Machine Learning

(Course Code: 18B1WCI634 / 18B11BI611)

Introduction to Machine Learning

Mr. Sandeep Kumar Patel
Assistant Professor
Jaypee University of Information Technology
Waknaghat, Solan, HP-173234

What is Machine Learning?

- Machine learning uses **computational methods** to Learn from **experience** to improve **performance** to make accurate **predictions**.

Definition

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Experience and Data

- **Definition:** Experience = past information (comes from Data) available to the learner.
- **Data storage:** Usually stored electronically in databases, files, or cloud.
- **Types of data:**
  - **Human-labeled data:** Data manually tagged with correct outputs.
    Example: Emails labeled as spam or not spam.
  - **Environment-generated data:** Data collected automatically through interactions.
    Example: Sensor readings from machines, user clicks on a website.
- **Importance of data quality and size:**
  - High-quality data ensures correct learning.
  - Larger datasets usually improve model performance.
  - Noisy or incomplete data can reduce prediction accuracy.

Example:1

Learning Problem: Handwriting Recognition

- **Task (T):** Recognizing and classifying handwritten words in images.
- **Performance (P):** Percentage of correctly classified words.
- **Experience (E):** A database of handwritten words with correct labels.

Frame Title

Learning Problem:Spam Detection

- **Problem:** We have a collection of emails, each labeled as **Spam** or **Not Spam**.
- **Training Data:** Use a finite, randomly selected subset of emails to train the model.
- **Goal:** Predict whether a new, unseen email is spam or not.
- **Factors Affecting Accuracy:**
  - **Sample Size:** More emails improve learning accuracy.
  - **Label Quality:** Misclassified emails reduce model performance.
  - **Email Variety:** Different formats or topics increase complexity.

Learning Algorithms and Complexity

- **Computational Algorithms:** Machine learning models are essentially algorithms designed to make predictions from data.
- **Algorithm Requirements:**
  - **Accurate:** Should correctly predict outputs for new data. Example: Spam detector correctly identifies spam emails.
  - **Efficient:** Should run quickly and use reasonable memory.
- **Quality Measures:**
  - **Time Complexity:** How fast the algorithm runs on large datasets.
  - **Space Complexity:** How much memory the algorithm uses.
  - **Sample Complexity:** How much training data is required to learn effectively.

Types of Problems Tackled by Machine Learning

- Machine learning can solve a wide variety of real-world problems.
- Document classification is one example: predicting the topic or label of a text.
- Many other applications exist across different domains.

Text and Document Processing

- **Text / Document Classification:**
  - Assigning topics to documents
  - Detecting inappropriate content on webpages
  - Spam detection in emails
- **Natural Language Processing (NLP):**
  - Part-of-speech tagging
  - Named-entity recognition
  - Context-free parsing, dependency parsing
  - These are structured prediction problems (output has structure)

Speech and Computer Vision Applications

- **Speech Processing:**
  - Speech recognition and synthesis
  - Speaker verification and identification
  - Language and acoustic modeling
- **Computer Vision:**
  - Object recognition and identification
  - Face detection
  - Optical Character Recognition (OCR)
  - Content-based image retrieval, pose estimation

Computational Biology and Other Applications

- **Computational Biology:**
  - Protein function prediction
  - Identification of key sites
  - Analysis of gene and protein networks
- **Other Applications:**
  - Fraud detection (credit cards, insurance, telecom)
  - Network intrusion detection
  - Game playing: chess, Go, backgammon
  - Autonomous vehicles: robots, self-driving cars
  - Medical diagnosis
  - Recommendation systems, search engines, information extraction

Standard Machine Learning Tasks

- Machine learning studies several common tasks, each with different goals.
- Main practical objectives:
  - Generate accurate predictions for unseen items
  - Design efficient and robust algorithms for large-scale data

Classification

- Assign a category or label to each item.
- Examples:
  - Document classification: Politics, Sports, Business, Weather
  - Image classification: Car, Train, Plane
  - OCR, Text classification, Speech recognition
- Number of categories: Often ¡ few hundreds, can be unbounded in complex tasks.

Regression and Ranking

- **Regression:** Predict a real-valued number for each item
  - Examples: Stock prices, Economic variable predictions
  - Error depends on magnitude of difference between predicted and true values
- **Ranking:** Learn to order items according to a criterion
  - Example: Web search results ranking
  - Used in information retrieval and NLP systems

Clustering and Dimensionality Reduction

- **Clustering:** Partition items into homogeneous subsets
  - Example: Identify communities in social networks
  - Useful for large dataset analysis
- **Dimensionality Reduction / Manifold Learning:**
  - Transform high-dimensional data to lower dimensions
  - Example: Preprocessing digital images in computer vision
  - Goal: Preserve key properties while reducing complexity

Key Questions in Machine Learning

- What kinds of concepts or patterns can be learned?
- Under what conditions can they be learned efficiently?
- How well can these concepts be learned computationally?
- Goal: Balance **accuracy**, **efficiency**, and **robustness**.

