$a + bi = a - bi \iff a - bi = a + bi$$

- Multiplying a vector by the conjugate transpose allows us to calculate the magnitude of the vector containing imaginary numbers by using the complex conjugate to "canceling out" all the imaginary units and give a real number answer (or rather, $a + 0i$).
- Matrices will return a matrix of real numbers—this is part of what makes using imaginary numbers useful for future computations.

Unit Vectors

- **Unit vectors** $\hat{\imath}, \hat{\jmath}, \hat{k}$: a vector scaled such that $(:)$ the length of the vector equals 1 in a normed vector space.

$$\hat{\imath} = \lambda \boldsymbol{v} : \|\lambda \boldsymbol{v}\| = \frac{1}{\|\boldsymbol{v}\|}\|\boldsymbol{v}\| = 1$$

- **Normed vector space**: a vector space, over the real or complex numbers, on which a norm$^\uparrow$ is defined.

Linear Combinations

- **Linear combination**: an expression constructed from a set of terms by multiplying each term by a scalar and adding the results, i.e.,

$$a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + a_3\boldsymbol{v}_3 + a_n\boldsymbol{v}_n$$

- For example, in a 3-D vector space $\mathbb{R}^3$, then any vector in the space is can be made by a linear combination of the following three vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$ (unit vectors) multiplied by some scalar:

$$\lambda\begin{bmatrix}1\\0\\0\end{bmatrix} + \lambda\begin{bmatrix}0\\1\\0\end{bmatrix} + \lambda\begin{bmatrix}0\\0\\1\end{bmatrix}$$

Subspace

- **Linear (vector) subspace**: a vector space that is a subset of some larger vector space.
- Algebraically, a subspace is the set of all vectors that can be created by taking a linear combination of some vector or a set of vectors.
- A vector subspace must be closed under addition and scalar multiplication while also containing the zero vector, i.e.,

$$\forall \boldsymbol{v}, \boldsymbol{w} \in L; \quad \forall \lambda, \alpha \in \mathbb{R}; \quad \lambda\boldsymbol{v} + \alpha\boldsymbol{w} \in L$$

  - For all ($\forall$) vectors $\boldsymbol{v}$, $\boldsymbol{w}$ in ($\in$) the linear subspace (L), and for all scalars λ, α in the set of real numbers ($\mathbb{R}$), then any linear combination of the two are in the same subspace.
  - The above equation also implies the inclusion of zero vector, which is a trivial subspace of the vector space.
- The geometric interpretation is best described through some examples:
  - A $\mathbb{R}^1$ vector and all the scaled possibilities added together describes a line stretches infinity in both directions, while a $\mathbb{R}^2$ vector would create a 2-D plane.
  - Both of the previous subspaces exist in the higher dimensional vector spaces, i.e., a 2-D plane and a 1-D line both exist in the 3-D vector space.
  - All subspaces also pass through the origin, including the 0-D subspace, which is just the origin.

Subsets

- **Subset** $\subseteq$: a set A is a subset of a set B if all elements of A are also elements of B; this makes B a **superset** $\supseteq$ of A.
- Not all subsets of vector spaces are subspaces; subsets don't need to include the origin, don't need to be closed, or could have boundaries.

Span

- **Linear span (hull)** $\text{span}(S)$: a set S of vectors (from a vector space) that is the smallest linear subspace that contains the set.
- The span is typically infinite, but it also can be defined as the set of all finite linear combinations[↑] of vectors of S given an arbitrary field K:

$$\text{span}(S) = \left\{ \sum_{i=1}^{k} \lambda_i \boldsymbol{v}_i \;\middle|\; k \in \mathbb{N}, \boldsymbol{v}_i \in S, \lambda_i \in K \right\}$$

- A frequent question is asked whether a vector is in a span or not, e.g., are vectors $v, w \in \text{span}(S)$:

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} w = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} S = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 7 \\ 0 \end{bmatrix} \right\}$$

- $\boldsymbol{v} \in \text{span}(S)$ since both vectors in S can be scaled then combined in some way to form $\boldsymbol{v}$, i.e.,

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \frac{5}{6} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \frac{1}{6} \begin{bmatrix} 1 \\ 7 \\ 0 \end{bmatrix}$$

- However, it's clear that $\boldsymbol{w} \notin \text{span}(S)$ since 0 cannot be scaled to equal 1.
- Geometrically, a useful intuitive example is a 2-D span in 3-D space; the 2-D plane can be moved anywhere in the 3-D space as long as the two vectors describing the 2-D plane are linearly independent[↓].

Linear Independence

- **Linearly dependent**: a set of V vectors where at least one vector in the set can be defined as a linear combination[↑] of the others.
  - Algebraically, a sequence of vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$ from a vector space are linearly dependent if there exist scalars $\lambda_1, \lambda_2 \ldots s, \lambda_k$ such that they form the zero vector $\boldsymbol{o}$, i.e.,

$$\lambda_1 \boldsymbol{v}_1, \lambda_2 \boldsymbol{v}_2, \ldots, \lambda_k \boldsymbol{v}_k = \boldsymbol{o} \qquad \lambda \in \mathbb{R}$$

- **Linearly independent**: when no vector in the set can be written in the above way, i.e.,

$$\lambda_1 \boldsymbol{v}_1 + \lambda_2 \boldsymbol{v}_2 + \cdots + \lambda_k \boldsymbol{v}_k = \boldsymbol{o} \qquad \lambda \in \mathbb{R}$$

- Geometrically, a set of V vectors in independent if each vectors points in a geometric dimension not reachable using other vectors in the set.
  - E.g., if two vectors are lie along the same line, then they are really just the same vector; same concept with a 2-D plane containing three vectors in $\mathbb{R}^3$.

Basis

- **Basis**: the combination of span↑ and linear independence↑, i.e., the set B of vectors in vector space V if every element of V may be written as a unique finite linear combination↑ of elements of B.
  - More concisely, a basis is simply a linearly independent spanning set.
  - **Components (coordinates)**: the coefficients of the linear combination and are referred to as of the vector with respect to B, or **basis vectors**.
- Examples of standard basis vectors:

$$\mathbb{R}^2 \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \mathbb{R}^3 \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

- The standard basis vectors of $\mathbb{R}^3$ was used in the example of linear combinations↑ to demonstrate how these unit vectors could all be scaled so that they could describe any vector in the space.
- Basis vectors can be changed, so it's better to think of them as the "rulers" used to describe any other vector in the space.

Matrices

Matrix Terminology

- **Matrix $A_{r,c}$**: a rectangular array of elements arranged in rows $\leftrightarrow$ and columns $\updownarrow$, e.g.,

$$A = \begin{bmatrix} 1 & 0 & 3 \\ 5 & 4 & 2 \\ 7 & 6 & 9 \end{bmatrix} \qquad A_{3,2} = 6$$

- **Block (partitioned) matrix**: a matrix that is interpreted as having been broken into sections called blocks or submatrices, e.g.,

$$A = \begin{bmatrix} D & N \\ Y & D \end{bmatrix} = \begin{bmatrix} 4 & 2 & 0 & 0 \\ 6 & 9 & 0 & 0 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 6 & 9 \end{bmatrix}$$

$$D = \begin{bmatrix} 4 & 2 \\ 6 & 9 \end{bmatrix} \quad N = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

  - Can be used for large matrices with high level structure, offering convenient notation, and sometimes providing computational benefits.

- **Diagonal**: the elements of matrix starting from the top left $\searrow$ lower right.
  - **Off-diagonal**: elements not along the diagonal (0s and 1s in example below)
  - Works for both square and rectangular matrices$^{\downarrow}$, e.g.,

$$\begin{bmatrix} 4 & 1 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 6 & 0 \\ 1 & 1 & 0 & 9 \end{bmatrix} \qquad \begin{bmatrix} 4 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 6 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 9 & 1 & 0 & 1 \end{bmatrix}$$

- **Matrix size**: a matrix with m rows and n columns is called an $m \times n$ matrix, or m-by-n matrix, while m and n are the dimensions.
  - Order matters—the convention is rows then columns, i.e., $m \times n \neq n \times m$.
  - **MR**. **NiC**e guy: a useful mnemonic to remember typical conventions.
- **Matrix dimensionality**:
  - $\mathbb{R}^{mn}$: describes the total number of elements, here the multiplication of the dimensions is commutative (order doesn't matter).
  - $\mathbb{R}^{m \times n}$: the specific matrix size using rows and columns as described above.
  - $C(A) \in \mathbb{R}^{m}$: a collection of column vectors, i.e., a matrix spanned by set column vectors with m elements.

- $R(\boldsymbol{A}) \in \mathbb{R}^n$: a collection of row vectors, the inverse of above; more on row and column spaces↓ later.

Square and Rectangular Matrices

- **Square matrix**: a matrix with the same number of rows and columns.
  - An $n \times n$ matrix is known as a square matrix of order n.
  - Any two square matrices of the same order can be added and multiplied.
- **Rectangular matrix**: a matrix with an unequal number of rows and columns, i.e., $m \neq n$.
- Both square and rectangular matrices have a diagonal↑, as described above.

Symmetric and Skew-Symmetric Matrices

- **Symmetric matrix**: a square matrix that can be mirrored across the diagonal, e.g.,

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

  - Algebraically, it's a square matrix $\boldsymbol{A}$ that is equal to its transpose↓, i.e., $\boldsymbol{A} = \boldsymbol{A}^T$.
  - It does not matter what is on the diagonal, as any number is equal to itself.
- **Skew-symmetric matrix**: a square matrix that is still symmetric, but all elements mirrored across the diagonal and inverted, e.g.,

$$\begin{bmatrix} 0 & +6 & +1 \\ -6 & 0 & -9 \\ -1 & 9 & 0 \end{bmatrix}$$

  - Algebraically, it's a square matrix $\boldsymbol{A}$ that is equal to its negative transpose, i.e., $\boldsymbol{A} = -\boldsymbol{A}^T$
  - Here all elements on the diagonal must be zero, as zero is the only number that can be equal its inverse.

Identity and Zero Matrices

- **Identity matrix $\boldsymbol{I}_n$**: a matrix with size $n \times n$ with all elements along the diagonal = 1 and all other elements = 0, e.g, $\boldsymbol{I}_3$ and $\boldsymbol{I}_n$ ($\cdots$ indicate continuation of pattern):

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

  - Essentially, the identity matrix is the equivalent of the number 1 in linear algebra.
- **Zero matrix 0**: a matrix of all zeros.

