VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

MACHINE LEARNING (CO3117)

Homework 2:

Decision Tree

Team LHPD2

Semester 2, Academic Year 2024 - 2025

Teacher: Nguyen An Khuong
Students: Nguyen Quang Phu - 2252621 (***Leader***)
Nguyen Thanh Dat - 2252145 (Member)
Pham Huynh Bao Dai - 2252139 (Member)
Nguyen Tien Hung - 2252280 (Member)
Nguyen Thien Loc - 2252460 (Member)

HO CHI MINH CITY, FEBRUARY 2025

Contents

Chapter 1

Introduction

1.1 Purpose of This Document

This chapter serves as an introduction to our team, LHPD2, for the Machine Learning (CO3117) course in Semester 2, Academic Year 2024 - 2025. The purpose of this writing is to formally present our team members and confirm our participation in solving the given homework exercise, "Homework 2: Decision Tree".

1.2 Workload

Specifically, our contributions included:

- **Understanding the Problem:** Each member participated in analyzing and discussing the given exercise to ensure a shared understanding of the requirements.
- **Exploring Concepts:** We collectively researched and reviewed relevant concepts, theories, and methods necessary for solving the problems.
- **Reasoning for the Solution:** The team worked together to identify the rationale behind each approach, ensuring that all steps were logical and well-justified.
- **Implementing the Solution:** The solutions were implemented with equal collaboration, where every member contributed to coding, calculations, and documentation.
- **Final Review:** The team jointly reviewed the solutions to verify their correctness, clarity, and coherence before submission.

We believe that this exercise has enhanced our understanding of the concepts involved and strengthened our ability to work collaboratively as a team.

Chapter 2

Problem Description - Our Solution

ID3 searches for just one consistent hypothesis, whereas the **Candidate-Elimination** algorithm finds all consistent hypotheses. Consider the correspondence between these two learning algorithms.

2.1 Question a

2.1.1 Description

Show the decision tree that would be learned by **ID3** assuming it is given the four training examples for the *EnjoySport?* target concept shown in Table 2.1 of Chapter 2.

Example	*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

TABLE 2.1
Positive and negative training examples for the target concept *EnjoySport*.

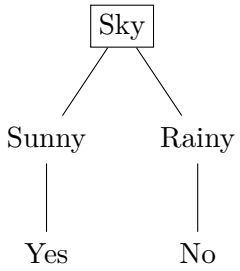
Figure 2.1

2.1.2 Solution

Based on the four training examples (Sky, AirTemp, Humidity, Wind, Water, Forecast, EnjoySport), we observe that the attribute `Sky` separates the data perfectly:

- All three positive (Yes) examples have `Sky = Sunny`.
- The single negative (No) example has `Sky = Rainy`.

Therefore, the decision tree generated by ID3 is quite simple:



Since `Sky = Sunny` covers all positive examples and `Sky = Rainy` covers the only negative example, no additional branches are needed.

2.2 Question b

2.2.1 Description

What is the relationship between the learned decision tree and the version space (shown in Figure 2.3 of Chapter 2) that is learned from these same examples? Is the learned tree equivalent to one of the members of the version space?

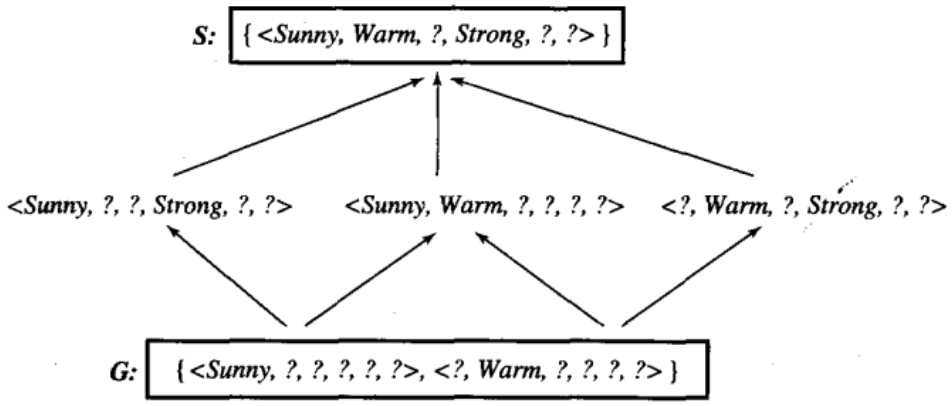


FIGURE 2.3
A version space with its general and specific boundary sets. The version space includes all six hypotheses shown here, but can be represented more simply by S and G. Arrows indicate instances of the *more_general_than* relation. This is the version space for the *EnjoySport* concept learning problem and training examples described in Table 2.1.

