Youssef SALMAN

Applied Mathematics
Lebanese University — Faculty of Sciences (Section V)

M1: Applied Mathematics & Statistics for Computational Sciences

October 8, 2025

AI (Artificial Intelligence): systems that perform tasks usually requiring human intelligence:

- **Language understanding**: e.g., *Google Translate* converting French $\rightarrow$ English.
- **Decision-making**: e.g., *autonomous cars* detecting pedestrians and deciding when to stop.

ML (Machine Learning): data-driven methods within AI that learn patterns from data:

- **Email spam filters**: trained on labeled messages ("spam" vs "not spam").
- **Recommendation systems**: e.g., *Netflix* predicting shows you may like.

AI is the broader ambition; ML is the statistical and algorithmic toolkit that makes it work.

Linear algebra:

- Images are matrices of pixel values.
- Neural nets apply linear transforms $Wx + b$ at each layer.
- PCA reduces dimensionality (e.g., compressing face images).

Calculus:

- Training $\approx$ minimizing a loss function.
- Gradient descent updates weights via derivatives.
- Example: adjusting millions of parameters in deep networks.

Probability & Statistics:

- Outputs are probabilities (e.g., disease detection with 85% confidence).
- Hypothesis testing $\rightarrow$ decide if improvements are significant.
- Bayesian models quantify uncertainty in predictions.

Face recognition: Image $\rightarrow$ vector (linear algebra) $\Rightarrow$ train weights with gradients (calculus) $\Rightarrow$ output a similarity score with confidence (probability/statistics).

Definition 1

A function $f : X \to Y$ assigns each $x \in X$ a unique $y = f(x) \in Y$.

Examples

- Linear: $f(x) = ax + b$ (e.g., linear regression).
- Quadratic: $f(x) = ax^2 + bx + c$ (optimization/control).
- Exponential: $f(x) = ae^{bx}$ (growth/decay, LR schedules).
- Linear: $3x + 2 = 0 \;\Rightarrow\; x = -\frac{2}{3}$
- Quadratic: $ax^2 + bx + c = 0 \;\Rightarrow\; x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

In ML: solving equations $\approx$ estimating parameters that satisfy optimality conditions (normal equations, gradients $= 0$).

Definition 2

A vector $\boldsymbol{v} = [v_1, \ldots, v_n] \in \mathbb{R}^n$.

Operations

- Addition: $\boldsymbol{a} + \boldsymbol{b} = [a_1 + b_1, \ldots, a_n + b_n]$
- Scalar mult.: $c\boldsymbol{v} = [cv_1, \ldots, cv_n]$
- Dot product: $\boldsymbol{a} \cdot \boldsymbol{b} = \sum_i a_i b_i$

Recommendation systems: user/movie vectors $\Rightarrow$ rating $\approx \boldsymbol{u} \cdot \boldsymbol{m}$.

- Angle: $\cos\theta = \dfrac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\| \, \|\boldsymbol{b}\|}$
- Orthogonality: $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = 0$
- Cauchy–Schwarz: $|\langle \boldsymbol{a}, \boldsymbol{b} \rangle| \leq \|\boldsymbol{a}\| \, \|\boldsymbol{b}\|$

Cosine similarity measures how close two vectors point in the same direction. It is the dot product divided by the product of their lengths (a normalized inner product), and it is widely used in Information Retrieval (IR) and Natural Language Processing (NLP) to compare text similarity.

Given $\boldsymbol{a} = (1, 2, 2)$ and $\boldsymbol{b} = (2, 0, 1)$:

1. Compute $\boldsymbol{a} \cdot \boldsymbol{b}$ and the angle.
2. Are they orthogonal?
3. Normalize both and recompute the dot product (cosine similarity). Do it with Python.

Definition 3

A **vector space** *V over a field F has operations $u + v \in V$ and $a \cdot v \in V$ ($a \in F$) satisfying the usual axioms (associativity, commutativity of addition, distributivity, identities, inverses).*

Example: $\mathbb{R}^3$ is a vector space over $\mathbb{R}$.

- **Linear independence:** $\sum_i \alpha_i \boldsymbol{v}_i = \mathbf{0} \;\Rightarrow\; \alpha_i = 0 \;\forall i$.
- **Span:** all linear combinations of a set.
- **Basis:** independent set that spans the space.
- **Dimension:** $\#$ vectors in a basis.

In $\mathbb{R}^2$, $e_1 = (1, 0)$ and $e_2 = (0, 1)$ form a basis $\Rightarrow \dim \mathbb{R}^2 = 2$.
PCA finds a low-dimensional basis capturing most variance.

- Stack vectors as columns in A and compute $\mathrm{rank}(A)$.
- If $\mathrm{rank}(A) = k$ (for k vectors), they are independent.
- Numerically: use SVD; beware near-dependencies (conditioning).

Consider

$$\boldsymbol{v}_1 = (1, 0, 1), \quad \boldsymbol{v}_2 = (0, 1, 1), \quad \boldsymbol{v}_3 = (1, 1, 2).$$

1. Are they independent? (hint: rank)
2. Find a basis of $\operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$.
3. What is the dimension of that span?Do it with Python.

- **Representation:** features $\rightarrow$ vectors/matrices.
- **Objective:** define loss (often quadratic, convex locally).
- **Optimization:** gradient-based updates.
- **Evaluation:** statistical metrics and uncertainty.

Strong math foundations $\Rightarrow$ interpretability, robustness, and reproducibility.

1. Refreshed algebra (functions, solving equations).
2. Built core objects: vectors, vector spaces, independence/basis/dimension.
3. Connected math to practice (representation, optimization, evaluation).

Next: hands-on practice in Python (NumPy) to operationalize these ideas.

C. M. Bishop and N. M. Nasrabadi (2006).
Pattern Recognition and Machine Learning.
Springer.

G. Weiss (1999).
Multiagent Systems: A Modern Approach to Distributed AI.
MIT Press.

Thank you for your attention!