Diagonal and Triangular Matrices

- **Diagonal matrix**: when all elements outside the main diagonal are zero, i.e.,

$$\begin{bmatrix} e_{1,1} & 0 & \cdots & 0 \\ 0 & e_{2,2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & e_{i,i} \end{bmatrix}$$

  - Elements along the diagonal don't have to be the same, and they can be zero (meaning rectangular matrices still can be diagonal, technically).
  - **Scaled matrix λI**: when all the elements along the main diagonal are the same and greater than 1, making it a scaled version of the identity matrix↑.
- **Triangular matrices**: when all elements above or below the diagonal are zero, but not on both sides.
  - Upper triangular matrix: when all the elements below the diagonal are zero.
  - Lower triangular matrix: when all the elements above the diagonal are zero.

$$\text{Upper} = \begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,3} \\ 0 & e_{2,2} & e_{2,3} \\ 0 & 0 & e_{3,3} \end{bmatrix} \qquad \text{Lower} = \begin{bmatrix} e_{1,1} & 0 & 0 \\ e_{2,1} & e_{2,2} & 0 \\ e_{3,1} & e_{3,2} & e_{3,3} \end{bmatrix}$$

Augmented and Complex Matrices

- **Augmented (concatenated) matrix $A \mid B$**: a matrix obtained by appending the columns of two given matrices, e.g.,

$$\left[\begin{array}{ccc} 4 & 2 & 0 \\ 3 & 7 & 6 \\ 1 & 6 & 9 \end{array}\right] \left|\begin{array}{c} 4 \\ 2 \\ 0 \end{array}\right. = \left[\begin{array}{ccc|c} 4 & 2 & 0 & 4 \\ 3 & 7 & 6 & 2 \\ 1 & 6 & 9 & 0 \end{array}\right]$$

  - Used typically for the purpose of performing the same elementary row operations↓ on each of the given matrices.
  - Matrices must have the same number of rows (or columns for vertical augmentation) for the concatenation to be applied.
- **Complex matrix**: A matrix whose elements may contain complex numbers.
- The conjugate transpose↑ discussed previously in vectors can is used here as well.
- Transposition↓ will be discussed shortly, but for now, the complex conjugate still behaves the same (just imaginary numbers change sign), e.g.,

$$\begin{bmatrix} 1 & -1+5i & 0 \\ 1 & -2 & -4 \\ 6i & -4 & 5-2i \end{bmatrix}^H = \begin{bmatrix} 1 & 1 & -6i \\ -1-5i & -2 & -4 \\ 0 & -4 & 5+2i \end{bmatrix}$$

Basic Matrix Operations

Matrix Addition

- **Matrix addition**: the operation of adding two matrices of equal dimensions $m \times n$ by adding the corresponding elements together, e.g.,

$$\begin{bmatrix} 1 & 2 & 5 \\ 0 & 6 & 8 \\ 9 & 6 & 4 \end{bmatrix} + \begin{bmatrix} 0 & 3 & 5 \\ 1 & -6 & 9 \\ -5 & -4 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 10 \\ 1 & 0 & 17 \\ 4 & 2 & 4 \end{bmatrix}$$

- Note: there are other operations which could also be considered addition for matrices, such as the direct sum and the Kronecker sum (not discussed as of now).
- ✓ Commutative: $\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}$
- ✓ Associative: $\boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C}) = (\boldsymbol{A} + \boldsymbol{B} + \boldsymbol{C})$

Matrix Scalar Multiplication

- **Matrix scalar multiplication**: the same as vector scalar multiplication↑ or simply, scalar multiplication, as vectors are $m \times 1$ (or $1 \times n$) matrices.
- Scalar multiplication is true when both left scalar and right scalar are equal, i.e.,

$$\lambda(\boldsymbol{A})_{ij} = (\lambda \boldsymbol{A})_{ij} = (\boldsymbol{A}\lambda)_{ij} = (\boldsymbol{A})_{ij}\lambda$$

- More explicitly:

$$\begin{bmatrix} \lambda e & \lambda e & \cdots & \lambda e \\ \lambda e & \lambda e & \cdots & \lambda e \\ \vdots & \vdots & \ddots & \vdots \\ \lambda e & \lambda e & \cdots & \lambda e \end{bmatrix} = \left(\lambda \text{ or } \begin{bmatrix} e & e & \cdots & e \\ e & e & \cdots & e \\ \vdots & \vdots & \ddots & \vdots \\ e & e & \cdots & e \end{bmatrix} \text{ or } \lambda \right) = \begin{bmatrix} e\lambda & e\lambda & \cdots & e\lambda \\ e\lambda & e\lambda & \cdots & e\lambda \\ \vdots & \vdots & \ddots & \vdots \\ e\lambda & e\lambda & \cdots & e\lambda \end{bmatrix}$$

- The above is true only where the underlying ring (algebraic structure that generalize fields…I need to learn more about this…) is commutative. This fact is essential for later proofs.

Transposition

- **Transpose** ^{T}: an operation where a matrix is flipped over its diagonal↑, i.e., it switches the row and column indices of the matrix, e.g.,

$$\begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 0 \\ 3 & 7 & 1 \\ 4 & 8 & 2 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 0 & 1 & 2 \end{bmatrix}$$

- Formally, the element of the i-th row, j-th column of matrix $\boldsymbol{A}$ when transposed becomes the element of the j-th row, i-th column of matrix $\boldsymbol{A}^T$, i.e.,

$$\boldsymbol{A}_{i,j} = \boldsymbol{A}^T_{j,i}$$

- Alternatively, with regard to dimensionality, if $\boldsymbol{A}$ is an $m \times n$ matrix, then $\boldsymbol{A}^T$ is an $n \times m$ matrix.
  - Thus, a transposed matrix that is transposed again will produce the original matrix, i.e.,

$$\left(\boldsymbol{A}^T_{j,i}\right)^T = \boldsymbol{A}_{i,j}$$

- Revisiting complex matrices↑:
  - **Hermitian matrix**: a square complex matrix whose transpose is equal to every entry being replaced with its complex conjugate↑.
    - Denoted: $\boldsymbol{A}^T = \overline{\boldsymbol{A}}$
  - **Skew-Hermitian matrix**: a Hermitian matrix whose transpose is equal to the negation of its complex conjugate.
    - Denoted: $\boldsymbol{A}^T = -\overline{\boldsymbol{A}}$

Diagonal and Trace

- The main diagonal↑ of a matrix can be extracted and turned into a vector.
  - Not to be confused with diagonalization↓ of a matrix, which is a result of matrix decomposition resulting from eigendecomposition↓.
- **Trace** $\operatorname{tr}(\boldsymbol{A})$: the sum of all diagonal elements, defined only for square matrices.

$$\operatorname{tr}(\boldsymbol{A}) = \sum_{i=1}^{n} e_{i,i} = e_{1,1} + e_{2,2} + \cdots + e_{n,n}$$

- The trace is a linear mapping (two vector spaces that preserves the operations of vector addition and scalar multiplication.), i.e.,

$$\operatorname{tr}(\boldsymbol{A} + \boldsymbol{B}) = \operatorname{tr}(\boldsymbol{A}) + \operatorname{tr}(\boldsymbol{B}) \qquad \operatorname{tr}(\lambda\boldsymbol{A}) = \lambda \operatorname{tr}(\boldsymbol{A})$$

- Additionally, a matrix and its transpose have the same trace, as elements along the main diagonal are not affected, i.e., $\operatorname{tr}(\boldsymbol{A}) = \operatorname{tr}(\boldsymbol{A}^T)$

Broadcasting

- **Broadcasting**: duplication of a vector so that the dimensionality matches a larger matrix, allowing for simplification of element wise addition or multiplication.
  - Technically not a valid operation in linear algebra at face value, but is used commonly in applied linear algebra and machine learning.

Matrix Multiplication

Standard Matrix Multiplication

- **Standard matrix multiplication** AB: a binary operation that produces a matrix from two matrices whose inner dimensions match.
  - Multiplication of matrices can be thought of as going from right to left ($A \leftarrow B$), or rather, A pre-multiplies B, or B post-multiplies A.
  - The number of columns (n) in the first matrix must be equal to the number of rows (m) in the second matrix.
  - **Matrix product**: the product, whose size is equal to rows of the first matrix and the number of columns of the second matrix.

$$(m_1 \times n_1)(m_2 \times n_2) = m_1 \times n_2$$

- Transposing↑ a matrix switches the dimensions, so it can enable computation in some cases, e.g,

$$\underbrace{A}_{5\times7} = \underbrace{A^T}_{7\times5} \qquad \underbrace{A^T}_{7\times5}\underbrace{B}_{5\times2} = \underbrace{C}_{7\times2}$$

- Revisiting the dot (inner) product↑ and the outer product↑ of vectors:
  - The dot product must have equal-length vectors since the transpose of left vector makes inner dimensions much match, e.g.,

$$\underbrace{v}_{5\times1}\underbrace{w}_{5\times1} \rightarrow \underbrace{v^T}_{1\times5}\underbrace{w}_{5\times1} = 1 \times 1 \text{ (scalar)}$$

  - However, the transpose of the right vector makes the 1×1 dimensions match, making the outer dimensions irrelevant to the validity of the computation, e.g.,

$$\underbrace{v}_{6\times1}\underbrace{w}_{9\times1} \rightarrow \underbrace{v}_{6\times1}\underbrace{w^T}_{1\times9} = 6 \times 9 \text{ (matrix)}$$

Standard Matrix Multiplication Perspectives

- There are four ways to compute and conceptualize the process of multiplication of two matrices, all of which give the same result.
- I.e, if A is an $m \times n$ matrix and B is an $n \times p$ matrix, then their product C =

$$\begin{bmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{bmatrix}$$

- **The element perspective**: building the product matrix directly, one element at a time, via the computation of the dot product↑ between the rows of the left matrix and the columns of the right matrix.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \begin{bmatrix}1a+2c & \cdots\\ \cdots & \cdots\end{bmatrix}$$
$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \begin{bmatrix}1a+2c & 1b+2d\\ \cdots & \cdots\end{bmatrix}$$
$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \begin{bmatrix}1a+2c & 1b+2d\\3a+4c & \cdots\end{bmatrix}$$
$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \begin{bmatrix}1a+2c & 1b+2d\\3a+4c & 3b+4d\end{bmatrix}$$

- This is generally the most common perspective as the others methods have all have two "pseudo steps" to reach the final product, rather than through one direct method.
- **The layer perspective**: the building of the product matrix one layer at a time, followed by a "flattening" to make the final product.
  - Each layer is the same size $(m_1 \times n_2)$ as the product, but is only a rank 1 matrix↓, or essentially the representation of only one column's worth of information.
  - Can be thought of as a left matrix of columns and a right matrix of rows, resulting in the computation of the outer product↑.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \begin{bmatrix}1a & 1b\\3a & 3b\end{bmatrix}$$
$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \begin{bmatrix}2c & 2d\\4c & 4d\end{bmatrix}$$
$$\begin{bmatrix}1a & 1b\\3a & 3b\end{bmatrix} \Downarrow \begin{bmatrix}2c & 2d\\4c & 4d\end{bmatrix} = \begin{bmatrix}1a+2c & 1b+2d\\3a+4c & 3b+4d\end{bmatrix}$$

- **The column perspective**: the building of the product matrix by columns, where the first column is the sum of the two columns of the left matrix weighted (scaled) by the elements of the first column of the right matrix.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \left(a\begin{bmatrix}1\\3\end{bmatrix} + c\begin{bmatrix}2\\4\end{bmatrix} \quad b\begin{bmatrix}1\\3\end{bmatrix} + d\begin{bmatrix}2\\4\end{bmatrix}\right) = \begin{bmatrix}1a+2c & 1b+2d\\3a+4c & 3b+4d\end{bmatrix}$$

- **The row perspective**: similar to the column perspective, but building up by row.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}a & b\\c & d\end{bmatrix} = \left(1\begin{bmatrix}a & b\end{bmatrix} + 2\begin{bmatrix}c & d\end{bmatrix}\right) = \begin{bmatrix}1a+2c & 1b+2d\\3a+4c & 3b+4d\end{bmatrix}$$

Properties of Matrix Multiplication

Diagonal Matrix Multiplication

- Square diagonal matrices↑ are often used to scale another matrix by the elements along such diagonal.
- You can scale the columns or rows depending on the placement of the diagonal matrix $\boldsymbol{D}$ relative to original matrix $\boldsymbol{A}$.
  - Pre-multiplication of the diagonal results in the scaling by rows, i.e., $\boldsymbol{DA}$:

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} a1 & a2 & a3 \\ b4 & b5 & b6 \\ c7 & c8 & c9 \end{bmatrix}$$

  - Post-multiplication of the diagonal results in the scaling by columns, i.e., $\boldsymbol{AD}$:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} = \begin{bmatrix} 1a & 2b & 3c \\ 4a & 5b & 6c \\ 7a & 8b & 9c \end{bmatrix}$$

- Again, when all the elements along the diagonal are the same, then it is simply a scaled version of the identity matrix↑, or sometimes referred to as a scaled matrix↑.

