Principles of Machine Learning

Lecture 1: Introduction and Course Overview

Sharif University of Technology
Dept. of Aerospace Engineering

February 11, 2025

Table of Contents

Course Overview

- **Course Title:** Principles of Machine Learning
- **Target Audience:** Undergraduate and graduate students from diverse academic backgrounds
- **Focus:** Balancing theoretical foundations with practical implementations
- **Key Tools:** Python, NumPy, Pandas, TensorFlow/Keras
- **Outcome:** Develop skills to understand, implement, and evaluate machine learning solutions

Course Objectives

By the end of this course, students will be able to:

1. Understand core machine learning concepts and algorithms
2. Design and implement ML solutions using Python
3. Select appropriate algorithms for different problems
4. Preprocess and prepare data for ML tasks
5. Evaluate and validate ML models
6. Create end-to-end ML solutions for real-world problems

Course Structure

- **Week 1:** Introduction to Machine Learning
- **Weeks 1-3:** Mathematical Foundations
- **Weeks 4-5:** Supervised Learning - Regression
- **Weeks 5-6:** Supervised Learning - Classification

- **Week 7:** Model Evaluation and Validation
- **Week 8:** Practical Considerations
- **Weeks 9-10:** Unsupervised Learning
- **Week 11:** Neural Networks Fundamentals
- **Weeks 12-13:** Deep Learning
- **Week 14:** Reinforcement Learning
- **Week 15:** Advanced Topics

Week 1: Introduction to Machine Learning

- What is machine learning?
- Types of machine learning problems
- Python programming basics (in Lab)
- Introduction to scientific computing (NumPy, Pandas) (in Lab)
- Case studies: ML applications in aerospace industry

Mathematical Foundations (Weeks 1-3)

- Linear algebra review
- Probability and statistics
- Calculus concepts
- Optimization basics

Supervised Learning (Weeks 4-6)

- **Regression:** Linear regression, polynomial regression, regularization techniques, gradient descent
- **Classification:** Logistic regression, SVMs, decision trees, random forests, k-Nearest Neighbors
- Applications: System modeling, fault detection, flight phase classification

Model Evaluation and Validation (Week 7)

- Cross-validation techniques
- Performance metrics (accuracy, precision, recall, F1-score)
- Bias-variance tradeoff
- Model selection and hyperparameter tuning

Practical Considerations (Week 8)

- Data cleaning, normalization, and transformation
- Feature selection and extraction techniques
- Industry case studies: Predictive maintenance, flight data analysis, aerospace design optimization
- Model deployment basics

Unsupervised Learning (Weeks 9-10)

- Clustering algorithms (K-means, Hierarchical, DBSCAN)
- Dimensionality reduction (PCA, t-SNE)
- Anomaly detection
- Applications: Process monitoring, pattern discovery, flight data clustering

Neural Networks and Deep Learning (Weeks 11-12)

- Neural Networks Fundamentals: Architecture, backpropagation, activation functions
- Deep Learning: CNNs, RNNs, time series data, modern architectures (Transformers), transfer learning
- Applications: Signal processing, computer vision, object detection in satellite imagery

Reinforcement Learning (Week 13)

- Markov Decision Processes
- Q-learning
- Policy Gradient methods
- Applications: Control systems, resource optimization

Advanced Topics (Week 15)

- Transfer learning
- Ensemble methods
- Explainable AI
- Few-shot and zero-shot learning

Machine Learning

- The science (and art) of programming computers to *learn from data*

- "To give computers the ability to learn without explicit programming"
  - *Arthur Samuel,* 1959

- "To learn from Experience E w.r.t some Task T and performance measure P, if its performance on T, as measured by P, improves with E"
  - *Tom Mitchell,* 1997

What is Machine Learning

There are three concepts at the core of ML [1]:

Data

- Training data
- The model input
- Stores valuable information

Model

- The input-output mapping
- Parametric
- Non-parametric

Learning

- Determining model's parameters
- Optimization
- Generalization

Why ML?

