# EEN1083
## Data Analysis and Machine Learning I

Ali Intizar

Semester 1
2024/2025

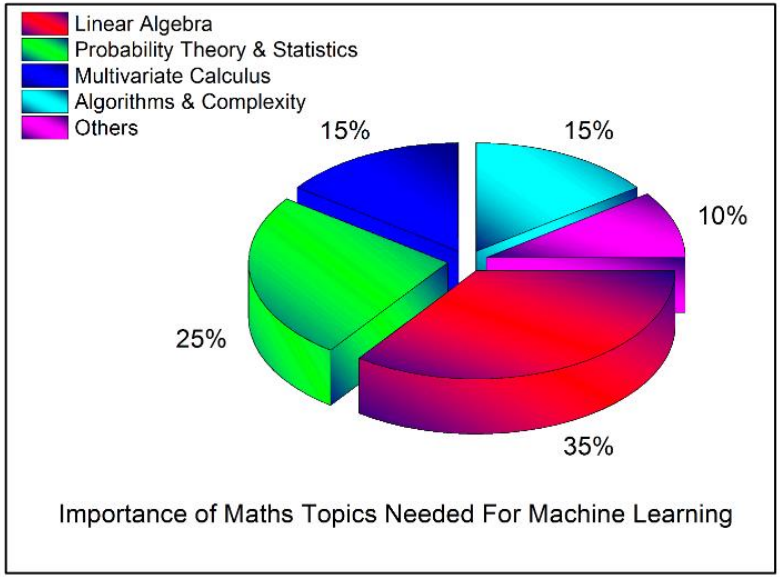
DCU
Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

# Mathematics for Machine Learning

DCU Ollscoil Chathair Bhaile Átha Cliath Dublin City University

# Why we need mathematics for machine learning?

- Mathematics is one of the core foundational skills
- Algorithms Selection
- Parameters Selection
- Fine Tuning Trained Algorithms
- Optimisation
- Training and Testing Algorithms



# What are core concepts I need to be aware of?

- Linear Algebra
  - Vectors and Matrixes
  - Transpose and Inverse
  - Dot products
  - Eigenvalues
  - Euclidean distance
- Calculus
  - Multi-variables
  - Limit, continuity, derivatives
  - Differential and Integral
- Statistics and Probability
  - Sampling
  - Regression
  - Conditional probability

# Sets

A set is a well-defined collection of **distinct** objects (possibly infinite or uncountable).

Some examples:

- $\{1, 2, 3\}$, $\{a, e, i, o, u\}$, $\{\pi, e\}$
- Integers $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$
- Positive integers $\mathbb{Z}_{++} = \{1, 2, 3, \ldots\}$
- Real numbers $\mathbb{R}$

If $x$ is an element of set $Z$ we write $x \in Z$. E.g. $x \in \mathbb{R}$ means $x$ is a real number.

Set builder notation:

- Positive reals $\mathbb{R}_{++} = \{x \in \mathbb{R} : x > 0\}$

# Sets: empty set, cardinality, intersection, union

The empty set is the set with no elements $\emptyset = \{\}$.

The cardinality of a set is the number of elements in the set. E.g. $X = \{1, 2, 3\}$, $|X| = \#X = 3$.

The intersection of two sets is the set containing all common elements. If $A = \{1, 2, 3\}$ and $B = \{3, 4, 5\}$, then the intersection $A \cap B = \{3\}$.

- $\mathbb{Z} \cap \mathbb{R} = \mathbb{Z}$

The union of two sets is the set containing all elements that occur in either set. If $A = \{1, 2, 3\}$ and $B = \{3, 4, 5\}$, then the union $A \cup B = \{1, 2, 3, 4, 5\}$.

- $\mathbb{Z} \cup \mathbb{R} = \mathbb{R}$

# Sets: subsets

$A$ is a subset of $B$ if all the elements of $A$ are also contained in $B$. Written as $A \subset B$.

- $\{1, 2\} \subset \{1, 2, 3\}$
- $\mathbb{Z} \subset \mathbb{R}$
- $\mathbb{Z}_{++} \subset \mathbb{Z}_{+} \subset \mathbb{Z} \subset \mathbb{R}$
- $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$

# Vectors

Vectors. In general, an abstract mathematical notation, but for purpose of this module can be thought of as an ordered list of numbers.

