CIS3333: Mathematics of Machine Learning Fall 2025

Lecture: Linear Algebra Review I

Date: October 1st, 2025 *Author: Surbhi Goel*

1 Why Linear Algebra for Machine Learning?

Recall from Lecture 1 that machine learning has three core components: data, models, and learning. Linear algebra is the mathematical foundation that underlies all three.

- **Data lives in vector spaces.** We represent data as vectors and matrices: images become vectors in $\mathbb{R}^D$, collections of N examples become matrices $X \in \mathbb{R}^{N \times D}$. Understanding vector spaces lets us reason about the structure and properties of our data.
- **Models are linear transformations.** Even complex models like neural networks are built from compositions of linear transformations (matrix multiplications) followed by simple non-linearities. Linear algebra gives us the tools to understand what these transformations can represent.
- **Learning involves optimization in high dimensions.** Finding $\hat{\theta} = \arg\min_\theta \hat{R}(\theta)$ means navigating through high-dimensional parameter spaces. Linear algebra provides the geometric intuition and computational tools (projections, decompositions) needed to solve these optimization problems efficiently.

This module builds toward answering the *representation* question from Lecture 1: what functions can we represent? We will develop tools that eventually let us work in infinite-dimensional spaces (Hilbert spaces, kernels) where linear models become arbitrarily expressive. The Representer Theorem shows that even in these infinite-dimensional spaces, the optimal solution has a finite, tractable form—a result that fundamentally relies on the linear algebra concepts we will cover.

Over the next few lectures, we will build up from basic matrix operations to abstract vector spaces, inner products, and decompositions. Our goal will be to learn about the the fundamental properties of linear systems, and generalize these properties to abstract vector spaces (function spaces like Hilbert spaces) that are not necessarily in the field of real numbers.

2 Linear Algebra Basics

Most likely you are familiar with basic operations on matrices and vectors. For example, $A \in \mathbb{R}^{3\times 5}$ is a matrix of real numbers with 3 rows and 5 columns, while $b \in \mathbb{R}^3$ is a vector of 3 elements. These form the basis of a system that we call linear algebra, which has several main properties.

- For $m, n \in \mathbb{N}$, a matrix A is a m, n tuple of elements a_{ij} where i denotes the row and j denotes the column.

- Addition: if $C = A + B$ and $A, B \in \mathbb{R}^{m \times n}$ then $c_{ij} = a_{ij} + b_{ij}$
- Product: If $C = AB$ and $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times n}$ then $c_{ij} = \sum_{l=1}^{k} a_{il} b_{lj}$ where $C \in \mathbb{R}^{m \times n}$
- Identity: $I_m \in \mathbb{R}^{m \times m}$ is an identity matrix when it is zero everywhere except the diagonal, i.e. $I_{ij} = \mathbf{1}[i = j]$
- Associativity: $(AB)C = A(BC)$
- Distributivity: $(A + B)C = AC + BC$, $A(C + D) = AC + AD$
- Multiplication with identity: $\forall A \in \mathbb{R}^{m \times n} : I_m A = A I_n = A$
- Inverse: Let $A \in \mathbb{R}^{n \times n}$. If $AB = I$ then $B = A^{-1}$ is the inverse of A
- Transpose: Let $A \in \mathbb{R}^{m \times n}$. The matrix $B = A^\top$ such that $b_{ij} = a_{ji}$ is called the transpose.
- Symmetric: $A \in \mathbb{R}^{n \times n}$ is symmetric if $A = A^\top$

We can also add scalars to the mix (single elements).

- Scalar multiplication: Let $\lambda \in \mathbb{R}$. Then, $\lambda A = K$ where $K_{ij} = \lambda a_{ij}$.
- Associativity: $(\lambda\phi)C = \lambda(\phi C)$. Actually, scalars can be moved around: $\lambda(BC) = (\lambda B)C = B(\lambda C) = (BC)\lambda$. Also, transpose doesn't affect matrices: $(\lambda C)^\top = C^\top \lambda = \lambda C^\top$
- Distributivity: $(\lambda + \phi)C = \lambda C + \phi C$ and $\lambda(B + C) = \lambda B + \lambda C$

One of the most common uses of matrices and vectors is to represent linear systems of equations in a compact form. I.e.

$$Ax = b$$

represents a series of linear equations, where each row of A is the coefficients for each variable x and the target scalar is the corresponding row in b.