Traditional approach vs. ML approach

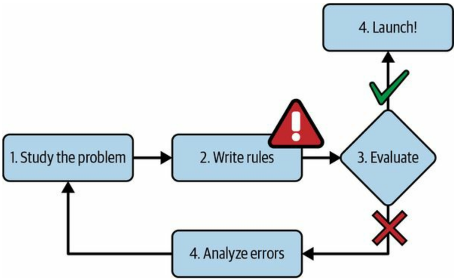


Figure: The traditional approach

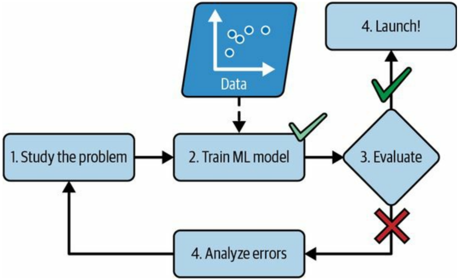


Figure: The machine learning approach

Traditional vs. Machine Learning-based solutions

Traditional approach:

- Expert knowledge
- High interpretability
- Well-known and typical scenarios
- Lacking access to sufficient amount of data

Machine learning-based approach:

- Data-rich scenarios
- Unseen settings
- Adaptability for various conditions
- Difficulty of deriving a governing model

Why ML?

Automatic adaptation for complex problems

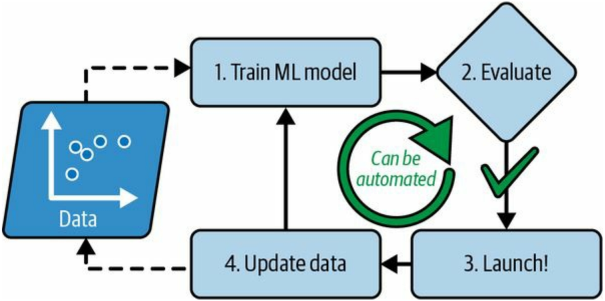


Figure: Automatically adapting to change

Why ML?

Inspect ML models to see what they have learned (*not always possible though*)

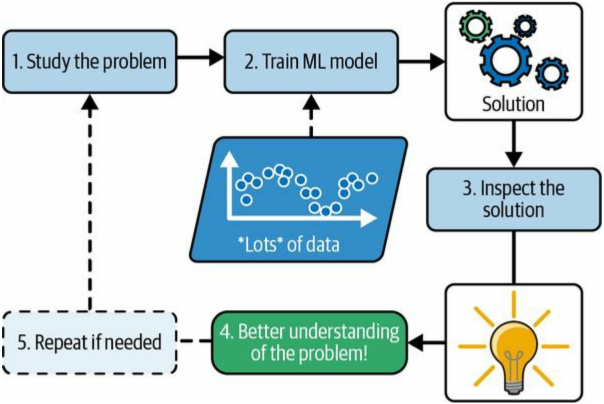


Figure: ML can help humans learn

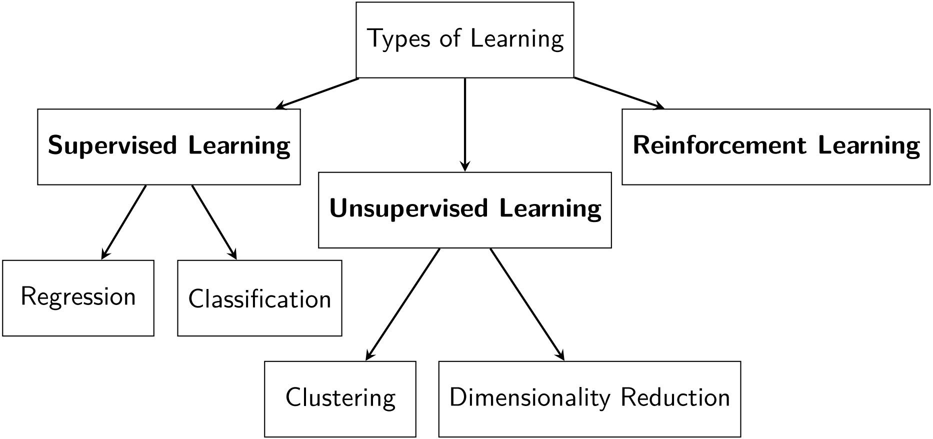
When ML?