Order of Operations

- "And love is evol, spell it backwards, I'll show ya"
- $(\boldsymbol{LOVE})^T = \boldsymbol{E}^T\boldsymbol{V}^T\boldsymbol{O}^T\boldsymbol{L}^T$: reversing the order of multiplication on a set of matrices is valid if the same operation (e.g., transpose↑ or inverse↓) can be applied to each matrix, e.g, (diagonal highlighted for easier time seeing transpose)

$$\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)^T = \begin{bmatrix} 1a+2c & 1b+2d \\ 3a+4c & 3b+4d \end{bmatrix}^T = \begin{bmatrix} 1a+2c & 3a+4c \\ 1b+2d & 3b+4d \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 1a+2c & 1b+2d \\ 3a+4c & 3b+4d \end{bmatrix}$$

Matrix Vector Multiplication

- Multiplying a matrix by a vector always results in a vector, regardless order.
- The order of multiplication does impact the orientation and size (dimensionality) of the product vector, i.e.,
  pre-multiplication → row vector (weighted combinations of the rows of $\boldsymbol{A}$)
  post-multiplication → column vector (weighted combinations of the columns of $\boldsymbol{A}$)

$$\underbrace{\boldsymbol{w}^T}_{1\times n}\underbrace{\boldsymbol{A}}_{m\times n} = \underbrace{\boldsymbol{v}}_{1\times n} \qquad \underbrace{\boldsymbol{A}}_{m\times n}\underbrace{\boldsymbol{w}}_{m\times 1} = \underbrace{\boldsymbol{v}}_{m\times 1}$$

- The order of multiplication does not matter when multiplying a vector with a symmetric matrix↑:

$$\begin{aligned} S^T &= S && \text{symmetric matrix} \\ Sw &= v && \text{column vector} \\ (Sw)^T &= v^T && \text{apply transpose} \rightarrow \text{row vector} \\ w^T S^T &= v^T && \text{love} \leftrightarrow \text{evol}^{\uparrow} \\ w^T S &= v^T && \text{symmetric matrix} \leftrightarrow \text{transpose} \\ v^T &= v && \text{row vector} \leftrightarrow \text{column vector} \end{aligned}$$

- If $S^T \neq S$, then order does matter, resulting in different product vectors.
- Multiplication of vectors and matrices form the basis of linear transformations.
- When multiplication between a matrix and vector is equal to the multiplication between a scalar and the same vector, then the scalar is the eigenvalue↓ and the vector is the eigenvector↓.

Additive and Multiplicative Matrices

- **Multiplicative identity matrix**: commonly referred to as the identity matrix↑, where both pre and post multiplication are equal and both produce the original matrix.

$$IA = AI = A$$

  - However, addition of the multiplicative identity matrix does not yield the same product, i.e.,

$$A + I \neq A$$

- **Additive identity matrix**: the compliment to the multiplicative matrix that uses the zero matrix (matrix of all zeros), hence why it is commonly referred to as simply the zero matrix.
  - Multiplication by the zero matrix of course does not yield the original matrix (unless the original matrix was a zero matrix), but the addition of zero will yield the original product, i.e.,

$$A0 = 0A \neq A, \qquad A + 0 = A$$

Creating Symmetric Matrices

- **Additive method**: using a square matrix A added to its own transpose will create a symmetric matrix↑, i.e.,

$$S = A + A^T$$

  - Non-square matrices result in invalid matrix addition↑ due to unequal dimensions.

- Optionally, a scaling factor of $\frac{1}{2}$ can be used to make the symmetric matrix more closely resemble the original matrix since all values are doubled during addition.

- **Multiplicative method**: multiplying a matrix $\boldsymbol{A}$ of any dimension with the transpose of itself to create a symmetric matrix $\boldsymbol{S}$.
  - The order of matrix multiplication matters if the matrix is a non-square matrix, i.e.,

$$\underbrace{A^T}_{n\times m}\underbrace{A}_{m\times n}=\underbrace{S}_{n\times n} \qquad \underbrace{A}_{m\times n}\underbrace{A^T}_{n\times m}=\underbrace{S}_{m\times m}$$

  - Note, again the order of multiplication does impact the dimensionality of the symmetric matrix produced, i.e.,
    pre-multiplication $\rightarrow$ size $n \times n$ = number of columns in the original matrix
    post-multiplication $\rightarrow$ size $m \times m$ = number of rows in the original matrix

- Proving multiplicative matrices using love $\leftrightarrow$ evol rule[↑]:

$$A^TA=(A^TA)^T=A^TA^{TT}=A^TA \qquad AA^T=(AA^T)^T=A^{TT}A^T=AA^T$$

- Example of producing a symmetric matrix:

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}\begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}=\begin{bmatrix} a^2+b^2+c^2 & ad+be+cf \\ ad+be+cf & d^2+e^2+f^2 \end{bmatrix}$$

- Notice that the diagonal of the symmetric matrix is the columns of the original (or rows of transposed matrix) squared while the off diagonal elements are the cross terms (or rows of the original and columns of transpose).
- The multiplicative method is widely used signal processing and statistics; the product is often called a covariance matrix, i.e., a matrix where the diagonal elements are the variances and the off diagonal elements are the covariances.

Hadamard Multiplication

- Covered briefly in while introducing other properties of vectors[↑]; it is often denoted with either $\circ$, or $\odot$.
- To reiterate, now with more context, the Hadamard product takes two matrices with equal dimensions and multiplies each element with the corresponding element in the other matrix, i.e.,

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix}\odot\begin{bmatrix} b_{1,1} & \cdots & b_{1,n} \\ \vdots & \ddots & \vdots \\ b_{m,1} & \cdots & b_{m,n} \end{bmatrix}=\begin{bmatrix} a_{1,1}b_{1,1} & \cdots & a_{1,n}b_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1}b_{m,1} & \cdots & a_{m,n}b_{m,n} \end{bmatrix}$$

- ✓ associative, ✓ distributive, and ✓ commutative, unlike standard matrix multiplication which is not commutative.

Matrix Norms

Matrix Norms Basics

- **Matrix norm**: a vector norm↑ in a vector space whose elements are matrices.
  - More specifically, given a field K (of either real or complex numbers) and the vector space $K^{m\times n}$ of matrices of the size $m \times n$ with entries in the field K, then a matrix norm is a norm in the vector space $K^{m\times n}$.
  - Thus, a matrix norm is a function $\|\boldsymbol{A}\| : K^{m\times n} \to \mathbb{R}$ that satisfies the following properties for all scalars $\lambda \in K$ and all matrices $\boldsymbol{A}, \boldsymbol{B} \in K^{m\times n}$:
    - $\|\lambda\boldsymbol{A}\| = |\lambda|\|\boldsymbol{A}\|$ (absolutely homogenous)
    - $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$ (sub-additive or satisfying the triangle inequality)
    - $\|\boldsymbol{A}\| \geq 0$ (positive)
    - $\|\boldsymbol{A}\| = 0 \leftrightarrow \boldsymbol{A} = 0_{m,n}$ (definite)
  - **Submultiplicative norm**: special cases of square matrices that also follow the following condition:
    - $\|\boldsymbol{AB}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\|$ for all matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in $K^{n\times n}$
- There are many types of norms, and special cases for each category of norms, but for now just the Frobenius norm, the induced 2-norm, and the Schatten p-norm will be discussed.

Frobenius Norm

- **Frobenius inner (dot) product** $\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F$: binary operation that takes two matrices of equal dimensions and computes a component-wise inner product↑ of two matrices as if they were vectors. There are several ways of thinking about how the Frobenius inner product is computed, either:
  - Element-wise multiplication → summation of all elements;
  - Vectorization of both matrices → vector dot product;
    - **Vectorization** $\text{vec}(\boldsymbol{A})$: a linear transformation that converts a matrix into a column vector, i.e., a $mn \times 1$ column vector is obtained by stacking the columns of the matrix A on top of one another:

$$\text{vec}(A) = [a_{1,1}, \ldots, a_{m,1}, a_{1,2}, \ldots, a_{m,2}, \ldots, a_{1,n}, \ldots, a_{m,n}]^T$$

  - Or the must efficient way, simply taking the trace↑ of $\boldsymbol{A}^T\boldsymbol{B}$
  - Thus, the Frobenius inner product can be defined as:

$$\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F = \sum_{i,j} \boldsymbol{A}_{i,j}\boldsymbol{B}_{i,j} = \text{vec}(\boldsymbol{A})^T \text{vec}(\boldsymbol{B}) = \text{tr}(\boldsymbol{A}^T\boldsymbol{B})$$

- **Frobenius (Euclidean) norm**: the square root of the Frobenius inner product of a matrix with itself, i.e.,

$$\text{norm}(\boldsymbol{A}) = \sqrt{\langle \boldsymbol{A}, \boldsymbol{A} \rangle_F} = \sqrt{\text{tr}(\boldsymbol{A}^T \boldsymbol{A})}$$

  - This works since the use of the multiplicative method↑ create a symmetric matrix whose diagonal can be traced, resulting in a vector consisting of each element being squared then summed, thus the norm of that vector↑ can easily be taken, yielding the norm of the matrix.
  - This is often the most commonly used matrix norm and is often simply called denoted as the $\text{norm}(\boldsymbol{A})$ of a matrix.

Induced p-Norm

- **Induced p-norm** $\|\boldsymbol{A}\|_p$: Essentially measures the effect of a matrix on a particular vector norm, particularly if its longer or shorter, hence $\sup()$ in following definition.
- If p (often 2, making it commonly referred to as the 2-norm) is used for both K^n and K^m, then the induced p-norm can be defined as:

$$\|\boldsymbol{A}\|_p = \sup_{x \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p} \qquad 1 \leq p \leq \infty$$

- The notation does not differ from the Schatten p-norm or other entry-wise p-norms.

Schatten p-Norm

- **Schatten p-norm** $\|\boldsymbol{A}\|_p$: arises when applying the p-norm to the vector of singular values of a matrix↓ (for now, a special set of scalars with a certain matrix).
- If the singular values of a matrix are denoted by σ, then the Schatten p-norm is defined the sum of all singular values of matrix, i.e.,

$$\|\boldsymbol{A}\|_p = \left(\sum_{i=1}^{r} \sigma_i^p \right)^{1/p}$$

- All Schatten norms are submultiplicative.
- Often used cases are when $p = 1, 2$, and ∞;
  - $p = 2$: yields the Frobenius norm.
  - $p = \infty$: yields the induced vector 2-norm.
  - $p = 1$: yields the nuclear norm, which is the sum of all the singular values of the matrix.

Matrix Rank

Rank Terminology

- **Matrix rank** r, $\text{rank}(\boldsymbol{A})$: a non-negative integer ($\mathbb{N}_0 = 0, 1, 2, 3, \ldots$) that describes the dimension of the vector space spanned↑ by its columns.
  - Rank can also be thought of as communicating the dimensionality of the information in a matrix, but not the dimensionality of the matrix itself.
    - E.g., the columns of a 2×3 matrix are in $\mathbb{R}^3$, but one column could be embedded in a two-dimensional subspace↑, which would make it a linearly dependent set.
  - Thus, rank can also be thought of as the maximal number of linearly independent↑ columns of a matrix, which also means rank is dimension of the vector space spanned by its rows (see column and row space of a matrix↓), i.e.,

$$\text{rank}(C(\boldsymbol{A})) = \text{rank}(R(\boldsymbol{A})) = \text{rank}(\boldsymbol{A})$$

Maximum Rank

- The maximum possible rank is equal to the smaller of the two dimensions, either the rows or columns, i.e.,

$$\max(r) = r \in \mathbb{N}_0 \mid 0 \leq r \leq \min(m, n)$$

  - **Full rank**: a square matrix with maximum possible rank, making it invertible↓.
  - **Full column rank**: when $m > n$ and $\max(r) = n$.
  - **Full row rank**: when $n > m$ and $\max(r) = m$.
  - **Rank deficient (reduced rank, degenerate, low-rank, singular, non-invertible)**: when the rank is less than the maximum possible rank.