Figure 2.2: Version Space Representation

2.2.2 Solution

Step 1: Understanding the Version Space

The **version space** is the set of all hypotheses consistent with the training examples. The Candidate-Elimination algorithm maintains two boundary sets:

- **S**: The most specific hypothesis consistent with the training data.
- **G**: The most general hypothesis consistent with the training data.

The version space consists of all hypotheses lying between these two boundary sets.

Step 2: Understanding the Decision Tree Learned by ID3

The ID3 algorithm constructs a single decision tree by recursively selecting the attribute with the highest **information gain** at each step. Unlike Candidate-Elimination, which maintains

a version space of all consistent hypotheses, ID3:

- Maintains **only one hypothesis** instead of multiple valid hypotheses.
- Uses a greedy approach without backtracking, meaning that once an attribute is chosen, it cannot be changed.
- Uses statistical properties from all examples to generalize efficiently **but may not always select the optimal hypothesis**.

Step 3: Comparing the Decision Tree to the Version Space

(i) Is the Decision Tree in the Version Space?

Yes, the decision tree learned by ID3 belongs to the version space because:

- It is consistent with all training examples.
- The version space contains **all** consistent hypotheses, including the one chosen by ID3.

(ii) How Does ID3 Differ from Candidate-Elimination?

While ID3 produces a single tree, Candidate-Elimination maintains a **set** of hypotheses. This leads to key differences:

- **Efficiency:** ID3 is computationally more efficient since it does not track multiple hypotheses.
- **Flexibility:** Candidate-Elimination can provide insights into alternative consistent hypotheses, while ID3 commits to one hypothesis.
- **Handling Noise:** ID3 can be modified (e.g., pruning) to handle noisy data, while Candidate-Elimination is not robust to noise.

Step 4: Why the Learned Tree May Not Be the Most General or Specific Hypothesis

The decision tree produced by ID3 falls within the version space but is not necessarily:

- The **most general hypothesis (G)**, since it makes decisions at each node that restrict generality.
- The **most specific hypothesis (S)**, since it generalizes beyond exact matches to training examples.

Instead, ID3 selects a hypothesis that is **somewhere in between**.

Step 5: Relationship Between Decision Trees and Version Space Hypothesis Representation

While the version space represents hypotheses as conjunctive attribute constraints, a decision tree provides a richer structure for expressing hypotheses. The decision tree captures dependencies between attributes that a conjunctive hypothesis representation in version space may not express.

If the target function is not represented within the hypothesis space (since conjunctions alone are not always a complete basis), the version space may be empty. In this example, the learned decision tree equivalent to **Sky = Sunny** corresponds to the hypothesis $< \textit{Sunny}, ?, ?, ?, ?, ? >$ from the **G** boundary set.

Conclusion

The decision tree learned by ID3 is **equivalent to one of the members of the version space**. It is consistent with the training examples and falls within the bounds defined by the **S** and **G** sets in the Candidate-Elimination algorithm. However, it may not be the most specific or most general hypothesis in the version space. Furthermore, the decision tree has a richer representational power compared to the version space, which only includes conjunctive attribute constraints.

2.3 Question c

2.3.1 Description

Add the following training example, and compute the **new decision tree**. This time, show the **value of the information gain** for **each candidate attribute** at each step in growing the tree.

Sky	Air-Temp	Humidity	Wind	Water	Forecast	Enjoy-Sport?
Sunny	Warm	Normal	Weak	Warm	Same	No

2.3.2 Solution

In this Exercises, we use these 2 formulas

$$Entropy(S) = -p_{(+)}log_2p_{(+)} - p_{(-)}log_2p_{(-)}$$

$$Gain(S, A) = \text{Entropy}(S) - \sum_{V \in \text{Value}(a)} \frac{|S_v|}{S} \text{Entropy}(S_v)$$

(1) First test:

We calculate the Entropy of S:

$$Entropy(S) = -3/5log_23/5 - 2/5log_22/5 \approx 0.971$$

And then calculate the IG:

$$\begin{aligned}
Gain(S, Sky) &= 0.971 - \frac{4}{5}(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}) \approx 0.322 \\
Gain(S, AirTemp) &= 0.971 - \frac{2}{4}(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}) \approx 0.322 \\
Gain(S, Humidity) &= 0.971 - \frac{3}{5}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) - \frac{2}{5} \approx 0.02 \\
Gain(S, Wind) &= 0.971 - \frac{4}{5}(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}) \approx 0.322 \\
Gain(S, Water) &= 0.971 - \frac{4}{5}(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}) \approx 0.171 \\
Gain(S, Forcast) &= 0.971 - \frac{3}{5}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) - \frac{2}{5} \approx 0.02
\end{aligned}$$

Based on the IG values, we can choose attributes like Sky, AirTemp, and Wind as potential roots for the tree. In this situation, we choose AirTemp as the root.

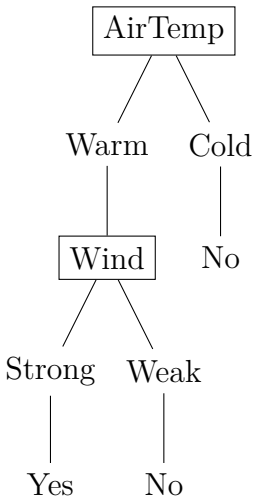
(2) Second test:

$$Entropy(S) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} \approx 0.811$$

And then calculate the IG:

$$\begin{aligned}
Gain(S, Sky) &= 0.811 - (-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}) = 0 \\
Gain(S, Humidity) &= 0.811 - \frac{2}{4} \approx 0.311 \\
Gain(S, Wind) &= 0.811 - 0 \approx 0.811 \\
Gain(S, Water) &= 0.811 - \frac{3}{4}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) \approx 0.122 \\
Gain(S, Forcast) &= 0.811 - \frac{3}{4}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) \approx 0.122
\end{aligned}$$

So after the second test, we choose Wind



2.4 Question d

2.4.1 Description

Suppose we wish to design a learner that (like **ID3**) searches a **space of decision tree hypotheses** and (like **Candidate-Elimination**) finds all hypotheses consistent with the data. In short, we wish to apply the **Candidate-Elimination** algorithm to searching the space of decision tree hypotheses. Show the **S** and **G** sets that result from the first training example from Table 2.1. Note **S** must contain the most specific decision trees consistent with the data, whereas **G** must contain the most general. Show how the **S** and **G** sets are refined by the second training example (you may omit syntactically distinct trees that describe the same concept). What difficulties do you foresee in applying **Candidate-Elimination** to a decision tree hypothesis space?

Example	*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

TABLE 2.1
Positive and negative training examples for the target concept *EnjoySport*.

Figure 2.3

2.4.2 Solution

2.4.2.1 Initialization

S-set (Specific Hypothesis):

$$S_0 = \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$$

if match --> positive classification
if not match --> negative classification

G-set (General Hypothesis):

we check the generic or general boundary first

$$G_0 = \{?, ?, ?, ?, ?, ?\}$$

if match --> positive classification
if not match --> negative classification

2.4.2.2 Processing the First Training Example

Example 1: (Sunny, Warm, Normal, Strong, Warm, Same) $\rightarrow$ Yes

When seeing this sample, check the S0, it is inconsistent so we make a new hypothesis

Since this is a positive example, the S-set is initialized as follows:

$$S_1 = \{\text{Sunny, Warm, Normal, Strong, Warm, Same}\}$$

The G-set remains unchanged:

$$G_1 = \{?, ?, ?, ?, ?, ?\}$$

2.4.2.3 Processing the Second Training Example

Example 2: (Sunny, Warm, High, Strong, Warm, Same) → Yes

Comparing with S_1, the only differing attribute is Humidity (Normal → High). To generalize, we replace it with '?': as S must be specific, so when we have the difference, create a new hypothesis

$$S_2 = \{\text{Sunny, Warm, ?, Strong, Warm, Same}\}$$

The G-set remains unchanged:

$$G_2 = \{?, ?, ?, ?, ?, ?\}$$

2.4.2.4 Challenges of Candidate Elimination for Decision Trees

- **Non-Linear Hypothesis Space:** Decision trees do not have a clear ordering from most specific to most general.
- **Branching Complexity:** One difficulty in applying candidate-elimination to a decision tree hypothesis space is that the number of hypotheses can grow exponentially as the size of the search space increases. This can make the algorithm computationally expensive and time-consuming, especially for large datasets.
- **Overfitting Issues:** Small changes in examples can force significant modifications in S and G.