- Problems with the traditional solutions being a long list of rules
- Complex problems with the traditional solutions being no good
- Fluctuating environments
- Revealing insights about the data

Examples of Applications

- Detecting tumors in brain scans
  (*CNNs/Transformers*)
- Forecasting a company's revenue next year
  (*Regression, Random forest, Artificial neural network*)
- Representing a complex, high-dimensional dataset in a clear and insightful diagram
  (*Dimensionality reduction*)
- Building an intelligent bot for a game
  (*RL*)

What is Machine Learning



* Some modern learning paradigms, such as semi-supervised learning, are not mentioned above.
* Deep learning, Deep Generative Modelling, etc. are not explicitly discussed in the domain of ML.

What is Machine Learning

Supervised Learning

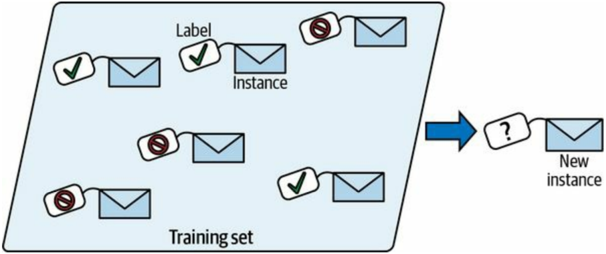


Figure: A labeled training set for spam classification (an example of supervised learning)

Supervised Learning - An Example

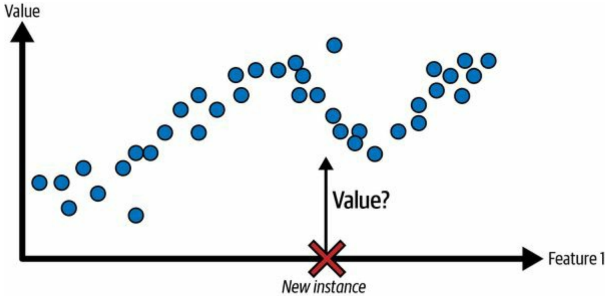


Figure: A regression problem: predict a value (or multiple values), given an input feature (or multiple input features)