Computing Rank

- In smaller matrices, it may be easily computed by counting the number of columns in a linearly independent set.
- Row reduction↓ to the row echelon form also allows for calculation of rank by counting number of pivots.
- Compute the singular value decomposition↓ and count the number of non-zero singular values.
- Compute the eigendecomposition↓ and count the number of non-zero eigenvalues.
- Rank after addition and multiplication:
  - $\text{rank}(\boldsymbol{A} + \boldsymbol{B}) \leq \text{rank}(\boldsymbol{A}) + \text{rank}(\boldsymbol{B})$

- $\text{rank}(\boldsymbol{AB}) \leq \min(\text{rank}(\boldsymbol{A}), \text{rank}(\boldsymbol{B}))$
- Key point: the rank is not necessarily the same as either of the original matrices.

Rank of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$

- The rank of a matrix when multiplied with itself is the same as the rank of the original matrix (and the transpose of said matrix), regardless of pre/post-multiplication, i.e.,

$$\text{rank}(\boldsymbol{A}) = \text{rank}(\boldsymbol{A}^T) = \text{rank}(\boldsymbol{A}^T\boldsymbol{A}) = \text{rank}(\boldsymbol{A}\boldsymbol{A}^T)$$

- Explanation 1, using column spaces↓:
  - Columns c_j are combinations of columns in A^T, i.e., combining rows of $\boldsymbol{A}$ results in column c_j of product matrix $\rightarrow \boldsymbol{A}^T a_j = c_j$
  - This means the dimensionality of the subspace spanned by the columns of $\boldsymbol{C}$ is the same as the dimensionality subspace spanned the columns of $\boldsymbol{A}^T$, which means the rank is the same↑, i.e.,

$$C(\boldsymbol{C}) = C(\boldsymbol{A}^T) \rightarrow \text{rank}(\boldsymbol{C}) = \text{rank}(\boldsymbol{A}^T)$$

- Explanation 2, using null space↓:
  - (to be explained)
- Explanation 3, using singular value decomposition↓:
  - (to be explained)
- The implications of this fact show that creating symmetric full rank matrices, via the multiplicative method↑, of desired size (pre=column sized, post=row sized) can easily be done.

Full Rank via "Shifting"

- First, "shifting" a matrix is different from a shift matrix, which may or may not be explicitly defined, for now.
- **"Shifting" $\tilde{\boldsymbol{A}}$**: a binary operation that involves the addition of a square matrix with a scaled identity matrix↑ in order to create a full rank matrix, i.e.,

$$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \lambda \boldsymbol{I}$$

- E.g., below is a rank deficient matrix (rank 2) that has been "shifted" by a slightly scaled identity matrix in order to create the full rank (3):

$$\begin{bmatrix} 1 & 3 & 3 \\ 5 & -7 & -7 \\ -5 & 2 & 2 \end{bmatrix} + 0.1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.01 & 3 & 3 \\ 5 & -6.99 & -7 \\ 5 & 2 & 2.01 \end{bmatrix}$$

- Determining the correct scaling factor without changing the information in the original matrix can be very difficult, but generally small shifts are better.

Matrix Spaces

Column Space

- **Column space (range, image)** $C(\boldsymbol{A})$: the vector subspace spanned (all possible weighted linear combinations) by the columns of a matrix $\boldsymbol{A}$, i.e.,

$$C(\boldsymbol{A}) = \{\lambda_1 a_1 + \lambda_2 a_2 + \cdots + \lambda_n a_n\} \quad \lambda \in \mathbb{R}$$

- Can also be expressed as the span of all the columns in a matrix, i.e.,

$$C(\boldsymbol{A}) = \text{span}(\{a_1, \ldots, a_n\})$$

  - **Image**: the set of all output values of a function.
  - The column space is often called the image of a matrix since the span of all columns of a corresponding linear transformation is the image of that transformation.
- A common and important question in applied linear algebra is determining whether a vector is in a particular column space, and if not, then how close is it.
  - Useful tools include reduced row echelon form↓ and least squares algorithm↓.

Row Space

  - **Row space** $R(\boldsymbol{A})$, $C(\boldsymbol{A}^T)$: the vector subspace spanned by all the rows of a matrix.
  - The row space is very similar to the column space, but does have some differences.
    - When talking about the column space of a matrix, you need to post-multiply by a column vector, while the row space requires pre-multiplication of the matrix with a row vector, i.e.,

$$\boldsymbol{A}x = \boldsymbol{b} \qquad x^T \boldsymbol{A} = \boldsymbol{b}$$

    - The difference may be trivial at first, but row spaces and column spaces are often different when using real data; if you try to filter the data (contained in the matrix) with a vector, then the distinction here matters.
  - Using Elementary operations↓ on a matrix will not change its row space, but it can (not always) change its column space.
    - Matrices can be taken to row echelon form↓ using such operations, thus it is possible to find a basis↑ for the row space.
    - If a matrix is taken to the reduced row echelon form↓, then one can find a unique basis for the span of the set of vectors determined by the row space.
  - For a matrix that represents a homogeneous system of linear equations↓, the row space consists of all linear equations that follow from those in the system.

Null Space

- **Null space (kernel)** $N(\boldsymbol{A})$: the set of all vectors $\boldsymbol{x}$ (excluding trivial case; $\boldsymbol{x} \neq \boldsymbol{o}$) for which $\boldsymbol{Ax} = \boldsymbol{o}$, where $\boldsymbol{o}$ is the zero vector.
  - **Empty set** $\{\ \}$: when there is no vector $\boldsymbol{x}$, besides the trivial case, such that $\boldsymbol{Ax} = \boldsymbol{o}$
- A matrix with an empty set contains linearly independent columns↑, while a matrix that contains a non-empty null space must have at least two linearly dependent columns.
- Any basis vector, or set of basis vectors, can be chosen for the null space, as any scalar applied to any basis can be used to represent any other vector in the null space.

Left Null Space

- **Left null space (cokernel)** $N(\boldsymbol{A}^T)$: similar to the null space (colloquially sometimes the same), except it is pre-multiplied (hence the name) by a row vector and yields the zero row vector, i.e.,

$$N(\boldsymbol{A}^T) = \boldsymbol{x}^T\boldsymbol{A} = \boldsymbol{o}^T$$

- Typically represented as the null space of the transposed matrix (hence the notation) i.e.,

$$N(\boldsymbol{A}^T) = \boldsymbol{A}^T\boldsymbol{x} = \boldsymbol{o}$$

- The null space can be thought of a void, that if vector is sent to, cannot be escaped.
- Geometrically, if $x \notin N(\boldsymbol{A})$, then the vector can be transformed by some matrix to any non-zero length.
  - Alternatively, if $x \in N(\boldsymbol{A})$, then the only transformation possible is a transformation to the origin, i.e., the zero vector.

Four Fundamental Subspaces

- **Four fundamental subspaces**: the four subspaces that are associated with a matrix, i.e., the column space, row space, null space, and left null space.
- The column space and left null space must be orthogonal↑, i.e., the dot product with any linear combination of the columns of matrix $\boldsymbol{A}$ must be zero:

$$\boldsymbol{x}^T\{\lambda_1 a_1 + \cdots + \lambda_n a_n\} = 0$$

- Likewise, the row space and the null space also must be orthogonal, i.e., the dot product with any linear combination of the rows of matrix $\boldsymbol{A}$ must be zero:

$$\boldsymbol{x}^T\{\lambda_1 a_1^T + \cdots + \lambda_m a_m^T\} = 0$$

  - This implies if the column space spans all of $\mathbb{R}^m$, then the left null space is empty.
  - Likewise, if the row space spans all of $\mathbb{R}^n$, then the null space is empty.

Dimensionality of the Subspaces

- **Nullity**: the dimensionality of the null space of a matrix.
- The dimensionality of the column or row space of a matrix is equal to the rank↑ of the matrix.
- **Rank-nullity theorem**: the dimension of the domain of a linear map is the sum of its rank and the nullity, i.e., the dimensionality of the column space and the nullity of the left null space must equal the rows of the original matrix:

  $$\dim(C(\boldsymbol{A})) + \dim(N(\boldsymbol{A}^T)) = m$$

  Likewise, the dimensionality of the row space and the nullity of the null space must equal the columns of the original matrix:

  $$\dim(C(\boldsymbol{A}^T)) + \dim(N(\boldsymbol{A})) = n$$

- An example that illustrates the four subspaces and their dimensionality:

  $$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 3 & -3 \end{bmatrix} \quad (2 \times 3)$$

  $$\downarrow$$

  $$C(\boldsymbol{A}) = \left\{ \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right\} \in \mathbb{R}^2 \quad \dim(C(\boldsymbol{A})) = 2$$
  $$N(\boldsymbol{A}^T) = \{ \quad \} \in \mathbb{R} \quad \dim(N(\boldsymbol{A}^T)) = 0$$

  $$2 + 0 = 2$$

  $$C(\boldsymbol{A}^T) = \left\{ \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}^T, \begin{bmatrix} 3 \\ 3 \\ -3 \end{bmatrix}^T \right\} \in \mathbb{R}^3 \quad \dim(C(\boldsymbol{A}^T)) = 2$$
  $$N(\boldsymbol{A}) = \left\{ \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \right\} \in \mathbb{R}^3 \quad \dim(N(\boldsymbol{A})) = 1$$

  $$2 + 1 = 3$$

  - Here you can see the rank is the same, but the columns are linearly dependent, leading to a non-zero nullity.

Systems of Linear Equations

Linear Equations

- **Linear equation**: an equation that may be put in the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n + b = 0$$

  - **Coefficients (parameters)** $a_1, a_2, \ldots, a_n$: typically real numbers; sometimes they can be arbitrary expressions as long as they don't contain any new variables, though often $\{x_{n+1}, x_{n+2}, x_{n+3}, \ldots\}$ are replaced with $\{s, r, t, \ldots\}$ for convenience.
  - **Variables** $x_1, x_2, \ldots, x_n$: the unknowns of the equation.
  - **Constant term** b: typically a real number; technically also a coefficient, but not attached to any variables.
- **System of linear equations**: a finite collection of linear equations involving the same set of variables, e.g.,

$$\begin{cases} 3x + 2y - z & = 1 \\ 2x + 2y + 4z & = -2 \\ -x + \frac{1}{2}y - z & = 0 \end{cases}$$

  - Note: some variables can have a coefficient of 0, so some equations may appear to not contain the same set at first glance.
  - Together, all individual equations are used to indicate one object, hence "system."

Solutions

  - **Solution**: an assignment of values to variables such that all assignments yield a valid equation, i.e.,

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$$

  - **Liner system solution**: an assignment of values to the variables such that all equations are simultaneously satisfied, e.g., using the above system of equations:

$$\begin{aligned} x &= 1 \\ y &= -2 \\ z &= -2 \end{aligned} \qquad \begin{aligned} 3(1) + 2(-2) - (-2) &= 1 \\ 2(1) + 2(-2) + 4(-2) &= -2 \\ -(1) + \tfrac{1}{2}(-2) - (-2) &= 0 \end{aligned}$$

  - **Solution set**: the set of all possible solutions that satisfy all variables in a system of linear equations; there are three possibilities for such system:

    1. The system has infinitely many solutions (the lines are the same).
    2. The system has a single unique solution (the lines intersect at single point).

3. The system has no solution (the lines are parallel).

- **Inconsistent**: a system that has no solution.
- **Consistent**: a system that has at least one solution.

