

UNIT 4

Learning

(4 Hours) — Artificial Intelligence (CACS410)

⚠ Please Note

These notes were prepared for quick revision and **may contain errors or omissions**. Please take care: cross-check important definitions, formulas, algorithms and worked examples against your prescribed textbook, class notes and teacher's instructions before relying on them (especially for exams).

4.1 Concepts of Machine Learning

What is Learning?

Learning is any change in a system that allows it to perform better the next time it faces the same or a similar task (Herbert Simon). **Machine learning (ML)** is the part of AI that builds programs which improve automatically through experience. Tom Mitchell's formal definition: a computer program is said to learn from experience **E** with respect to a class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.

Example — A spam filter

T: classify emails as spam or not spam. **E**: a collection of emails labelled by users. **P**: percentage of emails classified correctly.

Why Machine Learning?

- Programmers cannot anticipate all situations or write rules for tasks like face or speech recognition.
- Environments change over time, so the system must adapt.
- Large amounts of data contain patterns that humans cannot find manually (data mining).

Components of a Learning System

- **Training data / experience** — examples from which to learn.

- **Representation (model)** — how the learned knowledge is stored: rules, decision tree, neural network, probabilities.
- **Learning algorithm** — adjusts the model to fit the data.
- **Feedback** — labels, rewards or a performance measure telling how well the system is doing.
- **Evaluation** — testing on unseen data to measure generalization.

Learning process: collect data → prepare/clean data → split into *training* and *test* sets → train the model → evaluate on the test set → deploy and monitor. **Overfitting** means the model memorizes the training data and performs poorly on new data; **underfitting** means it is too simple to capture the pattern.

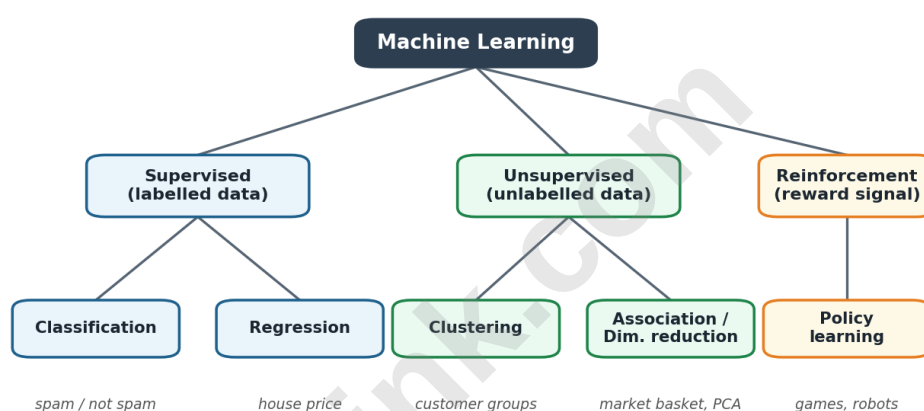


Figure 4.1 — Main categories of machine learning

4.2 Forms of Learning

Rote Learning

Learning by **memorization**: the system stores facts or computed results and retrieves them later instead of recomputing. There is no generalization. Example: Samuel's checkers program stored the evaluated value of board positions so that it did not need to search them again; a calculator caching results; a student memorizing a multiplication table.

Advantages: simple, speeds up repeated tasks. **Disadvantages:** needs a lot of storage, cannot handle new situations, retrieval cost grows with memory.

Learning by Analogy

The system solves a new problem by **transferring knowledge from a similar, previously solved problem**. Steps: (1) retrieve a similar source case, (2) map the elements of the source to the target problem, (3) transfer and adapt the solution, (4) evaluate and store the new case. Examples: understanding the atom as a “solar system” (nucleus = sun, electrons = planets); **case-based reasoning** where a help-desk system reuses the fix from a similar earlier ticket.

Inductive Learning (Learning from Examples)

Induction derives a **general rule from specific examples**: given examples $(x, f(x))$, find a hypothesis h that agrees with f on the examples and generalizes to new inputs. The conclusions are probable, not guaranteed. The preference for one hypothesis over another is the **inductive bias** (e.g., Ockham's razor: prefer the simplest consistent hypothesis).