Unsupervised Learning

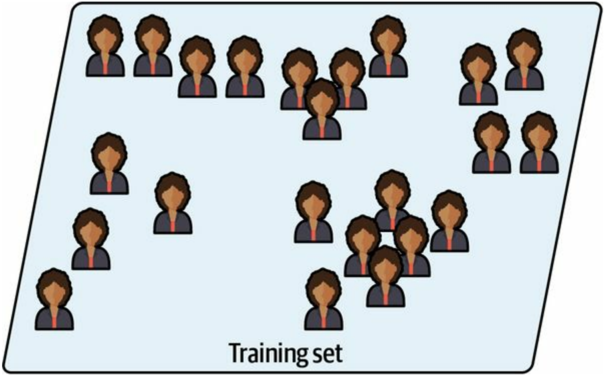


Figure: An unlabeled training set for unsupervised learning

Unsupervised Learning - An Example

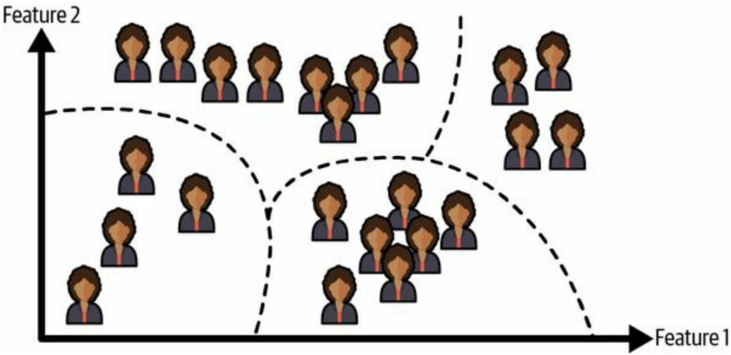


Figure: Clustering

Unsupervised Learning - An Example

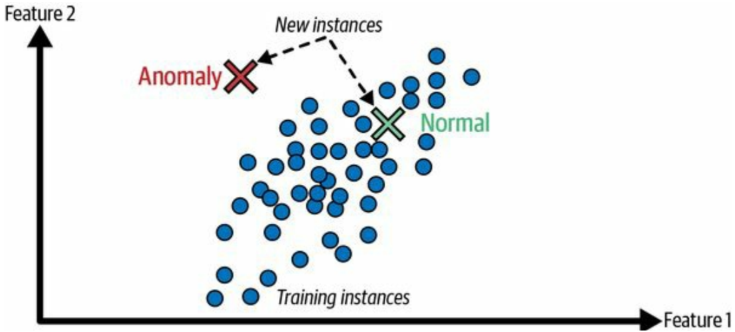


Figure: Anomaly detection

Semi-supervised learning

Combines unsupervised and supervised learning

- leverages both unlabeled and partially labeled data

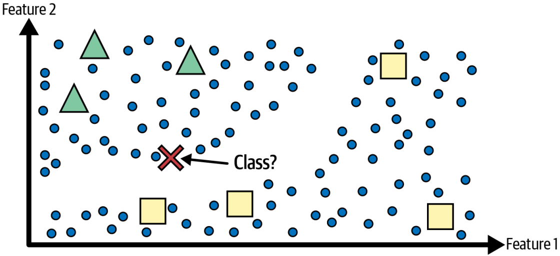


Figure: Semi-supervised learning with two classes (triangles and squares): the unlabeled examples (circles) help classify a new instance (the cross) into the triangle class rather than the square class, even though it is closer to the labeled squares

What is Machine Learning

Self-Supervised Learning

Generating a labeled dataset from a fully unlabeled dataset

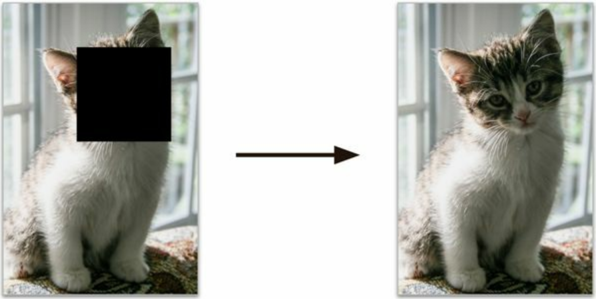


Figure: Self-supervised learning example: input (left) and target (right)

Reinforcement Learning

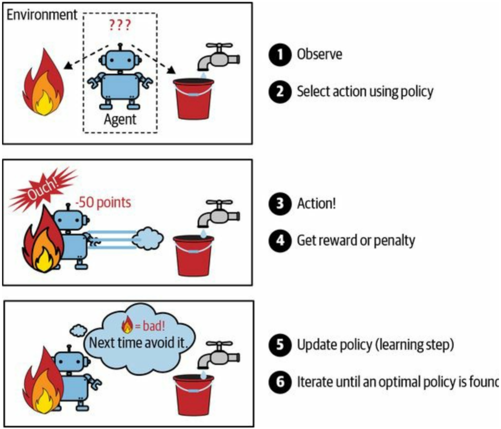


Figure: Reinforcement learning

Batch Learning vs. Online Learning

Batch Learning (Offline Learning)

- incapable of learning incrementally
- train using all available data
- training before launching into production
- susceptible to *model rot* or *data drift*
- a lot of computing resources (CPU, memory space, etc.)

Batch Learning vs. Online Learning

Online Learning

- capable of learning incrementally
- train using mini-batches
- capturing new data on the fly
- out-of-core learning
- adapt quickly to changing data

Batch Learning vs. Online Learning

Online Learning

- A model is trained, deployed, and continuously updated with new data.

