HCAI5TML01 – Mathematics of Learning.

Week – 2: Lecture – 01

A Refresher on the Mathematics Behind Machine Learning.

Fundamental Theory of Linear Algebra.

Siman Giri

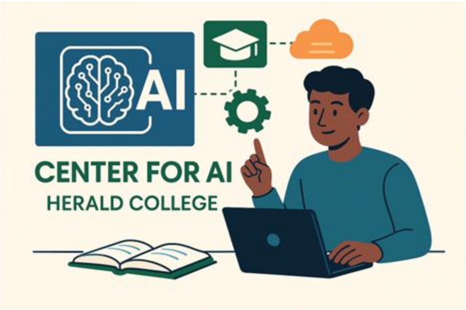
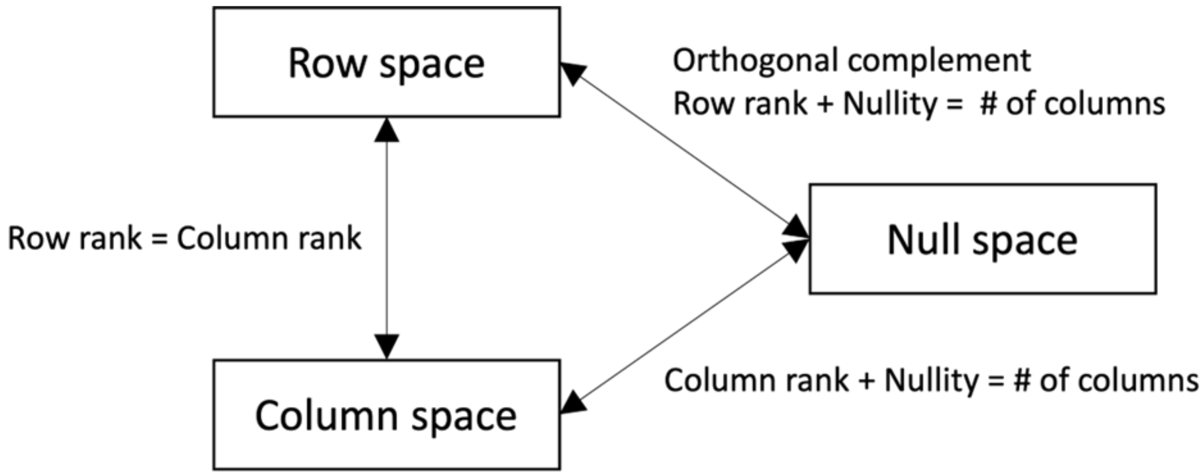


image generated via copilot.

1. Fundamental Theorem of Linear Algebra.
{FTLA}

1.1 What is FTLA?

- The **Fundamental Theorem of Linear Algebra (FTLA)** summarizes the relationships between the four fundamental subspaces associated with a matrix.

