Machine Learning

(Course Code: 18B1WCI634 / 18B11BI611)

Inductive Classification

Mr. Sandeep Kumar Patel

Assistant Professor

Jaypee University of Information Technology

Waknaghat, Solan, HP-173234

Concept Learning Task

A concept learning task is the problem of **finding a general rule (concept)** that correctly classifies instances.

The goal is to learn a function that maps **instances → class labels**.

   i.e. learn a target concept (function) from a set of training examples.

Therefore,

Definition

Concept learning is the task of inferring a Boolean-valued function from training examples of its inputs and outputs.

Notations

- **Instance Space (X):** All possible examples.

$$X = \{x_1, x_2, \ldots, x_n\}$$

- **Target Concept (c):** Unknown function to be learned.

$$c : X \to \{0, 1\}$$

- **Training Data (D):** Labeled examples.

$$D = \{\langle x_i, c(x_i) \rangle\}$$

- **Hypothesis Space (H):** Set of possible rules.

$$h : X \to \{0, 1\}$$

Learning Objective

Goal

Find a hypothesis $h \in H$ such that:

$$h(x) \approx c(x), \quad \forall x \in X$$

Hypotheses H: Each hypothesis is a conjunction of constraints on the attributes. Constraints can be:

- **?** (any value acceptable),
- ϕ (no value acceptable), or
- A specific value

Concept Learning as a Search Problem

Concept learning can be viewed as a **search problem** where:

- The search space is the **hypothesis space** H.
- Each hypothesis represents a possible concept.
- The goal is to find a hypothesis that is consistent with training data.

Example

- **Attributes:** (Sky, Temp, Humidity, Wind, Water, Forecast)
- **Target concept (c):** "Play Tennis"
- **Training example:**

$$\langle (Sunny, Warm, Normal, Strong, Warm, Same), \ Yes \rangle$$

- **Hypothesis (h):**

$$h(x) = (Sky = Sunny) \wedge (Temp = Warm)$$

Problem:

In The given Attribute table,

Attribute	Possible Values
Sky	{Sunny, Rainy, Cloudy} $\rightarrow$ 3
Temp	{Warm, Cold} $\rightarrow$ 2
Humidity	{Normal, High} $\rightarrow$ 2
Wind	{Strong, Weak} $\rightarrow$ 2
Water	{Warm, Cool} $\rightarrow$ 2
Forecast	{Same, Change} $\rightarrow$ 2

Q. Find Instance Space, Total number of concepts, Numbers of syntactic Hypothesis and semantic Hypothesis.

Number of Concepts

Instance Space Size:

$$|X| = 3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$$

Definition of a Concept:

$$c : X \to \{0, 1\}$$

Each instance can be labeled independently as positive or negative.

Total Number of Concepts:

$$\text{Number of concepts} = 2^{|X|} = 2^{96}$$

Number of Hypotheses

- **Syntactically:** For each attribute:

$$\text{Choices} = (\text{number of values}) + 2(\phi, ?)$$

$$|H| = (3 + 2)(2 + 2)^5 == 5 \times 4^5 = 4 \times 1024$$

$$\boxed{|H| = 5120}$$

- **Semantically:** For each attribute:

$$\text{Choices} = (\text{number of values}) + 1$$

$$|H| = (3 + 1)(2 + 1)^5 = 4 \times 3^5 = 4 \times 243 = 972$$

$$\boxed{|H| = 972 + 1(h_0) = 973}$$

General-to-Specific Ordering of Hypotheses

Specific hypothesis: only matches very few positive examples.
General hypothesis: matches many positive examples.

Definition

A hypothesis h_j is **more general than or equal to** hypothesis h_k (written $h_j \geq_g h_k$) if:

$$\forall x \in X, \ (h_k(x) = 1 \implies h_j(x) = 1)$$

Example with Attributes Attributes: $(Sky, Temp, Humidity, Wind)$

$$h_1 = (?, \ Warm, \ ?, \ ?)$$
$$h_2 = (Sunny, \ Warm, \ High, \ ?)$$

So,

$$h_1 \ \geq_g \ h_2$$

Concept Learning Task: Find-S Algorithm

Training Examples:

Ex.	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Question:
Using the **Find-S Algorithm**, determine the final hypothesis *h* after processing all the above examples.