E.g. **column vectors**:

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 3 \\ 1 \\ 8 \end{bmatrix}$$

To say that a vector $\mathbf{x}$ is real valued with $D$ dimensions, we write $\mathbf{x} \in \mathbb{R}^D$. E.g. $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{y} \in \mathbb{R}^3$.

Can write column vectors more compactly using parentheses $\mathbf{x} = (1\,3)$

The elements of a vector are usually denoted using subscripts. E.g. if $\mathbf{x} = (1\,4\,5)$ then $x_1 = 1, x_2 = 4, x_3 = 5$.

The transpose of a column vector is a **row vector**.

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \mathbf{x}^T = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

This gives us another way to write column vectors compactly:

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T \in \mathbb{R}^3$$

# Adding and scaling vectors

To add two row or column vectors of the same dimension, just add their components.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_D \end{bmatrix} \quad \mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_D + y_D \end{bmatrix}$$

You **cannot** add a row vector to a column vector or vice versa (unless the dimension is 1). You cannot add vectors of different dimensions.

To multiply a vector $\mathbf{x} \in \mathbb{R}^D$ by a scalar $\alpha \in \mathbb{R}$, just multiply each component.

$$\alpha\mathbf{x} = \begin{bmatrix} \alpha x_1 & \alpha x_2 & \cdots & \alpha x_D \end{bmatrix}^T$$

# Dot product

The **dot product** between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ is computed by multiplying the corresponding components of $\mathbf{x}$ and $\mathbf{y}$ and adding up the products.

$$\begin{aligned}\mathbf{x} \cdot \mathbf{y} &= x_1 y_1 + x_2 y_2 + \cdots + x_D y_D \\ &= \sum_{i=1}^{N} x_i y_i\end{aligned}$$

The dot product is also called the **inner product** or **scalar product**.

Alternative notations:

- $\mathbf{x}^T \mathbf{y}$ (dot product as a matrix multiplication)
- $(\mathbf{x}, \mathbf{y})$ or $\langle \mathbf{x} | \mathbf{y} \rangle$ (braket notation, common in Physics)
- $\mathbf{xy}$ implicit notation

# Matrices

A rectangular array of numbers. E.g.:

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 6 \end{bmatrix} \in \mathbb{R}^{2\times 3}$$

$A$ can be thought of as a horizontal stack of $M$ row vectors or a vertical stack of $N$ column vectors.

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_N \\ | & | & & | \end{bmatrix} \quad A = \begin{bmatrix} — & \mathbf{a}_1^T & — \\ — & \mathbf{a}_2^T & — \\ & \vdots & \\ — & \mathbf{a}_M^T & — \end{bmatrix}$$

**Beware of the overloaded notation!** $\mathbf{a}_i^T$ usually means the $i$th row rather than the $i$th column transposed.

# Matrices: transpose

The transpose of a matrix $A$, denoted $A^T$ is the same matrix with rows and columns interchanged. E.g.

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 6 \end{bmatrix} \in \mathbb{R}^{2\times 3} \qquad A^T = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 6 \end{bmatrix} \in \mathbb{R}^{3\times 2}$$

Note:

- $(A+B)^T = A^T + B^T$ (transpose is linear)
- $(\alpha A)^T = \alpha(A^T)$ for $\alpha \in \mathbb{R}$ (transpose is linear)
- $\alpha^T = \alpha$ for $\alpha \in \mathbb{R}$ (transpose a scalar does nothing)

# Adding and scaling matrices

Matrices of the same dimensions can be added together in the same way as with vectors: just add the corresponding components.

Matrices can be scaled by scalars by just multiplying each element by the scalar.

# Matrix multiplication

Two matrices can be multiplied if the number of columns in the first matrix equals the number of rows in the second matrix.

E.g. it is possible to multiply $A$ and $B$ if $A \in \mathbb{R}^{M \times N}$ and $B \in \mathbb{R}^{N \times D}$. The resulting matrix $AB$ will have dimension $M \times D$. **Rule**: inner dimension must match, outer dimension gives dimension of result.