Spam Detection: A Running Example

- Task: Automatically classify emails as **spam** or **non-spam**.
- Use this problem to illustrate:
  - Key definitions in machine learning
  - Stages of learning
  - Evaluation methods

Key Definitions in Machine Learning

- **Examples:** Items or instances of data used for learning or evaluation. *Example: Emails in our dataset.*
- **Features:** Attributes representing an example (often a vector). *Example: Email length, sender name, keywords, header info.*
- **Labels:** Categories or values assigned to examples. *Example:* {*spam*, *non-spam*} *for classification; real numbers for regression.*
- **Hyperparameters:** Free parameters specified before learning. *Example: Learning rate, regularization strength.*

Training, Validation, and Test Samples

- **Training sample:** Used to train the algorithm.
- **Validation sample:** Used to tune hyperparameters.
- **Test sample:** Used to evaluate performance on unseen data.
- Example (spam detection):
  - Training: 5000 labeled emails
  - Validation: 1000 emails to tune parameters
  - Test: 2000 emails to measure accuracy

Loss Functions

- Measure difference between predicted label and true label.
- Common loss functions:
  - **Zero-one loss:** Counts misclassifications $L(y, \hat{y}) = 1_{\hat{y} \neq y}$
  - **Squared loss:** Used for real-valued predictions $L(y, \hat{y}) = (y - \hat{y})^2$
- Loss guides the learning algorithm to improve predictions.

Hypothesis Set

- A set of functions mapping features to labels.
- Examples for spam detection:
  - Functions mapping email features to $\{\text{spam, non-spam}\}$
  - Linear functions mapping feature vectors to scores ($\mathbb{R}$)
- Higher scores may indicate higher likelihood of being spam.
- The learning algorithm selects the best function from the hypothesis set based on training data.

Learning Stages

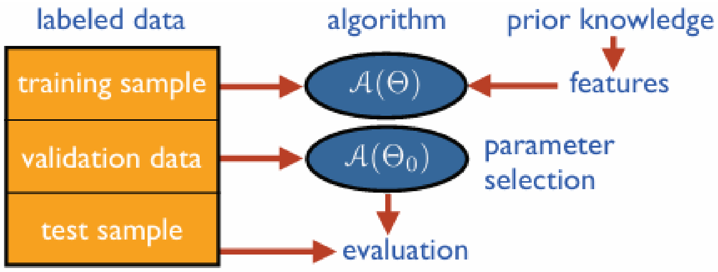

Figure 1: Data $\rightarrow$ Features $\rightarrow$ Training $\rightarrow$ Testing

Learning Stages: Spam Detection Example

Step 1: Data Partitioning

- Start with a collection of labeled emails.
- Randomly split data into:
  - Training sample
  - Validation sample
  - Test sample
- Training sample is usually larger when total data is small.
- Validation sample size depends on the number of hyperparameters (Θ) of the algorithm.

Learning Stages: Feature Selection

Step 2: Feature Extraction and Selection

- Assign relevant features to each email (e.g., word frequency, sender, subject line).
- Good features guide the learning algorithm effectively.
- Poor or uninformative features can mislead the model.
- Feature choice relies on prior knowledge about the problem.

Learning Stages: Training and Hyperparameter Tuning

Step 3: Algorithm Training and Hyperparameter Tuning

- Use selected features to train the learning algorithm A.
- Tune hyperparameters Θ (e.g., regularization, learning rate).
- Each hyperparameter setting produces a different hypothesis.
- Choose hypothesis Θ_0 that performs best on the validation sample.

Learning Stages: Testing and Evaluation

Step 4: Testing and Performance Evaluation

- Use the selected hypothesis to predict labels for the test sample.
- Evaluate performance using an appropriate loss function:
  - Example: zero-one loss for spam detection
- Test error, not training error, determines algorithm performance.

Learning Scenarios

- Machine learning scenarios differ based on:
  - Type of training data available
  - Order and method of data acquisition
  - Nature of test data used for evaluation
- These factors influence model design, evaluation, and performance.

Supervised Learning

- Learner receives a set of **labeled examples**.
- Objective: Predict labels for unseen data points.
- Commonly used for:
  - Classification
  - Regression
  - Ranking
- **Example:** Spam detection problem.