Example — Learning the concept “bird”

Examples: sparrow (feathers, wings, lays eggs) = bird; eagle = bird; bat (wings, no feathers) = not a bird; fish = not a bird.

Induced rule: **IF has feathers THEN bird.**

Decision tree learning is a common inductive method. The tree is built top-down; at each step the attribute giving the best split is chosen using **entropy** and **information gain**:

- **Entropy** $H(S) = - \sum p_i \log_2 p_i$ (0 for a pure set, 1 for a 50/50 binary set).
- **Information gain** $\text{Gain}(S, A) = H(S) - \sum (|S_v| / |S|) \cdot H(S_v)$, summed over the values v of attribute A .

Example — Entropy of a dataset

A dataset has 14 examples: 9 “play tennis = yes” and 5 “no”.

$$H(S) = -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) = 0.410 + 0.530 = \mathbf{0.940}.$$

For this classic weather dataset, the attribute Outlook has the highest information gain (0.246), so it becomes the root of the tree.

Explanation-Based Learning (EBL)

EBL learns a general rule from a **single training example** by using **domain knowledge**. The system *explains* why the example is an instance of the target concept, then *generalizes* the explanation. Inputs: target concept, one training example, a domain theory, and an **operationality criterion** (the learned rule must use easily checkable features).

Example — Learning the concept “Cup”

Domain theory: $\text{Cup}(x) \leftarrow \text{Stable}(x) \wedge \text{Liftable}(x) \wedge \text{OpenVessel}(x)$; $\text{Stable}(x) \leftarrow \text{HasFlatBottom}(x)$; $\text{Liftable}(x) \leftarrow \text{Light}(x) \wedge \text{HasHandle}(x)$; $\text{OpenVessel}(x) \leftarrow \text{HasConcavity}(x) \wedge \text{ConcavityPointsUp}(x)$.

Training example: obj1 is light, has a handle, a flat bottom, an upward concavity, is red, and is owned by Ram.

Learned rule

Explain why obj1 is a cup using the domain theory — colour and owner are irrelevant and are ignored. **Generalize** by replacing obj1 with a variable:

$\text{Cup}(x) \leftarrow \text{Light}(x) \wedge \text{HasHandle}(x) \wedge \text{HasFlatBottom}(x) \wedge \text{HasConcavity}(x) \wedge \text{ConcavityPointsUp}(x)$

Advantages: learns from very few examples, speeds up later reasoning. **Disadvantages:** needs a correct and complete domain theory; it only re-expresses knowledge already implied by the theory, it does not discover truly new facts.

Supervised, Unsupervised (and Reinforcement) Learning

Point	Supervised	Unsupervised	Reinforcement
Data	Labelled examples (x, y)	Unlabelled data x only	Reward / penalty from the environment
Goal	Learn a mapping $x \rightarrow y$	Discover structure in the data	Learn a policy that maximizes total reward
Tasks	Classification (discrete output), regression (continuous output)	Clustering, association rules, dimensionality reduction	Game playing, robot control
Algorithms	Decision trees, k-NN, SVM, Naïve Bayes, linear regression, neural networks	k-means, hierarchical clustering, PCA, Apriori	Q-learning, policy gradients
Example	Predicting house price; spam detection	Grouping customers by buying behavior	A program learning to play chess by winning and losing

Example — k-means clustering (unsupervised)

Data (1-D): 1, 2, 3, 10, 11, 12 with $k = 2$ and initial centroids 1 and 10.

Assign each point to the nearest centroid: cluster A = {1, 2, 3}, cluster B = {10, 11, 12}.

Recompute centroids: A = 2, B = 11.