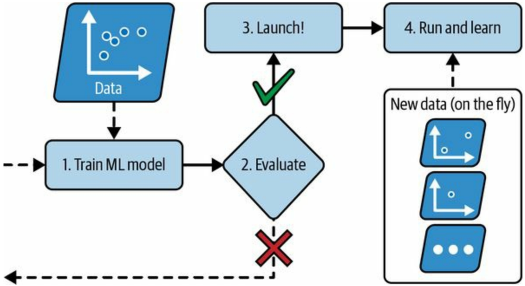


Figure: In online learning, a model is trained and launched into production, and then it keeps learning as new data comes in

Batch Learning vs. Online Learning

Online Learning

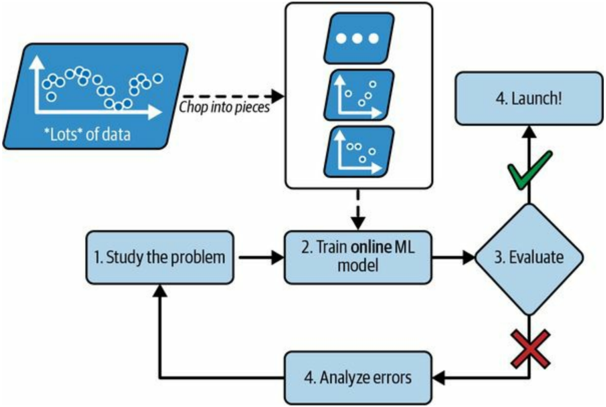


Figure: Using online learning to handle huge datasets

Instance-Based Learning vs. Model-Based Learning

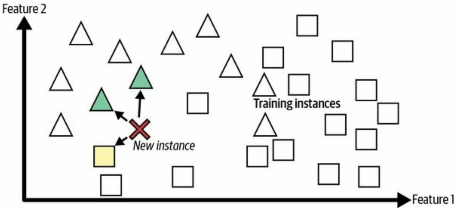


Figure: Instance-based learning

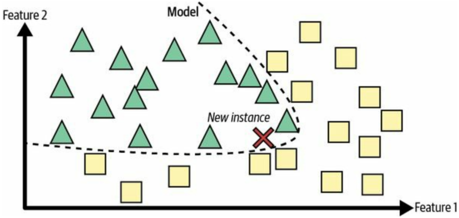


Figure: Model-based learning

Main Challenges of Machine Learning

1. Insufficient Quantity of Training Data

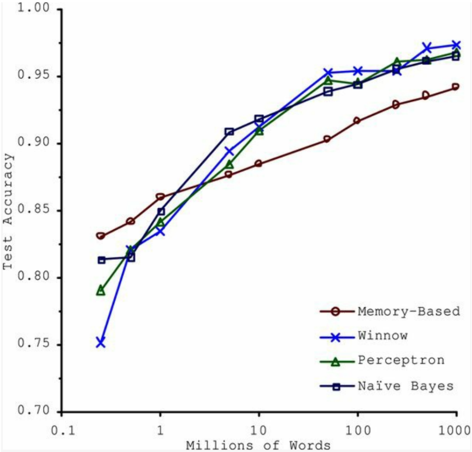


Figure: The importance of data versus algorithms

Main Challenges of Machine Learning

2. Nonrepresentative Training Data

Training data should be representative of the new cases to generalize to

- too small samples $\longrightarrow$ sampling noise
- very large samples $\longrightarrow$ sampling bias

Main Challenges of Machine Learning

3. Poor-Quality Data

Errors, outliers, noise, etc.