Find-S Algorithm: Step 1 & Step 2

Goal: Find the most specific hypothesis h consistent with all positive training examples.

Step 1: Initialization

- Start with the most specific hypothesis:

$$h \leftarrow \langle \varnothing, \varnothing, \ldots, \varnothing \rangle$$

Step 2: For each training example (x, y)

- If y is positive, then for each attribute i:
  - If $h[i] = \varnothing$, set $h[i] \leftarrow x[i]$.
  - If $h[i] \neq x[i]$, set $h[i] \leftarrow ?$.
  - Otherwise leave $h[i]$ unchanged.

Find-S Algorithm: Step 3 & Step 4

Step 3: Processing negative examples
- If y is negative:
    - Do nothing (hypothesis is not updated).

Step 4: Final hypothesis
- After all training examples are processed, return h.

Output: The most specific hypothesis consistent with all positive training examples.

Find-S Steps:

$$S_0 = \langle, \phi, \phi, \phi, \phi, \phi \rangle$$

$S_1 = \langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same} \rangle$ (after Example 1)

$S_2 = \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same} \rangle$ (after Example 2)

$S_3 = S_2$ (Example 3 is negative, ignored)

$S_{final} = \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ? \rangle$ (after Example 4)

Final Hypothesis:

$h(x) = \text{EnjoySport if } (Sky = Sunny) \wedge (AirTemp = Warm) \wedge (Wind = Strong)$

Concept Learning Task: Loan Approval

Ex.	Income	Credit	Collateral	Age	Employment	Location	LoanApproval
1	High	Good	Yes	Young	Salaried	Urban	Yes
2	High	Good	No	Young	Salaried	Urban	Yes
3	Low	Bad	Yes	Old	SelfEmp	Rural	No
4	High	Good	Yes	Young	Salaried	Urban	Yes

Q: Using the **Find-S Algorithm**, what is the final hypothesis h after processing all the above examples?

Find-S Solution: Loan Approval (Part 1)

Start with the most specific hypothesis:

$$S_0 = \langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle$$

Example 1 (Positive) : (High, Good, Yes, Young, Salaried, Urban)

$$S_1 = \langle High, Good, Yes, Young, Salaried, Urban \rangle$$

Example 2 (Positive) : (High, Good, No, Young, Salaried, Urban)
Collateral differs $\rightarrow$ generalize to ?:

$$S_2 = \langle High, Good, ?, Young, Salaried, Urban \rangle$$

Find-S Solution: Loan Approval (Part 2)

Example 3 (Negative): (Low, Bad, Yes, Old, SelfEmp, Rural)
Ignore negative example; hypothesis unchanged:

$$S_3 = S_2 = \langle High, Good, ?, Young, Salaried, Urban \rangle$$

Example 4 (Positive): (High, Good, Yes, Young, Salaried, Urban)
Already consistent, final hypothesis:

$$\boxed{S = \langle High, Good, ?, Young, Salaried, Urban \rangle}$$

Drawbacks of Find-S Algorithm

- **Ignores Negative Examples:** Updates hypothesis only on positive examples; negative examples are not used.

- **Single Hypothesis Output:** Produces only one final hypothesis, even if multiple consistent hypotheses exist.

- **Dependence on Initial Example:** The first positive example strongly influences the hypothesis, limiting flexibility.

Concept Learning Task: Candidate Elimination Algorithm

Training Examples:

Ex.	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Question: Using the **Candidate Elimination Algorithm**, compute the **final Version Space** $VS_{H,D}$ by updating the S (most specific) and G (most general) boundaries after each example.

Prerequisites for Candidate Elimination Algorithm

- **Hypothesis Space (H):** The set of all possible hypotheses. Each hypothesis is a conjunction of attribute constraints.
- **General-to-Specific Ordering:** A hypothesis h_j is more general than or equal to h_k if:

$$\forall x \in X, \ (h_k(x) = 1 \implies h_j(x) = 1)$$

- **Boundary Sets:**
  - S — the set of most specific hypotheses consistent with the data.
  - G — the set of most general hypotheses consistent with the data.