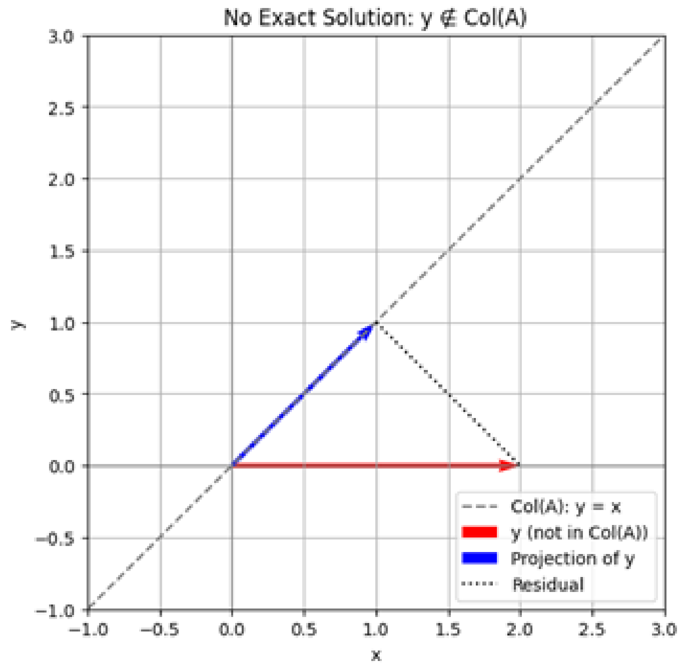


1.2 Fundamental Subspace and Machine Learning.

- We are solving the **linear regression problem**, **defined:**
  - $\mathbf{Ax} = \mathbf{y}$
  - where:
    - $\mathbf{A} \in \mathbb{R}^{\mathbf{m \times n}}$ is your design matrix (inputs/features),
    - $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ is the parameter vector (model weights),
    - $\mathbf{y} \in \mathbb{R}^{\mathbf{m}}$ is the target output.
- Evaluate Existence and Uniqueness of Solutions:
  - If **design matrix A** is full rank matrix.
  - If **design matrix A** is rank deficient matrix.
- Remember:
  - The Four Fundamental subspaces of A:
    - Column Space $\mathbf{C(A)} \subseteq \mathbb{R}^{\mathbf{m}}$**:**
      - All vectors y for which $\mathbf{Ax} = \mathbf{y}$ has at least one solution.
    - Null Space $\mathcal{N}(\mathbf{A}) \subseteq \mathbb{R}^{\mathbf{n}}$**:**
      - All vectors x such that $\mathbf{Ax} = \mathbf{0}$**:**
    - Row Space $\mathbf{C}(\mathbf{A}^{\mathbf{T}}) \subseteq \mathbb{R}^{\mathbf{n}}$**:**
      - The span of the rows of A.
    - Left Null Space $\mathcal{N}(\mathbf{A}^{\mathbf{T}}) \subseteq \mathbb{R}^{\mathbf{m}}$**:**
      - All vectors z such that $\mathbf{A}^{\mathbf{T}}\mathbf{z} = \mathbf{0}$.

1.2.1 Existence of the Solution.

- When does the linear system $\mathbf{Ax} = \mathbf{y}$ have at **least one solution** ?
  - **Fundamental Condition:**
    - $\mathbf{y} \in \mathbf{C(A)}$
    - y must lie within the column space of A for any solution to exist.
  - Then what does $\mathbf{y} \notin \mathbf{C(A)}$ mean?
      - y lies outside the subspace spanned by A.
    - In this case, the system is inconsistent there is **no exact solution**.
    - What does this mean?
      - We can still **approximate a solution** that is:
      - Find $\hat{\mathbf{x}}$ such that:
        - $\mathbf{Ax} \approx \mathbf{y}$
      - This is called the **least squares solution**:
        - $\hat{\mathbf{x}} = \mathbf{argmin_x} \|\mathbf{Ax} - \mathbf{y}\|^2$
      - It gives the closest point $\hat{\mathbf{y}} = \mathbf{A} \cdot \hat{\mathbf{x}} \in \mathbf{C(A)}$ **to y.**
    - Reminder:
      - This is not a solution to $\mathbf{Ax} = \mathbf{y}$, It is a solution to $\mathbf{min}_x \|Ax - y\|$.



1.2.2 Uniqueness of Solution.

- **When is the solution to $\mathbf{Ax} = \mathbf{y}$ unique (if it exists)?**
  - **Fundamental Condition:**
    - There exist a unique solution if **Null space has trivial solution** i.e.
      - $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$
    - Why?
      - Proof by Contradiction:
        - If $\mathbf{x_0}$ is one solution to $\mathbf{Ax} = \mathbf{y}$, there exist a vector $\mathbf{z} \neq \{\mathbf{0}\} \in \mathcal{N}(\mathbf{A})$:
          - $\mathbf{A}(\mathbf{x_0} + \mathbf{z}) = \mathbf{Ax_0} + \mathbf{Az} = \mathbf{y} + \mathbf{0} = \mathbf{y}$
          - so, any vector of the form $\mathbf{x_{new}} = \mathbf{x_0} + \mathbf{z}$ is also a valid solution!
      - Thus, you have infinitely many solutions.