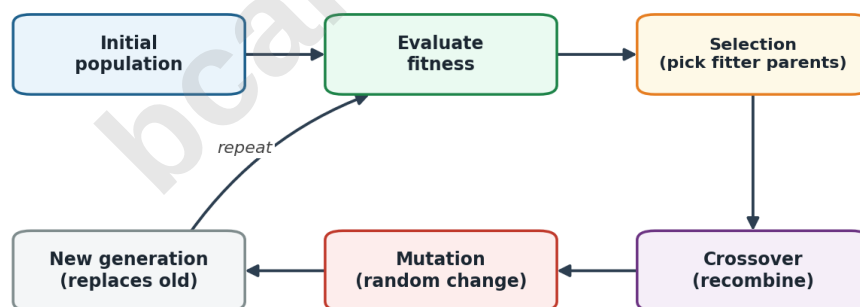
Re-assigning gives the same clusters, so the algorithm has converged.

Semi-supervised learning uses a small amount of labelled data with a large amount of unlabelled data.

Learning by Evolution — Genetic Algorithms

A **genetic algorithm (GA)** is a search and optimization technique inspired by **natural selection** (survival of the fittest). It evolves a **population** of candidate solutions over generations using selection, crossover and mutation (John Holland, 1970s).

Term	Meaning
Chromosome (individual)	A candidate solution encoded as a string, e.g., the bits 01101
Gene	One position/element of the chromosome
Population	The set of chromosomes in the current generation
Fitness function	Measures how good a chromosome is as a solution
Selection	Choosing fitter individuals as parents (roulette wheel, tournament, rank)
Crossover	Combining parts of two parents to create offspring (single-point, two-point, uniform)
Mutation	Randomly changing a gene (e.g., flipping a bit) to maintain diversity



Stop when a good-enough solution is found or the maximum number of generations is reached

Figure 4.2 — The genetic algorithm cycle

1. Initialize a random population of chromosomes
2. Evaluate the fitness of each chromosome
3. Repeat until a termination condition is met:
 - a. Select parents according to fitness
 - b. Apply crossover (with probability p_c) to produce offspring
 - c. Apply mutation (with a small probability p_m) to offspring
 - d. Replace the old population with the new generation
 - e. Evaluate the fitness of the new population
4. Return the best chromosome found

Example — Maximize $f(x) = x^2$ for x in $0 \dots 31$ using 5-bit chromosomes

Initial population of 4: 01101 ($x = 13$), 11000 ($x = 24$), 01000 ($x = 8$), 10011 ($x = 19$).

Fitness $f(x) = x^2$: 169, 576, 64, 361. Total = 1170, average = 292.5.

One generation

Selection (roulette wheel): probability = fitness / total: 0.144, 0.492, 0.055, 0.309. The strongest chromosome 11000 is expected to get about 2 copies; 01000 is likely to die out.

Crossover (single point after the 4th bit) of parents 01101 and 11000: 0110|1 $\times$ 1100|0 $\rightarrow$ offspring **01100** ($12 \rightarrow 144$) and **11001** ($25 \rightarrow 625$).

Mutation flipping the 3rd bit of 11001 gives **11101** ($29 \rightarrow 841$). Fitness is rising generation by generation; the optimum is 11111 ($31 \rightarrow 961$).

Advantages: works without derivatives, handles large and complex search spaces, can escape local optima, easy to parallelize. **Disadvantages:** no guarantee of the global optimum, needs a good fitness function and parameter tuning, can be slow. **Applications:** scheduling, travelling salesman problem, design optimization, evolving neural networks, game strategies. **Genetic programming** applies the same idea to evolve programs.

Key Takeaways

- ✓ Machine learning = improving performance P at task T with experience E (Mitchell).
- ✓ Rote learning memorizes; analogy transfers solutions from similar cases; induction generalizes rules from examples; EBL generalizes one example using domain theory.
- ✓ Supervised uses labelled data (classification/regression); unsupervised finds structure (clustering); reinforcement learns from rewards.
- ✓ Entropy and information gain choose the best attribute in decision trees.
- ✓ Genetic algorithms evolve a population using fitness-based selection, crossover and mutation.

Review Questions

- Define machine learning and explain the components of a learning system.
- Explain rote learning, learning by analogy and inductive learning with examples.
- What is explanation-based learning? Explain with the Cup example.
- Differentiate between supervised and unsupervised learning.
- Explain the genetic algorithm with its operators, and perform one generation on a small example.

— *End of Unit 4* —

Reminder: these notes may contain errors — verify with your textbook and teacher.

bcalink.com