Prerequisites for Candidate Elimination Algorithm

- **Consistent:** A hypothesis h is *consistent* with a set of training examples D if and only if

$$\forall (x, c(x)) \in D, \ h(x) = c(x).$$

- **Version Space:** The set of all hypotheses in H that are consistent with the training data:

$$VS_{H,D} = \{ h \in H \mid \forall (x, c(x)) \in D, \ h(x) = c(x) \}.$$

Characterization using boundaries S and G:

$$VS_{H,D} = \{ h \in H \mid S \leq_g h \leq_g G \}.$$

Candidate Elimination Algorithm: Step 1 – Initialization

Goal: Learn the version space (all hypotheses consistent with the training data) by maintaining:

- **S** — the set of most specific hypotheses
- **G** — the set of most general hypotheses

Step 1: Initialization

- Start with the most specific hypothesis:

$$S = \langle \varnothing, \varnothing, \dots \rangle$$

- Start with the most general hypothesis:

$$G = \langle ?, ?, \dots \rangle$$

Candidate Elimination: Positive Example Update

For a positive example $(x, y = 1)$**:**

1. Remove from G any hypotheses inconsistent with x
2. For each $s \in S$ inconsistent with x:
   - Generalize s minimally so that $s(x) = 1$
3. Remove any hypothesis in S that is more general than another hypothesis in S

Effect: S becomes more general to include positive examples, G stays consistent

Candidate Elimination: Negative Example Update

For a negative example $(x, y = 0)$**:**

1. Remove from S any hypotheses inconsistent with x
2. For each $g \in G$ inconsistent with x:
   - Specialize g minimally so that $g(x) = 0$
3. Remove any hypothesis in G that is more specific than another hypothesis in G

Effect: G becomes more specific to exclude negative examples, S stays consistent

Candidate Elimination: Output

After processing all training examples:

$$VS_{H,D} = \{h \mid S \leq_g h \leq_g G\}$$

- All hypotheses in the version space are consistent with all training examples
- $S =$ most specific hypotheses
- $G =$ most general hypotheses
- The version space represents all possible solutions

Candidate Elimination Algorithm: Step-by-Step

Initialization:

- $S_0 = \{\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle\}$
- $G_0 = \{\langle ?, ?, ?, ?, ?, ? \rangle\}$

Example 1 (Positive): (Sunny, Warm, Normal, Strong, Warm, Same)

- $S_1 = \{\langle Sunny, Warm, Normal, Strong, Warm, Same \rangle\}$
- $G_1 = \{\langle ?, ?, ?, ?, ?, ? \rangle\}$

Candidate Elimination Algorithm

Example 2 (Positive): (Sunny, Warm, High, Strong, Warm, Same)

- Generalize S_1 on Humidity $\rightarrow$?
- $S_2 = \{\langle Sunny, Warm, ?, Strong, Warm, Same \rangle\}$
- $G_2 = \{\langle ?, ?, ?, ?, ?, ? \rangle\}$

Example 3 (Negative): (Rainy, Cold, High, Strong, Warm, Change)

- Specialize G to exclude negative while $\geq S$
- $G_3 = \{\langle Sunny, ?, ?, ?, ?, ? \rangle, \langle ?, Warm, ?, ?, ?, ? \rangle, \langle ?, ?, ?, ?, ?, Same \rangle\}$
- $S_3 = S_2$

Candidate Elimination Algorithm

Example 4 (Positive): (Sunny, Warm, High, Strong, Cool, Change)

- Remove inconsistent hypotheses from G_3
- $G_4 = \{\langle Sunny, ?, ?, ?, ?, ? \rangle, \langle ?, Warm, ?, ?, ?, ? \rangle\}$
- Generalize S_3 to cover Water and Forecast differences
- $S_4 = \{\langle Sunny, Warm, ?, Strong, ?, ? \rangle\}$

Final Version Space

Most Specific Boundary S:

$$S = \{\langle Sunny, Warm, ?, Strong, ?, ?\rangle\}$$

Most General Boundary G:

$$G = \{\langle Sunny, ?, ?, ?, ?, ?\rangle, \langle ?, Warm, ?, ?, ?, ?\rangle\}$$