1.2.3 If $\mathbf{A} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ is Full Rank.

1. **If A is square and full rank (invertible):**
   - There exists a unique solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$.
   - The system $\mathbf{Ax} = \mathbf{y}$ can be solved exactly.
     - No null space exists except the zero vector, no infinite solutions.
2. **If A is full column rank but rectangular (tall matrix, $\mathbf{m} > \mathbf{n}$):**
   - The columns of A are linearly independent.
   - The system might be **overdetermined i.e. more equations than unknowns**.
   - Usually, no exact solution x exists that satisfies $\mathbf{Ax} = \mathbf{y}$.
   - Instead, you can find a unique least square solution ***minimizing*** $\|Ax - y\|_2$ such that:
     - $\mathbf{x} = \left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{y}$
     - Because $\left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)$ **is invertible due to full column rank**, the **least squares solution is unique**.

1.2.3 If $\mathbf{A} \in \mathbb{R}^{\mathbf{m} \times \mathbf{n}}$ is Full Rank.

3. **If A is full row rank but wide matrix (wide matrix, m < n):**
   - Rows are linearly independent.
   - The system is undetermined i.e. more unknowns than equations.
   - Infinite solutions may exist because null space is non – trivial.
     - Solution **x satisfy** $\mathbf{Ax} = \mathbf{y}$, but there are **infinitely many**.
   - Often, one picks the **minimum norm solution** using the **pseudoinverse:**
     - $\mathbf{x} = \mathbf{A}^{\mathbf{T}}\left(\mathbf{A}\mathbf{A}^{\mathbf{T}}\right)^{-1}\mathbf{y}.$

1.2.3.1 Summary Table.

Matrix Shape	Full Rank Condition	Solution Type	Notes
Square $n \times n$	rank = n	Unique exact solution: $x = A^{-1}y$	A is invertible.
Tall $m \times n, m > n$	rank = n	Unique least square solution	Overdetermined system
wide $m \times n, m < n$	rank = m	Infinité solutions; minimum norm solution via pseudoinverse	Undetermined System

1.2.4 If $\mathbf{A} \in \mathbb{R}^{\mathbf{m \times n}}$ is Rank Deficient.

- **Rank deficient means:**
  - $\mathbf{rank(A) < min(m, n)}$
    - i.e. Some columns of A are linearly dependent (redundant or perfectly correlated features)
- Then:
  - $\mathcal{N}$**(A) contains non – zero vectors.**
    - So, if a solution exists (i.e. if $y \in C(A)$) ,
      - there are infinitely many solution (because you can add any vector from the null space.)
  - If $y \notin C(A)$, **then no solution exists**.
- **Then what do we do when A is rank deficient?**

1.2.5 What do we do when A is rank deficient ?

1. **Remove or Combine redundant features:**
   - Perform **feature selection** or **dimensionality reduction** (e.g., PCA) to remove dependencies between columns.
   - After reducing feature space to full rank, the unique solution can be found as usual.
2. **Regularization:**
   - To avoid instability and deal with rank, deficiency regularization methods are used:
   - For Example, Ridge Regression (L2 regularization):
     - $\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathbf{T}}\mathbf{A} + \boldsymbol{\lambda}\mathbf{I}\right)^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{y}, \boldsymbol{\lambda} > \mathbf{0}$
     - Adding $\boldsymbol{\lambda}\mathbf{I}$ makes $\mathbf{A}^{\mathbf{T}}\mathbf{A} + \boldsymbol{\lambda}\mathbf{I}$ invertible even if $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ is singular.
   - This shrinks coefficients and selects a unique, stable solution.

1.2.5.1 What do we do when A is rank deficient ?

3. **Find a solution despite infinite possibilities:**
    - Since infinite solutions exist (due to null space vectors):
    - **Pick the minimum norm solution**,
        - which is the **solution x** with the smallest **Euclidean norm $\|x\|_2$.**
    - This solution can be found using the ***Moore-Penrose pseudoinverse A^+***:
        - $\hat{\mathbf{x}} = \mathbf{A}^+\mathbf{y}$.