- Discard the outliers or fix the errors manually
- Handling missing features (ignore feature, ignore data, or fill in the missing values)

Main Challenges of Machine Learning

4. Irrelevant Features

Feature engineering:

- *Feature selection* (selecting the most useful features)
- *Feature extraction* (combining existing features to produce a useful one)
- Gathering new data $\longrightarrow$ Creating new features

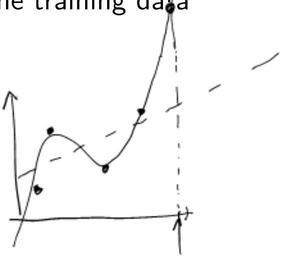
What is Machine Learning

Main Challenges of Machine Learning

5. Overfitting the Training Data

Performing well on training data but not generalizing well
Complex relative to the amount and noisiness of the training data
Possible solutions:

- Simplifying or constraining the model
- Gather more training data
- Reduce the noise in the training data



Main Challenges of Machine Learning

6. Underfitting the Training Data

Too simple to learn the underlying structure of data
Possible solutions:

- Select a model with more parameters
- Feed better features (feature engineering)
- Reduce the constraints

Testing and Validating

Hyperparameter Tuning and Model Selection

- Train the model on the *training set*
- Test the model on the *test set*
- *Generalization error*: The error rate on new cases
- Evaluate the model on the test set $\longrightarrow$ Estimating the generalization error
- The value above indicates how good your model performs on unseen data
- Common 80-20 splitting

But, what if the model is the best model only for *that particular set?*

Testing and Validation

Hyperparameter Tuning and Model Selection

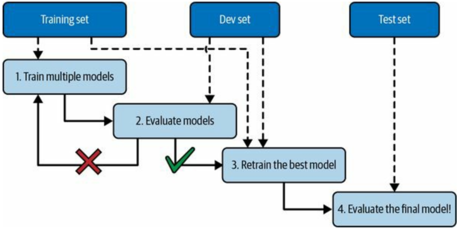


Figure: Model selection using holdout validation

Validation set, too small or too large $\longrightarrow$ cross validation

Data Mismatch

- Representativity of the test and validation (development) sets
- How to decide the poor performance is due to model overfitting or data mismatch $\longrightarrow$ using train-dev set

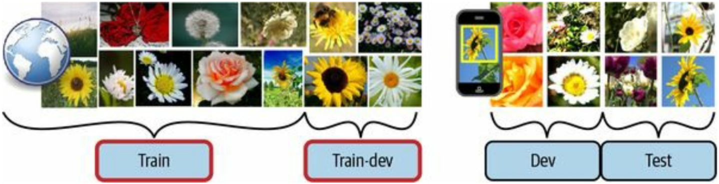


Table of Contents

Mathematical Foundations

- **Linear Algebra**
- Probability & Statistics
- Calculus
- Optimization

- Linear Algebra: The study of vectors and their manipulations
- Any mathematical object that satisfies the following two properties can be considered a **vector**:
  - $\mathbf{x} + \mathbf{y} = \mathbf{z}$
  - $\lambda \mathbf{x} = \mathbf{w}$, $\lambda \in \mathbb{R}$
- Examples of vectors: polynomials, audio signals, elements of $\mathbb{R}^n$

Definition (Matrix)

Matrix $\mathbf{A}$ is an $m \cdot n$-tuple of elements

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad a_{ij} = \mathbb{R}.$$

Definition (Matrix Addition and Multiplication)

The sum of two matrices $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{B} \in \mathbb{R}^{m\times n}$:

$$\mathbf{A} + \mathbf{B} := \begin{bmatrix} a_{11} + b_{11} & \dots & a_{1n} + b_{1n} \\ \vdots & & \vdots \\ a_{m1} + b_{m1} & \dots & a_{mn} + b_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

For matrices $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times k}$, the elements of the product $\mathbf{C} = \mathbf{AB} \in \mathbb{R}^{m\times k}$ are:

$$c_{ij} = \sum_{l=1}^{n} a_{il} b_{lj}, \qquad i = 1, \dots, m, \quad j = 1, \dots, k.$$

Mathematical Foundations: Linear Algebra

Definition (Associativity)

$$\forall \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{n \times p}, \mathbf{C} \in \mathbb{R}^{p \times q} : \ (\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$$

Definition (Distributivity)

$$\begin{aligned} \forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}, \mathbf{C}, \mathbf{D} \in \mathbb{R}^{n \times p} : (\mathbf{A} + \mathbf{B})\mathbf{C} &= \mathbf{AC} + \mathbf{BC} \\ \mathbf{A}(\mathbf{C} + \mathbf{D}) &= \mathbf{AC} + \mathbf{AD} \end{aligned}$$