Elementary Operations

- Solve system of equations is easily done if such systems are first converted to a form of matrix multiplication, i.e., taking this general form of a system of equations:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n & = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n & = b_2 \\ & \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n & = b_m \end{cases}$$

  and then turning it into the matrix vector multiplication↑ form, specifically the post-multiplication form in order to create a column vector of constants:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

- Augmented matrices↑ are a useful intermediate that is used to represent the final product and perform elementary operations on.
  - Creating an augmented matrix can easily be done by taking the above form, dropping the variables temporarily, then concatenating the vector of constants onto the matrix of coefficients, i.e.,

$$\left[\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & b_m \end{array}\right]$$

- **Elementary operations**: a set of operations that can be performed on a matrix while maintaining equivalence (when two systems have the same set of solutions); the operations are as follows:
  1. Interchange two rows: $R_i \updownarrow R_j$
  2. Multiply one row by a nonzero number: $kR_i \to R_i,\ k \neq 0$
  3. Add a multiple of one row to a different row: $R_i + kR_j \to R_i,\ i \neq j$

Gaussian Elimination

- **Gaussian elimination (row reduction)**: an algorithm for solving systems of linear equations by using a sequence of elementary operations on a matrix until the row echelon form is obtained.
  - **Gauss-Jordan elimination**: when Gaussian elimination is used until a matrix reaches the reduced row echelon form; sometimes it is computationally beneficial to stop and before such form is reached.
  - The reduced row echelon form is unique, unlike the row echelon form, i.e., it is independent of the sequences of row operations used.

Row Echelon form

- **Row echelon form**: a matrix that has been converted into a pseudo upper triangular matrix↑ using Gaussian elimination, e.g. (* = any number; can be zero),

$$\begin{bmatrix} \mathbf{1} & * & * & * & * & * \\ 0 & 0 & \mathbf{1} & * & * & * \\ 0 & 0 & 0 & \mathbf{1} & * & * \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

  - More specifically, a matrix is in row echelon form if:
    - All rows consisting of only zeroes are on the bottom.
    - **Pivots**: the leading coefficient of a nonzero row is always strictly to the right of the leading coefficient (sometimes required to be 1) of the row above it.
  - The row echelon form allows for determination of the matrix rank↑ by counting the number of pivots, e.g., the above example is rank 4.
    - If rank (r) = number of columns (n), then it has a unique solution.
    - If $r < n$, then it has infinite solutions.
- **Reduced row echelon form** $\operatorname{rref}(M)$: a matrix that satisfies all the requirements of the row echelon form while additionally satisfying the following:
  - The leading coefficient of each nonzero row must be 1 (often called a leading 1).
  - Columns containing a leading 1 have zeros in all other entries (below and above).
  - E.g., using the row echelon form above:

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

- **Parameters**: the nonleading variables ($*x_n$: the leftover variables associated with nonzero elements) are replaced with new variables $\{s, r, t, \dots\}$ when converting back systems of equations, hence why coefficients↑ are sometimes referred to as parameters.
- Any matrix can be brought to the (reduced) row echelon form by a sequence of elementary row operations; this theorem is the basis of the Gaussian algorithm.

The Gaussian Algorithm

1. Create an augmented matrix from a system of linear equations in order to perform elementary operations more easily.
   - Steps for how to do this is under elementary operations↑, as well valid operations for step 2.
2. Use Gaussian/Gaussian-Jordan elimination until the row echelon form/rref is obtained; the majority of the work involves this step (assuming not a zero matrix).
   2.1 Find the first column from the left containing a nonzero entry (call it a).
   2.2 Create a leading 1 in the top row by multiplying by a^{-1}.
   2.3 Subtract multiples of row that row from rows below; make each entry below the leading 1 zero.
   2.4 Repeat steps 2.1–2.3, on the next row.
      - Sometimes interchanging rows can make this process more efficient, look for invalid pivots to save computational effort.
   2.5 Stop when either no rows remain in step 4, or the remaining rows consist of entirely of zeros to obtain the row echelon form, continue to obtain rref.
   2.6 Repeat step 2.3, but now subtracting multiples of that row from rows above the row; make each entry above the leading 1 zero.
3. There a few different possibilities once solved, i.e.,
   - If a row with the form $\left[\begin{array}{ccc|c} 0 & 0 & 0 & \cdots & a \neq 0 \end{array}\right]$ occurs, then the system is inconsistent, i.e., at least one line is parallel with another—no solution possible.
   - If a row with all zeros occurs, then that row added no new information and was simply a multiple of another row.
   - If reduced form is found, then continue.
4. Map the matrix back to the equations.
5. Back-substitution of the nonleading coefficients (parameters) to solve for leading coefficients.
6. Double check for validity by using leading variables in the original system.

Homogeneous Equations

- **Homogeneous systems**: when a system of linear equations have zeros in all the constant terms, i.e.,

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= 0 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= 0 \\ &\vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{2,n}x_n &= 0 \end{aligned}$$

- Equivalent to a matrix equation of the form $\boldsymbol{A}\boldsymbol{v} = 0$.

Homogeneous Solution Sets

- **Trivial solution**: obtained by assigning the value of zero to each of the variables.
- If the system has a non-singular matrix↓, then it is also the only solution.
- If the system has a singular matrix↓ then there are infinite solutions and consists of a solution set with the following properties:
  - If $\boldsymbol{u}$ and $\boldsymbol{v}$ are two vectors representing solutions, then the vector sum $\boldsymbol{u} + \boldsymbol{v}$ is also a solution.
  - If $\boldsymbol{u}$ is a vector representing a solution, and λ is any scalar, then $\lambda\boldsymbol{u}$ is also a solution.
  - These are exactly the same properties required for the solution set to be a subspace↑ of $\mathbb{R}^n$.
  - Numerical solutions to a homogeneous can be found with singular value decomposition↓.

- There is a close relationship between the solutions to a linear system and the solutions to the corresponding homogeneous system:

$$\boldsymbol{A}\boldsymbol{x} = b \quad \text{and} \quad \boldsymbol{A}\boldsymbol{x} = 0$$

  - Specifically, if $\boldsymbol{p}$ is any specific solution to the linear system $\boldsymbol{A}\boldsymbol{x} = b$, then the entire solution set can be described as

$$\{\boldsymbol{p} + \boldsymbol{v} : \boldsymbol{v} \text{ is any solution to } \boldsymbol{A}\boldsymbol{x} = 0\}$$

  - Geometrically, $\boldsymbol{A}\boldsymbol{x} = b$ is simply translation of the solution set for $\boldsymbol{A}\boldsymbol{x} = 0$.
  - The above only applies if $\boldsymbol{A}\boldsymbol{x} = b$ has at least one solution.
  - There is much more that relates to the relationship between there two equations; this relationship will be revisited multiple times.

The Determinant

Determinant Basics

- **Determinant** $\det(\boldsymbol{A})$, $|\boldsymbol{A}|$: a scalar value that is a function of the entries of a square matrix that allows for characterization various properties of a matrix and its linear mapping.
  - The determinant is nonzero if and only if the matrix is invertible↓, and the linear map represented by the matrix is an isomorphism.
    - **Isomorphism**: when a linear map is bijection, i.e., a function between two sets with a one-to-one mapping.
  - I.e., $\det(A) = 0$ if the matrix contains a set of linearly dependent columns (or rows), making it singular, or non-invertible↓.
- There are methods for quickly calculating the determinant of 2×2 and 3×3 matrices, as well various formulas for calculating $n \times n$ matrices, but these are not described here (as of now).
- Geometrically, the determinant represents the signed ($\pm$) n-dimensional volume of an n-dimensional parallelepiped.
  - If the value is degenerate (not fully n-dimensional), then it indicates that the image↑ of the matrix is less than n.
- Determinants can also be used for defining the characteristic polynomial of a matrix, whose roots are the eigenvalues↓.

Properties of the Determinant

- $\det(I) = 1$, where I is an identity matrix.
- $\det(\boldsymbol{A}^T) = \det(\boldsymbol{A})$
- $\det(\boldsymbol{A}^{-1}) = \dfrac{1}{\det(\boldsymbol{A})} = [\det(\boldsymbol{A})]^{-1}$, often used to help find the inverse↓.
- $\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$, if $\boldsymbol{A}$ and $\boldsymbol{B}$ are of equal size.
- $\det(c\boldsymbol{A}) = c^n \det(\boldsymbol{A})$
- There are more properties, more of which may or may not be added later to these notes.

Matrix Inverse

Inverse Basics

- **Invertible (nonsingular, nondegenerate) matrix A^{-1}**: an n-by-n square full rank$^{\uparrow}$ matrix whose product with another matrix (its multiplicative inverse here) yields the identity matrix$^{\uparrow}$ of equal size, i.e.,

$$AA^{-1} = A^{-1}A = I_n$$

- **Singular (degenerate, non-invertible) matrix**: a matrix that is not invertible.
  - A square matrix is singular if and only if its determinant$^{\uparrow}$ is zero.
- The matrix inverse is side-dependent, multiplication by the inverse must be applied to the same side of the equation.
- Non-square matrices do not have an inverse, however, in some cases a matrix may have a left or right inverse since a full rank square matrix can be created$^{\uparrow}$ via the multiplicative method if the original matrix is either full column/row rank$^{\uparrow}$.
  - **Left inverse**: when $\boldsymbol{A}$ is a full column rank matrix, then $\boldsymbol{A}^T\boldsymbol{A}$ will yield a full rank, square matrix—making it invertible; the resulting identity matrix is I_n
  - **Right inverse**: when $\boldsymbol{A}$ is a full row rank matrix, then $\boldsymbol{A}\boldsymbol{A}^T$ will yield a full rank, square matrix—making it invertible; the resulting identity matrix is I_m
- A rank deficient matrix does not have an inverse, however, a pseudoinverse$^{\downarrow}$ can be created. The pseudoinverse is widely used in and will be covered later after singular value decomposition$^{\downarrow}$ is discussed.

Properties of Invertible Matrices

- $(\boldsymbol{A}^{-1})^{-1} = \boldsymbol{A}$
- $(\lambda\boldsymbol{A})^{-1} = \lambda^{-1}\boldsymbol{A}^{-1}$ for nonzero scalar λ
- $(\boldsymbol{A}\boldsymbol{x})^{\dagger} = \boldsymbol{x}^{\dagger}\boldsymbol{A}^{-1}$ if $\boldsymbol{A}$ has orthonormal columns and † denotes the Moore-Penrose inverse (pseudoinverse).
- $(\boldsymbol{A}^T)^{-1} = (\boldsymbol{A}^{-1})^T$
- If $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_k$ are invertible, then $(\boldsymbol{A}_1\boldsymbol{A}_2 \ldots \boldsymbol{A}_{k-1}\boldsymbol{A}_k)^{-1} = \boldsymbol{A}_1^{-1}\boldsymbol{A}_2^{-1} \ldots \boldsymbol{A}_{k-1}^{-1}\boldsymbol{A}_k^{-1}$
- $\det(\boldsymbol{A}^{-1}) = (\det(\boldsymbol{A}))^{-1}$
- Invertible matrix theorem: a list of equivalent statements for invertible matrices. I might revisit and expand on this if my understandings of inverse matrices is failing.

Computing the Inverse

- For now, the methods of how the inverse is created is not of importance to me, so only a basic overview will be discussed. Again, this subject may be revisited.

Projections and Orthogonalization

Projections

- **Projection**: an idempotent linear transformation P from a vector space to itself that results in the original transformation, i.e.,

$$P : V \to V \mid P^2 = P$$

  - **Idempotent**: a property of certain operations whereby they can be applied multiple times without changing the results.

