1. Explain the concept of Version Spaces with a suitable example. Draw the version space for a simple concept learning problem.

Solution

Concept

A Version Space is a set of all possible hypotheses that are consistent with the given training examples. It is bounded by two sets:

- General Boundary (G): Most general hypotheses that fit the data.
- Specific Boundary (S): Most specific hypotheses that fit the data.

Example: Learning "Play Tennis"

Attributes:

- Outlook (Sunny, Rainy, Overcast)
- Humidity (High, Normal)
- Wind (Strong, Weak)

Training Data:

Outlook	Humidity	Wind	Play Tennis?
Sunny	High	Weak	No (-)
Rainy	High	Strong	No (-)
Overcast	Normal	Weak	Yes (+)

Version Space Construction

1. **Initialize G and S:**
   - G: Most general hypothesis (<?, ?, ?> meaning any combination is possible).
   - S: Most specific hypothesis (<Ø, Ø, Ø> meaning no instance is covered).
2. **After 1st Example (Sunny, High, Weak → No):**
   - G: Exclude hypotheses that cover this negative example.
     - <?, ?, ?> is too general (covers a negative example), so we specialize it.
     - New G:
       - <Sunny, ?, ?> (if Outlook=Sunny → No)
       - <?, High, ?> (if Humidity=High → No)

        - <?, ?, Weak> (if Wind=Weak → No)
    - S: Still <Ø, Ø, Ø> (no positive example seen yet).
3. **After 2nd Example (Rainy, High, Strong → No):**
    - G: Further refine to exclude this negative example.
        - New G:
            - <Sunny or Rainy, High, ?>
            - <?, High, Weak or Strong>
    - S: Still <Ø, Ø, Ø>.
4. **After 3rd Example (Overcast, Normal, Weak → Yes):**
    - S: Now updates to <Overcast, Normal, Weak>.
    - G: Remains as before (since it must still exclude negatives).

Final Version Space

- G: { <?, Normal, ?>, <Overcast, ?, ?> }
- S: { <Overcast, Normal, Weak> }

Diagram:

G: { (Overcast, ?, ?), (?, Normal, ?) }

↑

All hypotheses in between

↓

S: { (Overcast, Normal, Weak) }

---

2. What is inductive bias? Explain its importance in machine learning algorithms.

Solution

Definition

Inductive bias refers to the assumptions a learning algorithm makes to generalize from limited training data to unseen examples.

Why is it Important?

- Without bias, a model cannot prefer one hypothesis over another (No Free Lunch Theorem).
- Helps in efficient learning by narrowing down the hypothesis space.

Examples of Inductive Bias

Algorithm	Inductive Bias
Decision Trees	Prefer shorter trees (Occam's Razor).
K-Nearest Neighbors	Nearby points belong to the same class.
Linear Regression	Assumes a linear relationship between features.
Neural Networks	Smooth interpolation between data points.

3. Draw and explain the basic structure of a decision tree for classifying whether a customer will buy a product based on age, income, and student status.

Solution

Problem Statement

Predict whether a customer will buy a product based on:

- Age (Young, Middle, Old)
- Income (High, Medium, Low)
- Student (Yes, No)

Decision Tree Construction

1. Choose Root Node (Attribute with highest Information Gain).
    - Suppose Income provides the best split.
2. Split Data:
    - If Income = High → Buys (Yes)
    - If Income = Medium → Further split by Age
        - If Age = Young → Buys (Yes)
        - If Age = Old → Buys (No)
    - If Income = Low → Further split by Student
        - If Student = Yes → Buys (No)
        - If Student = No → Buys (Yes)

Final Decision Tree

```
[Income?]
├── High → Yes
├── Medium → [Age?]
│   ├── Young → Yes
│   └── Old → No
└── Low → [Student?]
    ├── Yes → No
    └── No → Yes
```

4. List and explain the different types of learning problems in machine learning.

Solution

1. Supervised Learning

- Input: Labeled data (features + target).
- Goal: Learn a mapping from inputs to outputs.
- Examples:
  - Classification (Spam Detection).
  - Regression (House Price Prediction).

2. Unsupervised Learning

- Input: Unlabeled data.
- Goal: Discover hidden patterns.
- Examples:
  - Clustering (Customer Segmentation).
  - Dimensionality Reduction (PCA).

3. Semi-Supervised Learning

- Input: Mix of labeled + unlabeled data.
- Goal: Improve learning using unlabeled data.
- Example: Google Photos (few labeled images, many unlabeled).

4. Reinforcement Learning (RL)

- Input: Agent interacts with environment, gets rewards.
- Goal: Learn a policy to maximize rewards.
- Example: AlphaGo (plays Go by trial and error).

5. Online Learning

- Input: Data arrives sequentially.
- Goal: Update model incrementally.
- Example: Stock price prediction in real-time.

5. Assume S is a collection containing 14 examples, [9+, 5–]. Of these 14 examples, suppose 6 of the positive and 2 of the negative examples have Wind = Weak, and the remainder have Wind = Strong. What will be the information gain on attribute wind ?

Solution

Given Data:

- Total examples: **14** (9+, 5–).
- **Wind = Weak**: 6+, 2– → 8 examples.
- **Wind = Strong**: 3+, 3– → 6 examples.

Step 1: Compute Entropy of Original Data (S)

$$Ent(S) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) = 0.940 \text{ bits.}$$

Step 2: Compute Entropy After Splitting by Wind

1. **For Wind = Weak:**

$$Ent(S_{Weak}) = -\left(\frac{6}{8}\log_2\frac{6}{8} + \frac{2}{8}\log_2\frac{2}{8}\right) = 0.811 \text{ bits.}$$

2. **For Wind = Strong:**

$$Ent(S_{Strong}) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1.0 \text{ bit.}$$

Step 3: Compute Weighted Entropy

$$Weighted\ Entropy = \left(\frac{8}{14} \times 0.811\right) + \left(\frac{6}{14} \times 1.0\right) = 0.892 \text{ bits.}$$

Step 4: Compute Information Gain

$$Gain(S, Wind) = Ent(S) - Weighted\ Entropy = 0.940 - 0.892 = 0.048 \text{ bits.}$$

Conclusion

The **Information Gain for "Wind" is 0.048 bits**, meaning it provides a small improvement in classification.