Final Version Space:

$$VS_{H,D} = \{h \in H \mid S \leq_g h \leq_g G\}$$

$$G = \{\langle Sunny, Warm, ?, ?, ?, ?\rangle, \langle Sunny, ?, ?, Strong, ?, ?\rangle \langle ?, Warm, ?, Strong, ?, ?\rangle\}$$

Concept Learning Task: Car Buying Decision

Training Examples:

Ex.	Make	Doors	Engine	Color	Manual	Buy?
1	Toyota	4	Petrol	Red	Yes	Yes
2	Honda	2	Petrol	Blue	Yes	Yes
3	Ford	4	Diesel	Red	No	No
4	Toyota	2	Petrol	Red	Yes	Yes

Q. Use the **Candidate Elimination Algorithm** to update the S and G boundaries after each example.

Step 0: Initialization

- Most specific hypothesis: $S_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$
- Most general hypothesis: $G_0 = \langle ?, ?, ?, ?, ? \rangle$

Step 1: Example 1 (Positive)

- Example 1: $\langle$Toyota, 4, Petrol, Red, Yes, Yes$\rangle$
- Update S to first positive example:

$$S_1 = \langle\text{Toyota, 4, Petrol, Red, Yes}\rangle$$

- G remains maximally general:

$$G_1 = \langle ?, ?, ?, ?, ? \rangle$$

Step 2: Example 2 (Positive)

- Example 2: $\langle \text{Honda}, 2, \text{Petrol}, \text{Blue}, \text{Yes}, \text{Yes} \rangle$
- Generalize S where attributes differ:

$$S_2 = \langle ?, ?, \text{Petrol}, ?, \text{Yes} \rangle$$

- G unchanged:

$$G_2 = \langle ?, ?, ?, ?, ? \rangle$$

Step 3: Example 3 (Negative)

- Example 3: $\langle \text{Ford}, 4, \text{Diesel}, \text{Red}, \text{No}, \text{No} \rangle$
- Specialize G to exclude negative example while including S:

$$G_3 = \langle ?, ?, \text{Petrol}, ?, ? \rangle$$

- S remains unchanged:

$$S_3 = \langle ?, ?, \text{Petrol}, ?, \text{Yes} \rangle$$

Step 4: Example 4 (Positive)

- Example 4: $\langle$Toyota, 2, Petrol, Red, Yes, Yes$\rangle$
- Update S to include this positive example (already covered by S3):

$$S_4 = \langle ?, ?, \text{Petrol}, ?, \text{Yes} \rangle$$

- G unchanged:

$$G_4 = \langle ?, ?, \text{Petrol}, ?, ? \rangle$$

Final Version Space

- Most Specific Hypothesis:

$$S = \langle ?, ?, \text{Petrol}, ?, \text{Yes} \rangle$$

- Most General Hypothesis:

$$G = \langle ?, ?, \text{Petrol}, ?, ? \rangle$$

- Version space narrowed down to **Engine = Petrol** and **Manual = Yes**.

Concept Learning Task: Fruit Edibility Decision

Training Examples:

Ex.	Color	Size	Shape	Taste	Sweet	Edible
1	Red	Small	Round	Sour	Yes	Yes
2	Yellow	Small	Round	Sweet	Yes	Yes
3	Green	Large	Oval	Bitter	No	No
4	Red	Small	Oval	Sweet	Yes	Yes

Use the Candidate Elimination Algorithm to update the S and G boundaries after each example.

Step 1: Example 1 (Positive)

- **Initialization:**
  Most specific hypothesis: $S_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$ Most general hypothesis:
  $G_0 = \langle ?, ?, ?, ?, ? \rangle$

- Example 1: $\langle$ Red, Small, Round, Sour, Yes, Yes $\rangle$

- Update S to first positive example:

$$S_1 = \langle \text{Red, Small, Round, Sour, Yes} \rangle$$

- G remains maximally general:

$$G_1 = \langle ?, ?, ?, ?, ? \rangle$$

Step 2: Example 2 (Positive)

- Example 2: ⟨Yellow, Small, Round, Sweet, Yes, Yes⟩
- Generalize S where attributes differ:

$$S_2 = \langle ?, \text{Small}, \text{Round}, ?, \text{Yes} \rangle$$

- G unchanged:

$$G_2 = \langle ?, ?, ?, ?, ? \rangle$$

Step 3: Example 3 (Negative)