Example: Linear Models. Recall from Lecture 1 that a linear model is $f_\theta(x) = \theta^T x$. When we have N training examples $(x_1, y_1), \ldots, (x_N, y_N)$ where $x_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$, we can compute all predictions at once by stacking the data into a matrix $X \in \mathbb{R}^{N \times D}$:

$$\begin{bmatrix} f_\theta(x_1) \\ \vdots \\ f_\theta(x_N) \end{bmatrix} = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \theta = X\theta$$

To minimize the error between our predictions and the labels, we could try to find parameters θ such that our predictions exactly match the labels, i.e., solve:

$$X\theta = Y \quad \text{where} \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

Does such a θ always exist? If it exists, is it unique? How do we compute it efficiently? We will answer these questions in the this module.

3 Groups

The space of matrices and vectors behaves *nicely*, in that it has these properties of associativity, distributivity, an identity and an inverse. Let's now generalize this structure.

- Groups: Let $\mathcal{G}$ be a set and an operation $\otimes : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ be defined on $\mathcal{G}$. Then $G = (\mathcal{G}, \otimes)$ is called a group if
  1. Closure: $\forall x, y \in \mathcal{G} : x \otimes y \in \mathcal{G}$
  2. Associativity: $\forall x, y, z \in \mathcal{G} : (x \otimes y) \otimes z = x \otimes (y \otimes z)$
  3. Neutral element: $\exists e \in \mathcal{G}\ \forall x \in \mathcal{G} : x \otimes e = e \otimes x = x$
  4. Inverse element: $\forall x \in \mathcal{G}\ \exists y \in \mathcal{G} : x \otimes y = y \otimes x = e$. We write x^{-1} to denote the inverse element of x. This does not always mean $\frac{1}{x}$ and is with respect to the operator $\otimes$.
  5. (Commutivity) If $\forall x, y \in \mathcal{G} : x \otimes y = y \otimes x$ then $\mathcal{G}$ is an Abelian group
- Examples of Abelian groups: $(\mathcal{Z}, +), (\mathcal{R} \setminus \{0\})$
- Examples of not-groups: $(\mathcal{N} + 0, +), (\mathcal{Z}, \cdot), (\mathcal{R}, \cdot)$
- $(\mathcal{R}^n, +), (\mathcal{Z}^n, +)$ are Abelian if using component wise addition
- Matrices and addition: $(\mathcal{R}^{m \times n}, +)$ is Abelian with component-wise addition
- Matrices and multiplication: $(\mathcal{R}^{m \times n}, \cdot)$ is only a group if the inverse always exists
- General Linear Group: set of invertible matrices $A \in \mathcal{R}^{n \times n}$ is a group with respect to matrix multiplication, but is not Abelian (not commutative)

4 Vector Spaces

Groups have an operation with structure that stays within the group. This can be referred to as an *inner* operation (i.e. elementwise addition) as the operator stays within the group. We can also consider an *outer* operation which takes in an element outside of the group.

- A real-valued vector space $V = (\mathcal{V}, +, \cdot)$ is a set $\mathcal{V}$ with two operations:
  - $+ : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ (inner operation: vector addition)
  - $\cdot : \mathbb{R} \times \mathcal{V} \to \mathcal{V}$ (outer operation: scalar multiplication)

  where
  1. $(\mathcal{V}, +)$ is an Abelian group
  2. Distributivity:
     - $\forall \lambda \in \mathbb{R}, \forall x, y \in \mathcal{V} : \lambda \cdot (x + y) = \lambda \cdot x + \lambda \cdot y$
     - $\forall \lambda, \psi \in \mathbb{R}, x \in \mathcal{V} : (\lambda + \psi) \cdot x = \lambda \cdot x + \psi \cdot x$
  3. Associativity (outer operation): $\forall \lambda, \psi \in \mathbb{R}, x \in \mathcal{V} : \lambda \cdot (\psi \cdot x) = (\lambda\psi) \cdot x$

4. Neutral element with respect to the outer operation: $\forall x \in \mathcal{V} : 1 \cdot x = x$

Note: Closure for the outer operation $\forall \lambda \in \mathbb{R}, \forall x \in \mathcal{V} : \lambda \cdot x \in \mathcal{V}$ is implicit in the type signature $\cdot : \mathbb{R} \times \mathcal{V} \to \mathcal{V}$.