1.3 What is the Moore-Penrose Pseudoinverse?

- The Moore-Penrose pseudoinverse of a matrix $\mathbf{A} \in \mathbb{R}^{\mathbf{m \times n}}$ is a matrix:
    - $\mathbf{A}^{+} \in \mathbb{R}^{\mathbf{n \times m}}$ that generalizes the inverse of A and satisfies following **four specific conditions**.
- It provides unique, best possible solution to linear system, even when A is not square or not full rank.
- **Moore – Penrose Conditions:**
    - **Reflexivity** $\mathbf{AA^{+}A = A}$
        - It means that:
            - If we apply the pseudoinverse $\mathbf{A^{+}}$ **to A**
                - And then apply A again,
                    - You return to the original matrix, represents the generalized inverse because:
                    - It acts like a true inverse **when A is invertible:** $\mathbf{A^{+} = A}$**.**
                    - It gives best approximate solutions **when A is not invertible**.
    - **Reflexivity dual:** $\mathbf{A^{+}AA^{+} = A^{+}}$
    - **Symmetry:** $\mathbf{(AA^{+})^{T} = AA^{+}}$
    - **Symmetry dual:** $\mathbf{(A^{+}A)^{T} = A^{+}A}$

1.3.1 Computing Pseudoinverse $\mathbf{A}^{+}$.

Case	$\mathbf{A}^{+}$
A is invertible	$\mathbf{A}^{+} = \mathbf{A}^{-1}$
A full column rank	$\mathbf{A}^{+} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}$
A full row rank	$\mathbf{A}^{+} = \mathbf{A}^{\mathrm{T}}\left(\mathbf{A}\mathbf{A}^{\mathrm{T}}\right)^{-1}$

- Directly computing $\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}$ can be numerically *unstable or impossible if $A^T A$* is **singular or ill-conditioned**.
- This is one of the reason why we need to understand **Techniques of Matrix Decomposition.**

2. Getting Started with Matrix Decompositions.

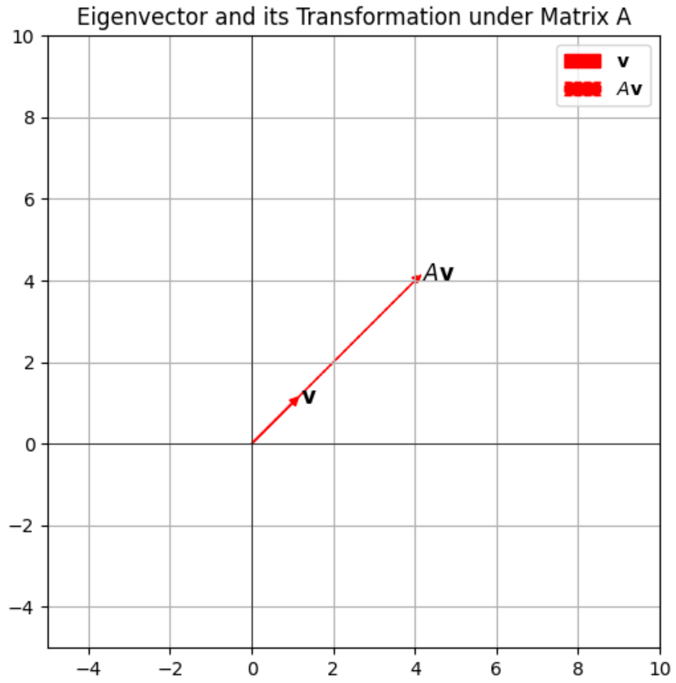
{Eigen Value Problem aka eigen – value Decompositions.}

2.1 Eigen Vector and Eigen Value.

- An **eigenvector** of a **square matrix A** is a **non-zero vector v** that, when **multiplied by A,** results in a scalar multiple of itself.
  - In other words, it is a vector that does not change direction when the linear transformation represented by A is applied to it.
    - Instead, it only gets scaled by a certain factor, called the **eigenvalue**.
- Mathematically, for any **Matrix – vector** pair if following holds:
  - $Av = \lambda v$
    - then the **vector v** is called eigen vector and the **scaling factor λ** is called eigen value.
- **Key points about eigen vectors:**
  - **Non-zero**: Eigenvectors are always non-zero vectors, $\mathbf{i.e.\, v \neq 0}$.
  - **Scaling:** The transformation A simply scales the eigenvector by the **eigenvalue λ**;
    - it does not change the **vector's direction**.
  - **Multiple eigenvectors**: For each eigenvalue, there can be infinitely many eigenvectors, all scalar multiples of each other. They form a subspace (called the eigenspace) corresponding to that eigenvalue.