- **Orthogonal projection**: a projection on V, when V is a Hilbert space, that satisfies:

$$\langle P\boldsymbol{x}, \boldsymbol{y} \rangle = \langle P\boldsymbol{x}, P\boldsymbol{y} \rangle = \langle \boldsymbol{x}, P\boldsymbol{y} \rangle \; \forall \, \boldsymbol{x}, \boldsymbol{y} \in V$$

  - **Hilbert space**: a generalized notion of Euclidean space that extends methods from $\mathbb{R}^2$ and $\mathbb{R}^3$ to space with any finite or infinite number of dimensions.
    - This means an orthogonal projection↑, with regard to vector space, V has an inner product↑ (denoted $\langle \cdot, \cdot \rangle$ here) and has enough limits in the space (i.e., it's complete) to allow the techniques of calculus and vector algebra to be used.
  - Essentially, an orthogonal projection is one that describes a mapping of one vector to another divided by the magnitude of the vector being mapped to.
  - E.g., in $\mathbb{R}^2$ a vector $\boldsymbol{y}$ can be projected onto vector some scaled vector $\beta\boldsymbol{x}$; the orthogonal projection occurs when the dot product between vector $\boldsymbol{x}$ and distance $\boldsymbol{y}$ from the scaled vector $\boldsymbol{x}$ $(\boldsymbol{y} - \beta\boldsymbol{x})$ is equal to zero, i.e., $\boldsymbol{x}^T(\boldsymbol{y} - \beta\boldsymbol{x}) = 0$

    Solving for β results in the described mapping over magnitude:

$$\beta = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\boldsymbol{x}^T\boldsymbol{x}}$$

    Which is often described as the projection of $\boldsymbol{y} \to \boldsymbol{x}$ is a scaled version of $\boldsymbol{x}$, which is equivocal to the generalized form above, and the common notation in $\mathbb{R}^2$, i.e.,

$$\text{proj}_x \, \boldsymbol{y} = \beta\boldsymbol{x}$$

- **Oblique projection**: a projection on a Hilbert space that is not orthogonal, often used to represent spatial figures in 2-D drawings or calculating the fitted value of instrumental variables in regression, though much less common than orthogonal projections.
  - Let vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ form a basis for the range of the projection, and assemble them into matrix $\boldsymbol{A}$.
  - Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ form a basis for the orthogonal complement of the null space of the projection, and assemble them into matrix $\boldsymbol{B}$.

- Then the projection is defined by:

$$P = A(B^T A)^{-1} B^T$$

Finding Projections

- The example for orthogonal projections can be generalized for $\mathbb{R}^N$.
- Let V be a vector space spanned by orthogonal vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p$, and let $\boldsymbol{y}$ be a vector.
- One can now define a projection of $\boldsymbol{y}$ on V as:

$$\text{proj}_V \boldsymbol{y} = \frac{\boldsymbol{u}^i \boldsymbol{y}}{\boldsymbol{u}^i \boldsymbol{u}^i} \boldsymbol{u}^i$$

  where repeated indices^i are summed over.
- The vector $\boldsymbol{y}$ can be written as an orthogonal sum such that:

$$\hat{\boldsymbol{y}} = \text{proj}_V \boldsymbol{y} + z \, \text{proj}_V \boldsymbol{y}$$

  where $\boldsymbol{z}$ is the shortest distance from $\boldsymbol{y} \to V$. This is commonly used in areas such as machine learning.

Orthogonalization

Orthogonal Matrices

- **Orthogonal matrix** Q: a matrix whose rows and columns are orthonormal vectors.
  - **Orthonormal vectors**: two vectors that are orthogonal↑ unit vectors↑.
  - **Orthonormal set**: all vectors that are mutually orthogonal and all of unit length.
  - **Orthonormal basis**: an orthonormal set which forms a basis↑.
- In other words, orthogonal matrices have columns that are all pairwise orthogonal, i.e., all the dot products between any two columns are orthogonal.
- Additionally, each column has as magnitude of 1 since the norm↑ = 1.
- Thus, algebraically, orthogonal matrices can be defined as:

$$\langle \boldsymbol{Q}_i, \boldsymbol{Q}_j \rangle = \begin{cases} 1, \text{ if } i = j \\ 0, \text{ if } i \neq j \end{cases}$$

  Which can be compacted further to express even in simpler notation:

$$\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$$

  This means that $\boldsymbol{Q}^T$ is also the inverse of $\boldsymbol{Q}$, leading to the equivalent characterization:

$$\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{Q}^{-1} \boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{Q}\boldsymbol{Q}^{-1} = \boldsymbol{I}$$

- Note: The above is true for square full rank matrices, but the left or right sided inverses↑ can be applied in some cases to make those portions valid, respectively.

Gram-Schmidt Process

- **Orthogonalization**: the process of finding a set of orthogonal vectors that span a particular subspace.
  - Every vector in the new set is orthogonal to every other vector in the new set; the new set and the old set have the same linear span↑.
- **Orthonormalization**: the process normalizing each vector in order to make them all unite vectors.
- Orthonormalization is typically included within orthogonalization, and together they satisfy the conditions needed in order to create orthogonal matrices↑.
- **Gram-Schmidt process**: the method of Orthonormalizing a set of vectors in an inner product space.
  - The process takes finite, linearly independent set of vectors $S = \{v_1, \dots, v_k\}$ and uses the methods of finding projections↑ to create an orthogonal set $S' = \{u_1, \dots, u_k\}$ that spans the same k-dimensional subspace of $\mathbb{R}^n$ as S.
  - There are other algorithms for orthogonalization, that offer certain benifites in various cases, but the main point here is that the application of such process yields the QR decomposition.

QR Decomposition

- **QR decomposition (factorization)**: a decomposition of a matrix $\boldsymbol{A}$ into a product $\boldsymbol{A} = \boldsymbol{QR}$ of an orthogonal matrix $\boldsymbol{Q}$ and an upper triangular matrix↑ $\boldsymbol{R}$.
  - Any information that was lost during orthonormalization in order to create $\boldsymbol{Q}$ is captured by $\boldsymbol{R}$, yielding a lossless transformation.
  - QR decomposition is often used to solve the linear least squares problem↓ and is the basis for a particular eigenvalue algorithm, the QR algorithm↓.
- Finding $\boldsymbol{R}$ is easy, since the orthogonal matrix $\boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$, thus:

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{QR} \\ \boldsymbol{Q}^T\boldsymbol{A} &= \boldsymbol{Q}^T\boldsymbol{QR} \\ \boldsymbol{Q}^T\boldsymbol{A} &= \boldsymbol{R} \end{aligned}$$

- $\boldsymbol{Q}$ is always size $m \times m$ with a rank = m.
- $\boldsymbol{A}$ is always size $m \times n$, but whose rank depends on the rank of $\boldsymbol{A}$.
- Finding the inverse of A can be done with QR decomposition, since finding the inverse of an upper triangular matrix is computationaly simple, i.e.,

$$\boldsymbol{A}^{-1} = (\boldsymbol{QR})^{-1} = \boldsymbol{R}^{-1}\boldsymbol{Q}^{-1} = \boldsymbol{R}^{-1}\boldsymbol{Q}^T$$

Least-Squares and Model-Fitting

Model-Fitting

- **Model fitting**: the combination of fixed features and free parameters in such a way that fits experimental data to a mathematical model based on adjustments to free parameters.
  - **Fixed features**: components on the imposed on the model based on previous knowledge, understanding, theories, hypotheses, or other evidence.
  - **Free parameters**: variables that cannot be predicted precisely or constrained by the model; they must be adjusted or estimated, which is the main goal of model fitting.
- **Model interpolation**: the prediction of other experimental results given prior experimental data and a fitted model based on those data.

Five Steps to Model-Fitting

1. Define the equation(s) underlying the model.
   - This step has less to do with linear algebra and more to do with data availability and theories around the fixed features a system.
   - For example, height is governed by a complex interaction between genetics and the environment, but a simplistic model can be made to estimate the importance of important features;

     $$h = \beta_1 s + \beta_2 p + \beta_3 n$$

     h = height, s = sex, p = parents' height, n = nutrition, and β = free parameters.
2. Map the data to the model equations.
   - This step is where one takes real, or simulated, data and maps them to each of the fixed features.
   - The end result produces a system of equations with a series of unknown parameters.
   - One other component—the **intercept,** ε—is often included to capture the expected value of the data when all the predictors are zero and account for global offsets.
3. Convert the equations into a matrix-vector equation.
   - This process was previously covered when describing elementary operations[↑] and involves converting the system of equations into a form of matrix-vector multiplication, i.e., $\boldsymbol{Ax} = \boldsymbol{b}$
4. Computer the parameters.

- There are numerous ways to compute the parameters (the focus of this chapter), but in general this step involves the actual fitting of the model via estimation of the parameters.

5. Statistical evaluation of the model.
   - This step uses inferential statistics, which is outside the scope of linear algebra, but is deeply connected. Thus, this step won't be discussed; most of the following information is to understand linear algebra's role in the process.
   - Statistics does have a different nomenclature, so it is useful to briefly cover the differences, i.e.,
     - $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$: the general linear model, sometimes with no $\boldsymbol{\varepsilon}$.
     - $\boldsymbol{X}$: the design matrix, where columns = independent variables/predictors/regressors.
     - $\boldsymbol{\beta}$: the vector of regression coefficients/beta parameters.
     - $\boldsymbol{y}$: vector of the dependent variables/outcome measure.
     - Essentially, $\boldsymbol{X\beta} = \boldsymbol{y}$ is equivalent to $\boldsymbol{Ax} = \boldsymbol{b}$

Least-Squares

- **Least-squares**: a standard approach in regression analysis to approximate the solution of overdetermined systems (systems of questions with more equations than unknowns) by minimizing the sum of the squares of the residuals, i.e.,

$$\|\boldsymbol{\varepsilon}\|^2 = \|\boldsymbol{X\beta} - \boldsymbol{y}\|^2$$

  - **Residuals** $\boldsymbol{\varepsilon}$: the vector that describes the difference between the observed value $\boldsymbol{y}$ and the estimated value $\boldsymbol{X\beta}$; $\hat{\boldsymbol{y}}$ provided by a model, i.e.,

$$\boldsymbol{\varepsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y} = \boldsymbol{X\beta} - \boldsymbol{y}$$

    Here, $\boldsymbol{\varepsilon}$ is the equivalent to $\boldsymbol{z}$—the orthogonal sum—described in projections ↑.
  - The goal is to minimize this difference, and the magnitude of such difference is heavily used in machine learning and statistics.
  - The only variables that can be changed in order to minimize the residuals are the β parameters, which can be done in multiple ways. Here we focus on tools and methods from linear algebra that allow us to perform the least-squares approach.

Least-Squares via Left Inverse and Orthogonal Projections

- Since the design matrix $\boldsymbol{X}$ is an overdetermined system, then the left inverse↑ can be used to isolate the regression coefficients if $\boldsymbol{X}$ has full column rank↑, i.e.,

$$\begin{aligned} Ax = b \rightarrow X\beta &= y \\ (X^T X)^{-1} X^T X\beta &= (X^T X)^{-1} X^T y \\ \beta &= (X^T X)^{-1} X^T y \end{aligned}$$

- Finding orthogonal projections↑ in arbitrary dimensions, $\mathbb{R}^N$, yields the following:

$$x = (A^T A)^{-1} A^T b$$

- Which is exactly the same as the solution to the regression coefficients,

$$\beta = (X^T X)^{-1} X^T y$$

- Where the key takeaway here is that $X\beta$, after the application of least-squares, produces an estimated value, $\hat{y}$. However, we start with the observed values y, and this difference between $\hat{y}$ and y is ε.
- This gives us the general form, since most models do not perfectly describe the observed data.

$$X\beta = y + \varepsilon$$

- Taking the squared magnitude values allows for an efficient regression (due to methods in calculus) towards to minimal values acceptable in β, yielding the original definition:

$$\|\varepsilon\|^2 = \|X\beta - y\|^2$$

Eigendecomposition

Eigendecomposition Fundamentals

- **Eigendecomposition**: the factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its eigen**values** and eigen**vectors**.
  - Only defined for square matrices.
  - Singular value decomposition$^{\downarrow}$ works for any $m \times n$ matrix.
- For an $n \times n$ matrix, there are n eigenvalues and n eigenvectors.
  - Each eigenvalue has an associated eigenvector, with the possibility of there being both distinct$^{\downarrow}$ and repeated$^{\downarrow}$ eigenvalues.
- **Eigenvector $\boldsymbol{v}$**: a nonzero vector that changes at most by a scalar$^{\uparrow}$ when a linear transformation is applied to it.
- **Eigenvalue λ**: the corresponding factor by which the eigenvector is scaled.
- Formally, if T is a linear transformation from vector space V over a field F into itself and $\boldsymbol{v}$ is a nonzero vector in V, then $\boldsymbol{v}$ is an eigenvector of T if $T(v)$ is a scalar multiple λ of $\boldsymbol{v}$, i.e.,

$$T(v) = \lambda \boldsymbol{v}$$

- **Eigenvalue equation**: if V is a finite-dimensional, then the above is equivalent to

$$\boldsymbol{A}\boldsymbol{u} = \lambda \boldsymbol{u}$$

  where $\boldsymbol{A}$ is the matrix representation of T and $\boldsymbol{u}$ is the coordinate vector (vector in terms of particular ordered basis) of $\boldsymbol{v}$.
- Essentially, this is useful as a single eigenvalue λ can represent an entire matrix $\boldsymbol{A}$ given the associated eigenvector $\boldsymbol{v}$. Thus, finding the eigenvectors allows for a set of basis vectors (principal axis) that can be used to represent a dataset, often in a more efficient way.
  - E.g., it can be very useful as each data point can be represented as a vector, leading to the application of a linear transformation that will not change the span$^{\uparrow}$ of the data, but instead will efficiently scale the data along the eigenvectors.
  - Alternatively, there are often various patterns within datasets that can be much more apparent when organized along new axes, whereby eigenvectors are the means of such reorganization.
- **Eigensystem**: the set of all eigenvectors of a linear transformation, each paired with corresponding eigenvalue.
- **Eigenspace**: the set of all eigenvectors of T corresponding to the same eigenvalue (and zero vector).