Unsupervised Learning

- Learner receives only **unlabeled data**.
- No explicit ground truth for evaluation.
- Quantitative evaluation is often difficult.
- Typical applications:
    - Clustering
    - Dimensionality reduction

Semi-Supervised Learning

- Training data contains:
  - A small set of labeled examples
  - A large set of unlabeled examples
- Useful when labeling is expensive.
- Applicable to classification, regression, and ranking tasks.
- Goal: Improve performance using unlabeled data distribution.

Transductive Inference

- Learner receives:
  - Labeled training data
  - A fixed set of unlabeled test points
- Objective: Predict labels only for the given test points.
- Often easier than inductive learning.
- Performance guarantees remain an active research topic.

Online Learning

- Learning occurs over multiple rounds.
- At each round:
  - Receive an unlabeled instance
  - Make a prediction
  - Observe true label and incur loss
- Goal:
  - Minimize cumulative loss or regret
- No distributional assumptions; data may be adversarial.

Reinforcement Learning

- Learner actively interacts with an environment.
- Takes actions and receives immediate rewards.
- Objective: Maximize cumulative reward.
- Challenges:
  - No explicit labeled data
  - Exploration vs. exploitation dilemma

Active Learning

- Learner selectively queries an oracle for labels.
- Goal: Achieve high accuracy with fewer labeled examples.
- Particularly useful when labeling is costly.
- Common applications:
  - Computational biology
  - Medical imaging

Generalization in Machine Learning

- Machine learning is fundamentally about **generalization**.
- Goal: Use a finite set of labeled examples to make accurate predictions on unseen data.
- Central challenge: Learning patterns that extend beyond the training sample.

Supervised Learning Perspective

- Given a finite sample of labeled examples.
- Learning task is formulated as:
  - Selecting a function from a **hypothesis set**
  - Hypothesis set is a subset of all possible functions
- The selected hypothesis is used to label all instances, including unseen data.

Choosing the Hypothesis Set

- A key question: **How should the hypothesis set be chosen?**
- Trade-off depends on the complexity of the hypothesis family.
- Two possible scenarios:
  - Complex hypothesis set
  - Simple hypothesis set

Complex vs. Simple Hypothesis Sets

- **Complex hypothesis set:**
  - Can perfectly fit the training data
  - May commit zero training error
  - Risk of memorizing training samples
- **Simple hypothesis set:**
  - May incur training errors
  - Provides smoother decision boundaries

Training Accuracy vs. Generalization

- Best predictor on training data is not always best overall.
- Perfect training accuracy does not guarantee good generalization.
- Generalization is different from memorization.

Illustration of Model Complexity

- A complex model may create a zig-zag decision boundary.
- A simple model produces a smoother decision boundary.
- Smooth boundaries often generalize better to unseen data.

Overfitting and Underfitting

- **Overfitting:**
  - Hypothesis set is too complex
  - Training error is low, but test error is high
- **Underfitting:**
  - Hypothesis set is too simple
  - Cannot capture underlying patterns

Sample Size and Complexity Trade-off

- Generalization depends on:
  - Size of the training sample
  - Complexity of the hypothesis set
- Small sample $+$ complex model $\rightarrow$ overfitting
- Simple model $+$ large bias $\rightarrow$ underfitting

Towards Theoretical Guarantees

- Understanding generalization requires formal analysis.
- Learning guarantees depend on:
  - Notions of hypothesis complexity
  - Sample size
- These concepts form the foundation of statistical learning theory.

Prerequisites of Machine Learning

- Machine learning systems require certain fundamental components to function effectively.
- These components enable learning from data and generalization to unseen examples.

Core Prerequisites

1. Data
2. Model (Hypothesis Space)
3. Learning Algorithm

Prerequisite 1: Data

- Data provides the **experience** from which the machine learns.
- Can be:
  - Labeled (Supervised Learning)
  - Unlabeled (Unsupervised Learning)
  - Partially labeled (Semi-supervised Learning)
- Quality, quantity, and diversity of data strongly influence performance.

Prerequisite 2: Model

- A model represents a set of possible functions (hypotheses).
- It defines how input data is mapped to output predictions.
- Examples:
  - Linear models
  - Decision trees
  - Neural networks
- Model complexity affects generalization and overfitting.

Prerequisite 3: Learning Algorithm

- The learning algorithm selects the best model using training data.
- It optimizes model parameters by minimizing a loss function.
- Common techniques:
  - Gradient Descent
  - Backpropagation
  - Maximum Likelihood Estimation