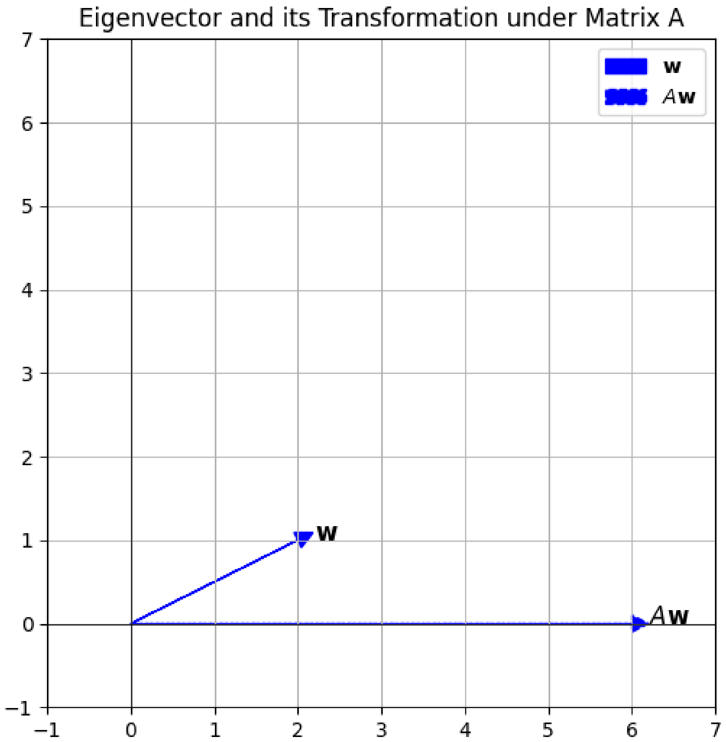
2.1.1 Identify the Eigen Vector.

- **Consider a matrix**
  - $\mathbf{A} = \begin{bmatrix} 2 & 2 \\ -4 & 8 \end{bmatrix}$ and
    - vectors $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$; $\mathbf{w} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$.
- **Which are Eigen vectors?**
  - For $\mathbf{v} \rightarrow$ **we check if v is an eigenvector by calculating Av**:
    - $\mathbf{Av} = \begin{bmatrix} 2 & 2 \\ -4 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow 4\mathbf{v}.$
    - So, $\mathbf{v}$ is an **eigen vector** with **eigenvalue** $\lambda = 4$.
- What about w?



2.1.1 Identify the Eigen Vector.

- **Consider a matrix**
  - $\mathbf{A} = \begin{bmatrix} 2 & 2 \\ -4 & 8 \end{bmatrix}$ and
    - vectors $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$; $\mathbf{w} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$.
- **Which are Eigen vectors?**
  - For w → **we check if w is an eigenvector by calculating Aw**:
    - $\mathbf{Aw} = \begin{bmatrix} 2 & 2 \\ -4 & 8 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix} \neq \lambda w.$
    - So, **w** is not an **eigen vector there does not exist a**
      - **scalar λ** under which $\mathbf{Aw} = \lambda\mathbf{w}$ holds true.



2.2 Eigen Value Problem.

- The **eigenvalue problem** is a fundamental concept in linear algebra and plays a critical role in various fields such as machine learning, physics, and computer science.
- It involves **finding scalar values** (called **eigenvalues**) and corresponding **non-zero vectors** (called **eigenvectors**) for a given **square matrix**.
  - Mathematically, Given a **square matrix A**, the eigenvalue problem is to find **scalars λ** and **eigen vector v** that satisfy the following equation.
    - $\mathbf{Av} = \lambda\mathbf{v}$.

2.3 Steps to solve the Eigenvalue Problem.

- Write the characteristic equation:
  - To find the eigenvalues, we rewrite the equation as:
    - $(\mathbf{A}-\boldsymbol{\lambda}\mathbf{I})\mathbf{v} = \mathbf{0}$ {called characterstic equation}
    - Where:
      - **I** is the **identity matrix** of the **same size as A**,
      - $\boldsymbol{\lambda} \rightarrow$ **eigen values**.
      - $\mathbf{v} \rightarrow$ **eigen vector**.
      - *Cautions: the matrix $A - \lambda I$ must be singular i.e. $det(A - \lambda I) = 0$.*
- Compute the characteristic polynomial:
  - Solve
    - $\mathbf{det(A-\lambda I) = 0}$,
  - which gives a **polynomial equation in $\boldsymbol{\lambda}$** which is called characteristic polynomial.
- Solve the characteristic polynomial:
  - Solve the polynomial equation to find the eigenvalues $\boldsymbol{\lambda_1, \lambda_2, \ldots \lambda_n}$.
- Find the eigen vectors:
  - For each **eigen value $\boldsymbol{\lambda_i}$**,
    - substitute it back into the equation $(\mathbf{A}-\boldsymbol{\lambda}\mathbf{I})\mathbf{v} = \mathbf{0}$ and solve for the **eigenvector $\boldsymbol{v}$.**