- **Eigenbasis**: the set of eigenvectors of T that forms a basis[↑] of the domain of T.

Finding Eigenvalues

- **Characteristic polynomial**: the polynomial of a matrix that is invariant under matrix similarity and has eigenvalues as its roots, i.e.,

$$p(\lambda) = \det(\boldsymbol{A} - \lambda I)$$

- **Characteristic equation**: when the character polynomial is equated to zero, i.e.,

$$\det(\boldsymbol{A} - \lambda I) = 0$$

- The characteristic equation is derived from the eigenvalue equation, i.e.,

$$\begin{aligned} \boldsymbol{A}\boldsymbol{v} &= \lambda \boldsymbol{v} \\ \boldsymbol{A}\boldsymbol{v} - \lambda \boldsymbol{v} &= \boldsymbol{o} \\ \underbrace{(\boldsymbol{A} - \lambda I)}_{\text{nonzero kernel}^{\uparrow}}\, \boldsymbol{v} &= \boldsymbol{o} \end{aligned}$$

  If $\boldsymbol{v}$ is the zeros vector, then it is a trivial solution, thus $(\boldsymbol{A} - \lambda I)$ must have a nonzero kernel and thus is not invertible[↑], meaning the determinant must be zero, yielding the characteristic equation.

- Since λ is unknown, then the determinant yields a polynomial, wherein the roots are in terms of λ and thus yields a means to find the eigenvalues.
- The characteristic equation is an Nth order polynomial equation in the unknown λ, meaning it will have N_λ distinct solutions where $1 \leq N_\lambda \leq N$, i.e., an $n \times n$ matrix will have n eigenvalues which may or may not be repeated.
- A shortcut for finding the eigenvalues for a 2×2 matrix involves simply taking the trace and the determinant of the original then solving the polynomial, i.e,

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

- Any triangular[↑] matrix (upper or lower) has eigenvalues that are simply the elements along their diagonal.

Finding Eigenvectors

- In most cases, the eigenvalues are only needed to determine the eigenvectors, which are generally the primary objects of interest.
- Each λ allows you to find the corresponding $\boldsymbol{v}$ by shifting the matrix by the λ and finding the nontrivial vector that is in the null space of the shifted matrix, i.e.,

$$\forall \lambda, \quad \boldsymbol{v}_i \in N(\boldsymbol{A} - \lambda_i I)$$

Diagonalization

- **Diagonalization**: the result of eigendecomposition on a square $n \times n$ matrix $\boldsymbol{A}$ with n linear independent[↑] eigenvectors that yields a factorized equation:

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$$

  where $\boldsymbol{V}$, whose ith column is the eigenvector $\boldsymbol{v}_i$ of $\boldsymbol{A}$, and a diagonal matrix $\boldsymbol{\Lambda}$, whose diagonal elements are the corresponding eigenvalues $\boldsymbol{\Lambda}_{ii} = \lambda_i$.

  - Diagonalization takes all the eigenvalue equations ($\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$) of a matrix and returns the "eigenmatrix equation," i.e.,

$$\boldsymbol{A}\boldsymbol{v}_1 = \lambda_1\boldsymbol{v}_1, \quad \boldsymbol{A}\boldsymbol{v}_2 = \lambda_2\boldsymbol{v}_2, \quad \cdots, \quad \boldsymbol{A}\boldsymbol{v}_n = \lambda_n\boldsymbol{v}_n$$

$$\downarrow$$

$$\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} = \begin{bmatrix} \lambda_1\boldsymbol{v}_1 & \lambda_2\boldsymbol{v}_2 & \cdots & \lambda_n\boldsymbol{v}_n \\ \lambda_1\boldsymbol{v}_1 & \lambda_2\boldsymbol{v}_2 & \cdots & \lambda_n\boldsymbol{v}_n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1\boldsymbol{v}_1 & \lambda_2\boldsymbol{v}_2 & \cdots & \lambda_n\boldsymbol{v}_n \end{bmatrix}$$

$$\downarrow$$

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda}$$

  - Thus, $\boldsymbol{V}$ must be invertible[↑], which means there must be n distinct eigenvalues.
  - Essentially, $\boldsymbol{V}$ transforms $\boldsymbol{A} \to \boldsymbol{\Lambda}$ and $\boldsymbol{V}^{-1}$ transforms $\boldsymbol{\Lambda} \to \boldsymbol{A}$.

- **Diagonalizable (non-defective) matrix**: when there exists an invertible matrix $\boldsymbol{V}$ and diagonal matrix $\boldsymbol{\Lambda}$ such that $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$.
  - Diagonalization is really just a name for the process of finding such matrices, but is often synonymous with eigendecomposition.
  - The eigenvectors are typically normalized, though magnituded is canceled by the inverse. However, when discussing the $\boldsymbol{V}$ in finite space, then $\boldsymbol{V}$ usually refers to an ordered basis[↑] of eigenvectors that describe the transformation.
- **Defective matrix**: a square matrix that is not diagonalizable, which means there exist no invertible matrix $\boldsymbol{V}$ and diagonal matrix $\boldsymbol{\Lambda}$ consisting of only real numbers.
  - Note: many defective matrices may still be diagonalizable using complex entries, e.g., a rotation matrix has no eigenvectors consisting of only real numbers.

Matrix Powers

- Multiplication of a matrix with itself can easily be done using diagonalizable matrices, as $\boldsymbol{V}^{-1}\boldsymbol{V} = I$, i.e.,

$$\boldsymbol{A}^n = \boldsymbol{V}\boldsymbol{\Lambda}_1(\boldsymbol{V}^{-1}\boldsymbol{V})\boldsymbol{\Lambda}_2(\boldsymbol{V}^{-1}\boldsymbol{V})\cdots(\boldsymbol{V}^{-1}\boldsymbol{V})\boldsymbol{\Lambda}_n\boldsymbol{V}^{-1} = \boldsymbol{V}\boldsymbol{\Lambda}^n\boldsymbol{V}^{-1}$$

Properties of Eigendecomposition

Eigenvectors of Distinct Eigenvalues

- Distinct $\lambda \to$ distinct $N(\boldsymbol{A} - \lambda I) \to$ distinct $\boldsymbol{v}$
- Proof by contradiction with the assumptions $\lambda_1 \neq \lambda_2$ and $\alpha_1 \boldsymbol{v}_1 = \alpha_2 \boldsymbol{v}_2$:

$$
\begin{aligned}
\alpha_1 \boldsymbol{v}_1 &= \alpha_2 \boldsymbol{v}_2 \\
\alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 &= \boldsymbol{o} \\
&\swarrow \quad \searrow \\
\boldsymbol{A}\,\alpha_1 \boldsymbol{v}_1 + \boldsymbol{A}\,\alpha_2 \boldsymbol{v}_2 &= \boldsymbol{o} \qquad \lambda_1\,\alpha_1 \boldsymbol{v}_1 + \lambda_1\,\alpha_2 \boldsymbol{v}_2 = \boldsymbol{o} \\
\lambda_1\,\alpha_1 \boldsymbol{v}_1 + \lambda_2\,\alpha_2 \boldsymbol{v}_2 &= \boldsymbol{o} \\
\lambda_1\,\alpha_1 \boldsymbol{v}_1 + \lambda_2\,\alpha_2 \boldsymbol{v}_2 &= \boldsymbol{o} \;-\; \lambda_1\,\alpha_1 \boldsymbol{v}_1 + \lambda_1 \alpha_2\, \boldsymbol{v}_2 = \boldsymbol{o} \\
&\downarrow \\
\lambda_2\,\alpha_2 \boldsymbol{v}_2 - \lambda_1\,\alpha_2 \boldsymbol{v}_2 &= \boldsymbol{o} \\
(\lambda_2 - \lambda_1)\alpha_2 \boldsymbol{v}_2 &= \boldsymbol{o} \\
\alpha_2 &= 0 \qquad \text{Ignoring trivial solution} \\
\alpha_1 \boldsymbol{v}_1 + 0\boldsymbol{v}_2 &= \boldsymbol{o} \\
\alpha_1 &= 0 \qquad \text{Ignoring trivial solution}
\end{aligned}
$$

- Thus, if the only scalar that makes a sequence of vectors the zero vectors zero, then by definition they are linearly independent[↑] and there are distinct eigenvectors for each eigenvalue.

Eigenvectors of Repeated Eigenvalues

- Repeated $\lambda \to$ same $N(\boldsymbol{A} - \lambda I) \to$ eigenspace of $\boldsymbol{v}_1 + \boldsymbol{v}_{\dots} + \boldsymbol{v}_n$
- **Geometric multiplicity** $\gamma_T(\lambda)$: the dimension of the eigenspace associated with λ, i.e., the maximum number of linearly independent eigenvectors associated with λ.
  - If $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are eigenvectors of a linear transformation T and share a repeated eigenvalue λ, then the eigenspace associated with λ is a linear subspace, i.e.,

$$T(\boldsymbol{v}_1 + \boldsymbol{v}_2) = \lambda(\boldsymbol{v}_1 + \boldsymbol{v}_2)$$

$$T(\alpha \boldsymbol{v}) = \lambda(\alpha \boldsymbol{v})$$

  - **Eigenline, eigenplane**: when the eigenspace has dimension 1 or 2, respectively.
  - When $\gamma_T(\lambda) \geq 1$ since every eigenvalue has at least one eigenvector
  - When $\gamma_T(\lambda) = \dim((\boldsymbol{A}))$, then there are distinct λ each with distinct $\boldsymbol{v}$ as described above.

Eigendecomposition of Symmetric Matrices

- Every $n \times n$ real symmetric matrix↑ has eigenvalues that are real and eigenvectors that are real and orthonormal.
- Since orthogonal matrices↑ transposed are equal to the inverse ($\boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$), then a real symmetric matrix $\boldsymbol{A}$ can be decomposed as

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$$

  - Note: $\boldsymbol{Q}$ is typically used to indicate orthogonal matrices, but here $\boldsymbol{Q}$ indicates an orthogonal matrix composed of the eigenvectors that are normalized↑.
  - Again, this also means that $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$, which can be very useful.
- This is useful property since the transpose is much easier and more accurate to compute than the inverse avoids the use of imaginary numbers that are sometimes needed in order to perform eigendecomposition.