- The elements $x \in V$ are called vectors. The neutral element of $(\mathcal{V}, +)$ is the zero vector 0.
- **Example: $\mathbb{R}^n$.** $\mathcal{V} = \mathbb{R}^n$, $n \in \mathbb{N}$ is a vector space with:
  - Addition: $x + y = (x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$ for all $x, y \in \mathbb{R}^n$
  - Multiplication by scalars: $\lambda x = \lambda(x_1, \ldots, x_n) = (\lambda x_1, \ldots, \lambda x_n)$ for all $\lambda \in \mathbb{R}, x \in \mathbb{R}^n$

  This is the most common vector space in machine learning—our data matrices $X \in \mathbb{R}^{N \times D}$ are collections of vectors from $\mathbb{R}^D$.
- **Example: Polynomials.** Let $\mathcal{P}_n = \{p(x) = a_0 + a_1 x + \cdots + a_n x^n : a_i \in \mathbb{R}\}$ be the set of polynomials of degree at most n. This is a vector space with:
  - Vector addition: $(p + q)(x) = p(x) + q(x)$
  - Scalar multiplication: $(\lambda p)(x) = \lambda \cdot p(x)$
  - Neutral element: the zero polynomial $0(x) = 0$
  - Inverse: for any polynomial p, its inverse is $-p$ where $(-p)(x) = -p(x)$

  For example, if $p(x) = 2 + 3x$ and $q(x) = 1 - x$, then $(p+q)(x) = 3 + 2x$ and $(2p)(x) = 4 + 6x$. The key point: this is a vector space of *functions*, not tuples of numbers.
- **Example: Positive reals with non-standard operations.** Let $\mathcal{V} = \mathbb{R}_{>0} = \{x \in \mathbb{R} : x > 0\}$ (positive real numbers) with operations:
  - Vector addition: $x \oplus y = xy$ (usual multiplication)
  - Scalar multiplication: $\lambda \odot x = x^\lambda$ (exponentiation)

  This turns out to be a vector space, check yourself!

4.1 Vector Subspaces

- A subspace of a vector space is a vector space: if $\mathcal{U} \subset \mathcal{V}$ and $V = (\mathcal{V}, +, \cdot)$ is a vector space, then if $U = (\mathcal{U}, +, \cdot)$ is a vector space we call it a subspace of V restricted to $\mathcal{U}$.
- Since U uses the same operations as V (restricted to $\mathcal{U} \times \mathcal{U}$ and $\mathbb{R} \times \mathcal{U}$), properties like associativity, commutativity, and distributivity automatically hold for elements in U.
- However, we must explicitly verify that U contains the neutral element and is closed under the operations. To show that U is a subspace, we need to show that
  - $0 \in \mathcal{U}$
  - $\forall \lambda \in \mathbb{R}, \forall x \in \mathcal{U} : \lambda x \in \mathcal{U}$
  - $\forall x, y \in \mathcal{U} : x + y \in \mathcal{U}$

- **Example: Solutions to homogeneous systems.** The solution set of $Ax = 0$ (where $A \in \mathbb{R}^{m \times n}$) forms a vector subspace of $\mathbb{R}^n$. To see this:
  - Contains 0: $A \cdot 0 = 0$
  - Closed under addition: If $Ax_1 = 0$ and $Ax_2 = 0$, then $A(x_1 + x_2) = Ax_1 + Ax_2 = 0 + 0 = 0$
  - Closed under scalar multiplication: If $Ax = 0$, then $A(\lambda x) = \lambda(Ax) = \lambda \cdot 0 = 0$

  This subspace is called the *null space* or *kernel* of A.

- **Counterexample:** The solution set of $X\theta = Y$ (with $Y \neq 0$) from our earlier example is *not* a subspace because it doesn't contain 0: if $Y \neq 0$, then $X \cdot 0 = 0 \neq Y$. Also see Example 2.12 from the textbook.

- **Example: Polynomial subspaces.** Consider $\mathcal{P}_2 = \{a_0 + a_1 x + a_2 x^2 : a_i \in \mathbb{R}\}$, the space of polynomials of degree at most 2. This is a subspace of $\mathcal{P}_3$:
  - Contains 0: the zero polynomial $0(x) = 0$ is in $\mathcal{P}_2$
  - Closed under addition: sum of two degree-2 polynomials has degree at most 2
  - Closed under scalar multiplication: scaling a degree-2 polynomial gives a degree-2 polynomial

  More generally, $\mathcal{P}_m$ is a subspace of $\mathcal{P}_n$ whenever $m \leq n$.

Why abstract vector spaces? While our data lives in $\mathbb{R}^D$, the abstract framework we've developed applies to much more general spaces. The polynomial example shows that vector spaces can be sets of *functions*, not tuples of numbers. Later in this module, we'll extend this further to infinite-dimensional function spaces (Hilbert spaces, kernels), using the same linear algebra tools. This generalization is key to the Representer Theorem and kernel methods, which allow us to learn arbitrarily complex functions while maintaining computational tractability.