Definition (Inverse)

For $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ is the *inverse* of $\mathbf{A}$ and denoted as $\mathbf{A}^{-1}$ if:

$$\mathbf{AB} = \mathbf{I}_n = \mathbf{BA}$$

The inverse does exist $\longrightarrow$ $\mathbf{A}$ is *invertible/nonsingular*

Definition (Transpose)

For $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$ is the *transpose* of $\mathbf{A}$ and denoted as $\mathbf{A}^\top$ if:

$$b_{ij} = a_{ji}$$

If $\mathbf{A} = \mathbf{A}^\top \longrightarrow \mathbf{A}$ is *symmetric*

Mathematical Foundations: Linear Algebra

A system of linear equations as below

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$
$$\vdots$$
$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$

can be formulated using matrices notation as

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

How to solve them?

$$\begin{bmatrix} a_{11} & a_{1n} \\ & \\ a_{m1} & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Example 1. For $a \in \mathbb{R}$, we seek all solutions of the following system of equations:

$$
\begin{array}{ccccccccccc}
-2x_1 & + & 4x_2 & - & 2x_3 & - & x_4 & + & 4x_5 & = & -3 \\
4x_1 & - & 8x_2 & + & 3x_3 & - & 3x_4 & + & x_5 & = & 2 \\
x_1 & - & 2x_2 & + & x_3 & - & x_4 & + & x_5 & = & 0 \\
x_1 & - & -2x_2 & & & - & 3x_4 & + & 4x_5 & = & a
\end{array}
$$

Mathematical Foundations: Linear Algebra

Example 1.

$$\mathbf{Ax} = \mathbf{b} \xrightarrow{\textit{augmented matrix}} [\mathbf{A} \mid \mathbf{b}]:$$

$$\left[\begin{array}{ccccc|c} -2 & 4 & -2 & -1 & 4 & -3 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ 1 & -2 & 1 & -1 & 1 & 0 \\ 1 & -2 & 0 & -3 & 4 & a \end{array}\right]$$

After applying some *elementary transformations*:

$$\rightsquigarrow \left[\begin{array}{ccccc|c} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 3 & -2 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array}\right]$$

Known as *row-echelon form*.

Only for $a = 1$ this system can be solved.

Example 1.

A *particular solution* is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}.$$

The *general solution* is

$$\left\{ \mathbf{x} \in \mathbb{R}^5 : \mathbf{x} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 2 \\ 0 \\ -1 \\ 2 \\ 1 \end{bmatrix}, \quad \lambda_1, \lambda_2 \in \mathbb{R} \right\}.$$

Definition (Row-Echelon Form)

A matrix is in row-echelon form if

- All rows that contain only zeros are at the bottom of the matrix; correspondingly, all rows that contain at least one nonzero element are on top of rows that contain only zeros.
- Looking at nonzero rows only, the first nonzero number from the left (also called the *pivot*) is always strictly to the right of the pivot of the row above it.

Definition (Reduced Row-Echelon Form)

An equation system is in *rref* or *row canonical form* if

- It is in row-echelon form.
- Every pivot is 1.
- The pivot is the only non-zero entry in its column.

Definition

Gaussian elimination is an algorithm that performs elementary transformations to bring a system of linear equations into reduced row-echelon form.

Example 2. Calculating an Inverse Matrix by Gaussian Elimination:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\xrightarrow{augmented\ \ matrix}$$

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Example 2.

After using Gaussian Elimination to reach rref:

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -1 & 2 & -2 & 2 \\ 0 & 1 & 0 & 0 & 1 & -1 & 2 & -2 \\ 0 & 0 & 1 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & -1 & 2 \end{array}\right]$$

Thus,

$$\mathbf{A}^{-1} = \begin{bmatrix} -1 & 2 & -2 & 2 \\ 1 & -1 & 2 & -2 \\ 1 & -1 & 1 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

Check this by $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$