- Example 3: $\langle$Green, Large, Oval, Bitter, No, No$\rangle$
- Specialize G to exclude negative example while including S:

$$G_3 = \langle ?, \text{Small}, ?, ?, \text{Yes} \rangle$$

- S remains unchanged:

$$S_3 = \langle ?, \text{Small}, \text{Round}, ?, \text{Yes} \rangle$$

Step 4: Example 4 (Positive)

- Example 4: $\langle$Red, Small, Oval, Sweet, Yes, Yes$\rangle$
- Update S to include this positive example (already partially generalized):

$$S_4 = \langle ?, \text{Small}, ?, ?, \text{Yes}\rangle$$

- G remains:

$$G_4 = \langle ?, \text{Small}, ?, ?, \text{Yes}\rangle$$

Final Version Space

- Most Specific Hypothesis:

$$S = \langle ?, \text{Small}, ?, ?, \text{Yes} \rangle$$

- Most General Hypothesis:

$$G = \langle ?, \text{Small}, ?, ?, \text{Yes} \rangle$$

- Version space narrowed down to: **Size = Small** and **Sweet = Yes**.

Hypothesis Space and Convergence

- Candidate-Elimination converges to the true target concept if:
    - Training examples are noise-free
    - Target concept is contained in hypothesis space H
- If the target concept is not representable in H:
    - Version space may become empty
    - Learning fails despite correct data

Why Learning is Difficult

- Training data is limited
- Infinitely many functions can fit the same data
- Model must predict unseen inputs

Question: How does the model choose one function?

Answer: Inductive Bias

What is Inductive Bias?

Definition:

Inductive Bias

Inductive bias is the set of assumptions a learning algorithm makes to generalize from training data to unseen data.

- Prior belief of the model
- Guides generalization

Core Idea

- Same training data
- Different learning algorithms
- Different predictions

Reason: Different Inductive Biases

Example Dataset

Training Data:

x	y
1	2
2	4
3	6

Question: What is y when $x = 4$?

Model 1: Linear Regression

Inductive Bias:

- Assumes a linear relationship

Model:

$$y = wx + b$$

Learned Function:

$$y = 2x$$

Prediction:

$$y(4) = 8$$

Model 2: High-Degree Polynomial

Inductive Bias:

- Allows complex functions
- Can perfectly fit training data

Prediction on unseen data:

$$y(4) = 100 \quad \text{(or any value)}$$

Result:

- Overfitting
- Poor generalization

What Did We Observe?

- Both models fit training data
- Predictions differ on unseen data
- Difference comes from assumptions

Assumptions = Inductive Bias

Why is Inductive Bias Important?

1. Enables generalization
2. Reduces data requirement
3. Prevents overfitting
4. Improves learning efficiency

Inductive Bias in Common Models

Model	Inductive Bias
Linear Regression	Linearity
k-NN	Similar inputs have similar outputs
Decision Tree	Rule-based splits
CNN	Locality and translation invariance
RNN	Sequential dependency

No Free Lunch Theorem: Definition

Definition:
The *No Free Lunch (NFL) Theorem* states that:

No single learning or optimization algorithm performs better than all others when averaged over all possible problems.

- If an algorithm performs well on some problems,
- it must perform poorly on others.
- There is **no universally best algorithm**.

Intuition and Example

Intuition:

- Different problems have different structures
- Algorithms rely on different assumptions
- Performance depends on how well assumptions match the problem

Example:

- CNNs perform well on image data
- RNNs perform well on sequential data
- Random Forests perform well on tabular data

Conclusion: One algorithm cannot be optimal for all problem types.

Mathematical Statement of NFL

Let:

- f denote all possible target functions
- A_1, A_2 denote two learning algorithms
- L denote a loss function

The No Free Lunch Theorem states:

$$\sum_f \mathbb{E}[L(A_1, f)] = \sum_f \mathbb{E}[L(A_2, f)]$$

Interpretation:

- Averaged over all possible problems,
- all algorithms have identical expected performance.

No Free Lunch Theorem: Summary

- No algorithm works best for all problems
- Performance depends on inductive bias
- Choosing a model = choosing a bias