The $ij$ element of the product $AB$ is equal to the inner product of row $\mathbf{a}_i^T$ from $A$ and column $\mathbf{b}_j$ from $B$:

$$(AB)_{ij} = \mathbf{a}_i^T \mathbf{b}_j = \sum_{k=1}^{M} a_{ik} b_{kj}$$

Multiplying two matrices requires performing $M \times D$ inner products.

# Matrix-vector multiplication

A matrix-vector multiplication $A\mathbf{x}$ can be performed if the number of columns in $A$ equals the number of rows (entries) in $\mathbf{x}$.

A matrix-vector product can be interpreted as a linear combination of the columns of $A$.

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_N \\ | & | & & | \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad A\mathbf{x} = x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots + x_N\mathbf{a}_N$$

I.e. each element of $\mathbf{x}$ scales a column of A and the result is the sum of these scaled columns.

It can also be interpreted as a set of $M$ inner products between rows of $A$ and $\mathbf{x}$ with each output being put in the resulting vector.

# Matrix vector multiplication

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

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

# Matrix-matrix multiplication

Matrix-matrix multiplication $AB$ can be interpreted as a series of matrix vector multiplications, one for each of the columns of $B$, with the results being stacked side-by-side in a matrix.

$$AB = \begin{bmatrix} A\mathbf{b}_1 & A\mathbf{b}_2 & \cdots & A\mathbf{b}_D \end{bmatrix}$$

Rules of matrix multiplication:

1. Not commutative: in general $AB \neq BA$
2. Distributive: $A(B + C) = AB + AC$
3. Associative: $A(BC) = (AB)C = ABC$
4. Transposes: $(AB)^T = B^T A^T$

# Square matrices, identities

A matrix is square if it has the same number of rows and columns. E.g. $A \in \mathbb{R}^{3\times 3}$.

The matrix $I$ is called the identity matrix. The $3 \times 3$ identity matrix is:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplication by the identity matrix gives back the original matrix.

- $AI = A$
- $IA = A$

# Inverses

The square matrix $B$ is said to be an inverse of the square matrix $A$ if $AB = I$.

The inverse of $A$ (if it exists) is denoted $A^{-1}$

- $AA^{-1} = I$
- $A^{-1}A = I$
- $(AB)^{-1} = B^{-1}A^{-1}$

Terminology:

- A square matrix that has an inverse is called **nonsingular** or **invertible** (also **nondegenerate**)
- A square matrix that has no inverse is called **singular** or **degenerate**

# Computing inverses

Most numerical computation packages include functions for computing inverses. The algorithm used is usually LU decomposition or Gauss-Jordan elimination.

E.g. Python `numpy.linalg.inv(A)`. MATLAB `inv(A)`.

You can invert $2 \times 2$ matrices by hand with:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

The matrix has no inverse if the determinant $\det(A) = 0$. In the $2 \times 2$ case the determinant is given by $\det(A) = ad - bc$

# Systems of linear equations

A system of linear equations can be solved by using the inverse. E.g. consider the linear system:

$$\begin{aligned} 3x + \quad 2y - \quad z &= \quad 1 \\ 2x - \quad 2y + 4z &= -2 \\ -x + 0.5y - \quad z &= \quad 0 \end{aligned}$$

This can be written in matrix form as $A\mathbf{x} = b$ with:

$$A = \begin{bmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & 0.5 & -1 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

And can be solved by finding the inverse of $A$ and multiplying by $\mathbf{b}$:

$$\mathbf{x} = A^{-1}\mathbf{b}$$

# Solving systems of linear eqns numerically

You could do the calculation in Python as follows:

```
import numpy as np
A = np.array([[3,2,-1],[2,-2,4],[-1, 0.5, -1]])
b = np.array([1, -2, 0])
x = np.dot(np.linalg.inv(A), b)
print(x)
```

Which gives the answer $(1, -2, -2)$.

Note that computing inverses is usually not the most efficient way of doing this. Just call `numpy.linalg.solve` directly.

```
x = np.linalg.solve(A, b)
```