2.3.1 Example Problem.

Eigenvalues of Matrix A

Consider a matrix A:

$$A = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}$$

The characteristic equation is:

$$\det(A - \lambda I) = \det \begin{pmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{pmatrix} = 0$$

Expanding the determinant:

$$(4-\lambda)(3-\lambda) - 2 \times 1 = 0$$

Simplifying:

$$\lambda^2 - 7\lambda + 10 = 0$$

Solving this quadratic equation:

$$\lambda_1 = 5, \quad \lambda_2 = 2$$

2.3.1 Example Problem.

Eigenvectors of Matrix A

Next, we find the eigenvectors:
For $\lambda_1 = 5$, solve $(A - 5I)v = 0$:

$$(A - 5I) = \begin{pmatrix} -1 & 2 \\ 1 & -2 \end{pmatrix}$$

Solving the system:

$$\begin{pmatrix} -1 & 2 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0$$

This gives the eigenvector:

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

For $\lambda_2 = 2$, solve $(A - 2I)v = 0$:

$$(A - 2I) = \begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix}$$

Solving the system:

$$\begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0$$

This gives the eigenvector:

$$v_2 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

Conclusion: For the matrix A, the eigenvalues are $\lambda_1 = 5$ and $\lambda_2 = 2$, with corresponding eigenvectors $v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $v_2 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$.

2.4 Eigenvalue Decomposition.

- **Eigenvalue Decomposition** is a process where a **square matrix is factorized** into
  - its eigenvalues and eigenvectors.
  - Specifically, for a matrix A, if it can be decomposed into a product of three matrices:
    - $\mathbf{A} = \mathbf{V\Lambda V^{-1}}$
    - where:
      - $\mathbf{A}$ is the original matrix.
      - $\mathbf{V}$ is the matrix whose columns are the eigenvectors of A.
      - $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal entries are the eigenvalues of A.
      - $\mathbf{V^{-1}}$ is the inverse of the matrix V.
- One of the application of Eigenvalue decomposition is **Principal Component Analysis** used for dimensionality reduction purposes.

3. Getting Started with Matrix Decompositions.

{Singular Value Decompositions.}

3.1 What is SVD?

- Defintion:
  - For any real matrix $A \in \mathbb{R}^{m \times n}$ there exist orthogonal matrices U, V and a diagonal matrix Σ such that:
    - $\mathbf{A = U\Sigma V^T}$
- where:
  - $\mathbf{U} \in \mathbb{R}^{\mathbf{m \times m}}$: columns are left singular vectors.
    - These vectors form an **orthonormal basis** for $\mathbb{R}^m$(column space).
    - Orthogonal columns $\mathbf{U^T U = I}$
    - They **capture the directions** in which A sends the unit basis vectors after scaling.
  - $\mathbf{V} \in \mathbb{R}^{\boldsymbol{n \times n}}$: columns are **right singular vectors.**
    - These vectors form an **orthonormal basis** for $\mathbb{R}^n$(row space).
    - Orthogonal columns $\mathbf{V}^{\boldsymbol{T}}\mathbf{V} = \mathbf{I}$
    - They are eigenvectors of $A^T A$ and describe the directions of **principal axes** in the domain.
  - $\boldsymbol{\Sigma} \in \mathbb{R}^{\boldsymbol{m \times n}}$: diagonal matrix of **singular values i.e.** $\boldsymbol{\sigma_1 \geq \sigma_2 \geq \cdots \geq 0}$
    - Σ is a **diagonal matrix**, meaning all nonzero entries are on the diagonal.
    - The number of **non-zero singular values** = **rank of A.**

Orthogonal Matrices:

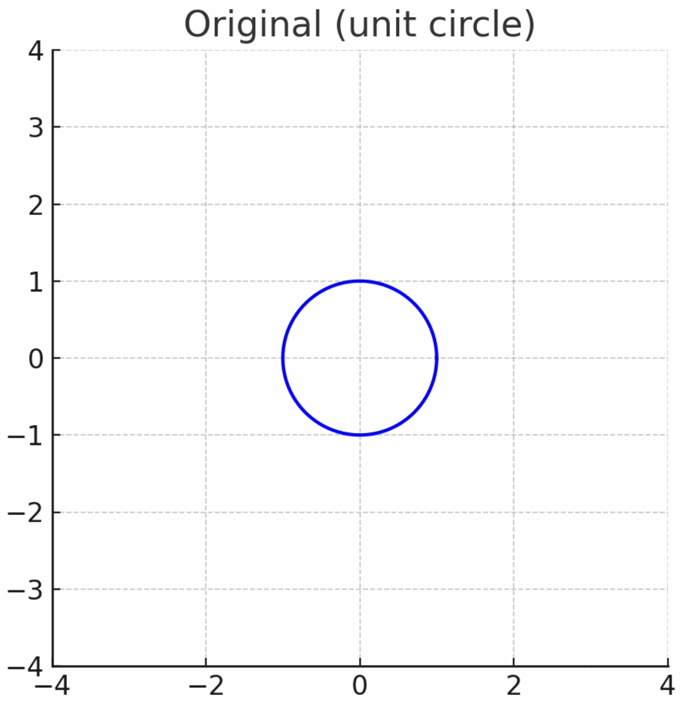
- An **orthogonal matrix** is a real square matrix $\mathbf{Q} \in \mathbb{R}^{\mathbf{n \times n}}$ whose **columns and rows** are **orthonormal vectors**, meaning:
  - **Columns are orthonormal:**
    - $\mathbf{Q^T Q = QQ^T = I}$
    - (where I is the identity matrix).
  - **Rows are also orthonormal:**
    - $\mathbf{QQ^T = I}$
- This implies that:
  - The **inverse of Q** is its transpose: $\mathbf{Q^{-1} = Q^T}$
  - The determinant of Q is either $\mathbf{+1}$ **or** $\mathbf{-1}$**:**
    - $\mathbf{det}(\boldsymbol{Q}) = \pm\mathbf{1}.$

Orthogonal Matrices: Key Properties.

1. **Preserves Lengths (Norm):**
   - For any vector **x:**
     - $\|\mathbf{Qx}\|_2 = \|\mathbf{x}\|_2$
   - This makes orthogonal matrices useful in rotations and reflections.
2. **Preserves Angles & Dot Products:**
   - For any vectors **x,y:**
     - $(\mathbf{Q} \cdot \mathbf{x})^{\mathbf{T}}(\mathbf{Q} \cdot \mathbf{y}) = \mathbf{x}^{\mathbf{T}}\mathbf{y}$
3. **All Eigenvalues Have Magnitude 1:**
   - If λ is an eigen values of Q, then $\|\boldsymbol{\lambda}\| = \mathbf{1}$
4. **Product of Orthogonal Matrices is Orthogonal:**
   - **If** Q_1 and Q_2 are **orthogonal**, then $\mathbf{Q_1} \cdot \mathbf{Q_2}$ are also **orthogonal.**

3.2 What does this decomposition says:

- **A** maps an input vector as:
  - First **rotate** it via $\mathbf{V}^T$ i.e. change coordinates into principal directions.
  - Then **scale** it via $\mathbf{\Sigma}$ (stretch or shrink).
  - Then **rotate again** via U (to map into output space).



3.3 Computing SVD.

- **Theoretical Methods using Eigen Decomposition:**

Component	Computation
V(right singular vectors)	Eigenvectors of A^TA
σ_i(singular values)	$\sqrt{\text{eigenvalues of } A^TA}$
U (left singular values)	$u_i = \frac{1}{\sigma_i}Av_i$

- **Practical Approach:**
  - Most real-world SVDs are computed using **numerical algorithms** (like Golub–Kahan SVD algorithm), typically via libraries like:
    - U, s, VT = np.linalg.svd(A)

Thank You.