Useful Facts Regarding Eigenvalues

- The product of the eigenvalues is equal to the determinant↑ of $\boldsymbol{A}$, i.e.,

$$\det(A) = \prod_{i=1}^{N_\lambda} \lambda_i^{n_i}$$

  - Thus, the amount of nonzero eigenvalues is equal to the rank of the matrix.
  - Also, any matrix with an eigenvalue equal to 0 means the determinant is also 0, indicating that the matrix is singular↑.
- The sum of the eigenvalues is equal to the trace↑ of $\boldsymbol{A}$, i.e.,

$$\operatorname{tr}(\boldsymbol{A}) = \sum_{i=1}^{N_\lambda} n_i \lambda_i$$

- If the eigenvalues of $\boldsymbol{A}$ are λ_i and $\boldsymbol{A}$ is invertible, then the eigenvalues of $\boldsymbol{A}^{-1}$ are simply λ_i^{-1}

Singular Value Decomposition

Singular Value Decomposition Fundamentals

- **Singular value decomposition (SVD)**: an extension of eigendecomposition[↑] to any $m \times n$ matrix via polar decomposition.
  - **Polar decomposition**: the factorization of a matrix $\boldsymbol{A}$ in the form $\boldsymbol{A} = \boldsymbol{UP}$, where $\boldsymbol{U}$ is a unitary matrix and $\boldsymbol{P}$ is a positive-semidefinite Hermitian matrix[↑], both square and the same size.
    - **Unitary**: when a complex square matrix has a conjugate transpose[↑] $\boldsymbol{A}^*$ that is also it's inverse, i.e.,
      $$\boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^* = \boldsymbol{I}$$
    - The real analogue of a unitary matrix is an orthogonal matrix[↑].
  - The rest of the details regarding polar decomposition will not be covered, as it detracts from understanding of single value decomposition; for now we will only deal with real matrices.
  - Other important notes before continuing:
    - Recall how creating symmetric matrices[↑] is done, i.e.,
      - $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{S}_{n\times n}$ and $\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{S}_{m\times m}$
    - And that the column and row space[↑] of matrices multiplied with their transpose are the same the original matrix column/row space, i.e.,
      $$C(\boldsymbol{A}\boldsymbol{A}^T) = C(\boldsymbol{S}) = C(\boldsymbol{A}), \quad R(\boldsymbol{A}^T\boldsymbol{A}) = R(\boldsymbol{S}) = R(\boldsymbol{A})$$
  - Finally, the singular value decomposition in more detail:
    $$\underbrace{\boldsymbol{A}}_{m\times n} = \underbrace{\boldsymbol{U}}_{m\times m}\underbrace{\boldsymbol{\Sigma}}_{m\times n}\underbrace{\boldsymbol{V}^T}_{n\times n}$$
    - $\boldsymbol{U}$ — the orthogonal basis for column space of $\boldsymbol{A}$.
      - Termed the **left singular vectors** as $\boldsymbol{U}^T\boldsymbol{A} = \boldsymbol{\Sigma}\boldsymbol{V}^T$
      - In the form of eigenvalue equation $\boldsymbol{u}^T\boldsymbol{A} = \sigma\boldsymbol{v}^T$.
    - $\boldsymbol{\Sigma}$ — a diagonal matrix consisting of the **singular values** σ of $\boldsymbol{A}$.
      - The number of non-zero singular values is equal to the rank[↑] of $\boldsymbol{A}$.
    - $\boldsymbol{V}^T$ — the orthogonal basis for row space of $\boldsymbol{A}$.
      - Termed the **right singular vectors** as $\boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}$
      - In the form of eigenvalue equation $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{u}\sigma$

Computing SVD

- Assuming A is an $m \times n$ matrix, then multiplying by the transpose yields two results that are the similar to eigendecomposition and dependent on the order of multiplication:

$$
\begin{aligned}
A &= U\Sigma V^T \\
A^TA &= (U\Sigma V^T)^T U\Sigma V^T \\
S &= V^{TT}\Sigma^T U^T U\Sigma V^T \\
S &= V\Sigma^T I\Sigma V^T \\
S &= V\Sigma^2 V^T \\
&\downarrow \\
S &= Q\Lambda Q^T \\
R(S) &= Q\Lambda Q^T \\
R(A) &= Q\Lambda Q^T \\
&\downarrow \\
R(A) &= V\Sigma^2 V^T
\end{aligned}
\qquad
\begin{aligned}
A &= U\Sigma V^T \\
AA^T &= U\Sigma V^T (U\Sigma V^T)^T \\
S &= U\Sigma V^T V^{TT}\Sigma^T U^T \\
S &= U\Sigma I\Sigma^T U^T \\
S &= U\Sigma^2 U^T \\
&\downarrow \\
S &= Q\Lambda Q^T \\
C(S) &= Q\Lambda Q^T \\
C(A) &= Q\Lambda Q^T \\
&\downarrow \\
C(A) &= U\Sigma^2 U^T
\end{aligned}
$$

- Thus, in the case of a real square matrix, $\Sigma^2 = \Lambda$ when A is transposed with itself and V^T / U act as the orthogonal basis for the row/column spaces, respectively.
- Reviewing the singular value equation:

$$
\begin{array}{cc}
A = U\Sigma V^T & \\
\swarrow \quad \searrow & \\
U^TA = \Sigma V^T & AV = U\Sigma \\
u^TA = \sigma v^T & Av = u\sigma
\end{array}
$$

- I.e., the left singular matrix U^T is a matrix whose rows describe a transformation of $A \rightarrow$ an orthogonal basis V^T wherein each column scaled by its respective singular value σ describes the row space of A.
- Likewise, the right singular matrix V is a matrix whose columns describe a transformation of $A \rightarrow$ an orthogonal basis U wherein each column scaled by its respective singular value σ describes the column space of A.

Singular Values vs. Eigenvalues

- $\text{eig}(A^TA) = \text{svd}(A)^2$
- $\text{eig}(A^TA) = \text{svd}(A^TA)$, as $\text{svd}(A^TA)$ essentially squares the singular values.
- $\text{eig}(A)$ and $\text{svd}(A)$ are usually not related at all, unless A consists all real entries.
  - Singular values are always real, whereas eigenvalues are likely to be complex.

Relation to Matrix Subspaces

- Since the SVD provides orthogonal basis for the row and column spaces of a matrix, then it also tells provides the kernel and cokernel[↑] of the matrix.
- Again, revisiting the singular value equation and taking note of dimensions of each of the matrices yields plenty of useful information:

$$\underbrace{\boldsymbol{A}}_{m\times n} = \underbrace{\boldsymbol{U}}_{m\times m}\underbrace{\Sigma}_{m\times n}\underbrace{\boldsymbol{V}^T}_{n\times n}$$

  - Again, the rank r is equal to the sum of singular values greater than 0, i.e.,

$$\text{rank}(\boldsymbol{A} = \Sigma\,(\sigma > 0))$$

  - Where the orthogonal basis $\boldsymbol{U}$ spanning all of R^m with:
    - a column space of $1:r$
    - and a cokernel of $r+1:m$
  - Likewise, the orthogonal basis $\boldsymbol{V}^T$ spanning all of R^n has:
    - a row space of $1:r$
    - and a kernel of $r+1:n$

Applications of SVD

Low-Rank Approximations

- **Low-rank approximation**: a minimization problem, in which the cost function measures the fit between a given matrix (the data) and the approximation matrix (optimization variable), subject to a constraint that the approximation matrix has reduced rank↑.
- The goal is to reduce data used in the system, while retaining as much information as necessary; the balance between the trade-off is often application dependent.
- This is done by taking a subset of matrices within the SVD, up to range k, which can be determined with percent variance↓.
  - The relative importance of singular values and corresponding basis vectors is normally sorted in the output of the algorithm, so typically values beyond k are noise and offer little data, leading to minimization opportunities.

$$\underbrace{A}_{m\times n} = \underbrace{U}_{m\times k}\,\underbrace{\Sigma}_{k\times k}\,\underbrace{V^T}_{k\times n}$$

Percent Variance

- Comparing or interpreting singular values at directly is often a challenge, and for many applications one might want to know the relative importance of each singular value, which is done through analysis of the variance of the singular values.
- Singular values are scale-dependent, i.e., any scaling (Φ) of the matrix $\boldsymbol{A}$ will scale the singular values by the same amount:

$$\Phi \boldsymbol{A} = \boldsymbol{U}(\Phi\Sigma)\boldsymbol{V}^T$$

- Any matrix can be formed by combining rank 1 matrices, and in context of SVD, those matrices are derived from the outer product↑ of columns of the $\boldsymbol{U}$ and corresponding row of $\boldsymbol{V}^T$ (provides direction) and scaled by the corresponding singular value (provides magnitude), i.e.,

$$\boldsymbol{A} = \boldsymbol{u}_1\,\sigma_1\,\boldsymbol{v}_1^T + \boldsymbol{u}_2\,\sigma_2\,\boldsymbol{v}_2^T + \cdots + \boldsymbol{u}_n\,\sigma_n\,\boldsymbol{v}_n^T$$

  - The outer product of $\boldsymbol{u}_n\boldsymbol{v}_n^T$ involves singular vectors from the orthogonal bases, meaning they have unit length $|1|$ and only encodes the direction.
  - The corresponding singular value encodes the importance, or magnitude, of the direction given by the vectors.
  - The sum of the singular values tells you the total importance (variance) of all the directions in the matrix.

- Thus, the proportion of each singular value of total variance can easily be found, yielding the percent variance:

$$\sigma_i = 100 \frac{\sigma_i}{\sum \sigma}$$

- This normalization allows for easy identification of the complexity of the matrix, and where data compression (i.e., range k of relevant values) might be beneficial.

Pseudoinverse

- **Pseudoinverse† (Moore-Penrose inverse)**: a widely known generalization of the inverse matrix, used for computing the inverse of singular matrices$^\uparrow$.
  - There are multiple generalizations of the inverse, or different definitions of the pseudoinverse, as well as ways to compute each. However, the Moore-Penrose is the most common, and SVD is generally used to compute the inverse.
- Finding the regular inverse of a matrix using SVD is useful to understand how the pseudoinverse is computed:

$$\begin{aligned} A &= U\Sigma V^T \\ A^{-1} &= (U\Sigma V^T)^{-1} \\ A^{-1} &= V^{-T}\Sigma^{-1}U^{-1} \\ A^{-1} &= V\Sigma^{-1}U^T \end{aligned}$$

- Σ^{-1} is found simply by taking the inverse of each element along the diagonal; this clearly shows why singular matrices cannot be used, as 0^{-1} is undefined.
- Thus, a pseudoinverse† is needed, i.e.,

$$\begin{aligned} A &= U\Sigma V^T \\ A^\dagger &= (U\Sigma V^T)^\dagger \\ A^\dagger &= V^{-T}\Sigma^\dagger U^{-1} \\ A^\dagger &= V\Sigma^\dagger U^T \end{aligned}$$

- The difference is that $\Sigma^\dagger$ only inverts the non-zero singular values.

Condition Number

- **Condition number** (κ): the measure of how much the output of a function changes based in the input, i.e., it measures how sensitive a function is to changes or errors in the input.

- In regard to SVD, the condition number defines the spread of the information in a matrix by comparing the ratio of the maximum singular value to the smallest, i.e.,

$$\frac{\sigma_{\max}}{\sigma_{\min}}$$

- The condition number for any singular matrix is undefined, as the minimum singular value would be zero. Additionally, the closer minimum value gets to zero, the less conditioned a matrix is (larger spread), i.e.,
  - Small min value → large spread → large condition number → **ill-conditioned**.
  - Large min value → small spread → small condition number → **well-conditioned**.
- There is no specified threshold between ill- and well-conditioned matrix—it is often dependent on the dataset.
- Having an ill-condition matrix isn't necessarily bad, instead, the condition number is often used as an indicator of large-scale structure within the matrix.

Quadratic Form

Quadratic Form Fundamentals

Quadratic Form in Algebra

-

Quadratic Form in Geometry

-

Properties of Quadratic Form

Normalized Quadratic Form

-

Eigenvectors and the Quadratic Form Surface

-

Definiteness

-