Team ML_ShibaInu

Group Members:

1. 2211649 - Trần Đình Đăng Khoa
2. 2252938 - Phan Chí Vỹ
3. 2252734 - Nguyễn Đức Tâm
4. 2252338 - Huỳnh Kiệt Khải
5. 2252289 - Hà Tuấn Khang

Homework 2: Decision Tree

ID3 Decision Tree for EnjoySport

1. Problem (a)

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`

*Training Examples for `EnjoySport` Decision Tree

ANSWER:

The target attribute `EnjoySport` has two values: `Yes` and `No`

- Yes: 3 occurrences (Examples 1, 2, 4)
- No: 1 occurrence (Example 3)

Using the entropy formula:

$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

$$H(S) = -\left(\frac{3}{4}\log_2 \frac{3}{4} + \frac{1}{4}\log_2 \frac{1}{4}\right)$$

$$H(S) \approx 0.811$$

1.1. Information Gain for `Sky`

- `Sunny` $\rightarrow$ {Yes, Yes, Yes} (Entropy = 0)
- `Rainy` $\rightarrow$ {No} (Entropy = 0)

$$IG(S, \text{Sky}) = 0.811 - (\frac{3}{4} \times 0 + \frac{1}{4} \times 0) = 0.811$$

1.2. Information Gain for `AirTemp`

- `Warm` $\rightarrow$ {Yes, Yes, Yes} (Entropy = 0)
- `Cold` $\rightarrow$ {No} (Entropy = 0)

$$IG(S, \text{AirTemp}) = 0.811 - (\frac{3}{4} \times 0 + \frac{1}{4} \times 0) = 0.811$$

1.3. Information Gain for `Humidity`

- `Normal` $\rightarrow$ {Yes} (Entropy = 0)
- `High` $\rightarrow$ {Yes, No, Yes}

Entropy for High subset:

$$H(High) = -\left(\frac{2}{3}\log_2 \frac{2}{3} + \frac{1}{3}\log_2 \frac{1}{3}\right) \approx 0.918$$

$$IG(S, \text{Humidity}) = 0.811 - \left(\frac{1}{4} \times 0 + \frac{3}{4} \times 0.918\right) = 0.122$$

1.4. Information Gain for `Wind`

- `Strong` $\rightarrow$ {Yes, Yes, No, Yes} (Entropy = 0.811)

$$IG(S, \text{Wind}) = 0.811 - 0.811 = 0$$

1.5. Information Gain for `Water`

- `Warm` $\rightarrow$ {Yes, Yes, No}
- `Cool` $\rightarrow$ {Yes} (Entropy $= 0$)

Entropy for Warm subset:

$$H(Warm) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

$$IG(S, \text{Water}) = 0.811 - \left(\frac{3}{4} \times 0.918 + \frac{1}{4} \times 0\right) = 0.122$$

1.6. Information Gain for `Forecast`

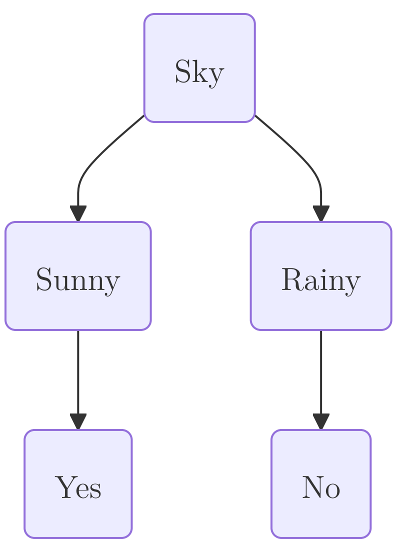
- `Same` $\rightarrow$ {Yes, Yes} (Entropy $= 0$)
- `Change` $\rightarrow$ {No, Yes}

Entropy for Change subset:

$$H(Change) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.0$$

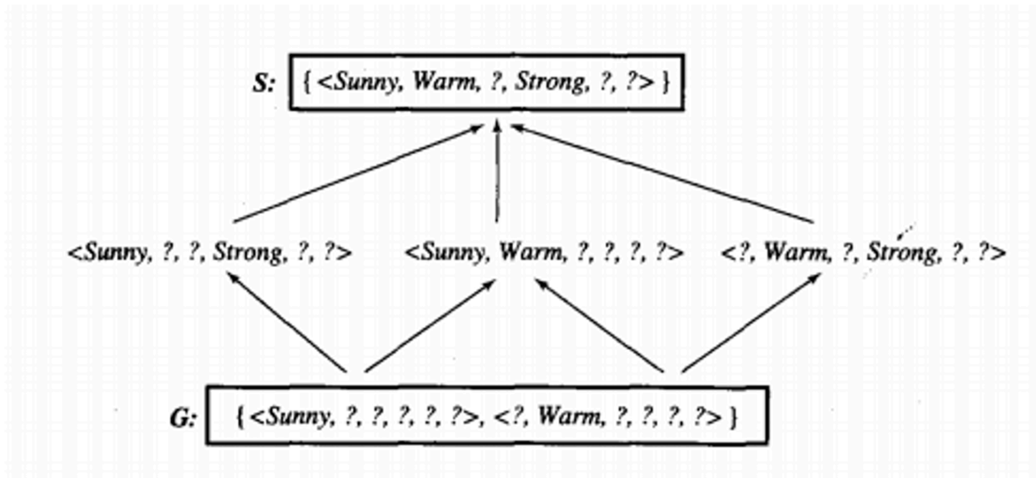
$$IG(S, \text{Forecast}) = 0.811 - \left(\frac{2}{4} \times 0 + \frac{2}{4} \times 1\right) = 0.311$$

Since `Sky` and `AirTemp` has the highest information gain, we choose the first in order which is `Sky` as the root node.



2. Problem (b)

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?



ANSWER:

Key Observations:

- The **decision tree** learned by ID3 is one of the **hypotheses within the version space**.
- The version space contains **multiple possible hypotheses**, while ID3 selects **only one**.
- The decision tree is **not necessarily the most specific** or **most general hypothesis**.
- If multiple hypotheses remain in the version space, the decision tree is **one of the possible valid hypotheses**.

Thus, the learned decision tree is a member of the version space, but it may not be the only hypothesis that fits the data.

3. Problem (c)

Adding a New Training Example and Computing the New Decision Tree.

ANSWER:

We now add the following new training example to our dataset:

`Sky`	`Air-Temp`	`Humidity`	`Wind`	`Water`	`Forecast`	`EnjoySport`
`Sunny`	`Warm`	`Normal`	`Weak`	`Warm`	`Same`	`No`

Example	`Sky`	`AirTemp`	`Humidity`	`Wind`	`Water`	`Forecast`	`EnjoySport`
`1`	`Sunny`	`Warm`	`Normal`	`Strong`	`Warm`	`Same`	`Yes`
`2`	`Sunny`	`Warm`	`High`	`Strong`	`Warm`	`Same`	`Yes`

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes
5	Sunny	Warm	Normal	Weak	Warm	Same	No

*Training Examples for `EnjoySport` Decision Tree

3.1. First Splitting

The entropy of the dataset before the split:

$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Training examples have 3 `Yes` and 2 No for `EnjoySport`, we compute:

$$H(S) = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right)$$

$$H(S) = 0.971$$

Compute Information Gain for Each Attribute

Sky:

- Sunny: 3 Yes, 1 No $\rightarrow$

$$H(S_1) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) \approx 0.811$$

- Rainy: 1 No $\rightarrow$

$$H(S_2) = 0$$

$$IG(Sky) = 0.971 - \left(\frac{4}{5} \times 0.811 + \frac{1}{5} \times 0\right) \approx 0.322$$

Air-Temp:

- Warm: 3 Yes, 1 No $\rightarrow$

$$H(S_1) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) \approx 0.811$$

- Cold: 1 No $\rightarrow$

$$H(S_2) = 0$$

$$IG(AirTemp) = 0.971 - \left(\frac{4}{5} \times 0.811 + \frac{1}{5} \times 0\right) \approx 0.322$$

Humidity:

- High: 2 Yes, 1 No $\rightarrow$

$$H(S_1) = -\left(\frac{2}{3}\log_2 \frac{2}{3} + \frac{1}{3}\log_2 \frac{1}{3}\right) \approx 0.918$$

- Normal: 1 Yes, 1 No $\rightarrow$

$$H(S_2) = -\left(\frac{1}{2}\log_2 \frac{1}{2} + \frac{1}{2}\log_2 \frac{1}{2}\right) = 1.0$$

$$IG(Humidity) = 0.971 - \left(\frac{2}{5} \times 1 + \frac{3}{5} \times 0.918\right) = 0.0202$$

Wind:

- Weak: 1 No $\rightarrow$

$$H(S_1) = 0$$

- Strong: 3 Yes, 1 No $\rightarrow$

$$H(S_2) = -\left(\frac{3}{4}\log_2 \frac{3}{4} + \frac{1}{4}\log_2 \frac{1}{4}\right) \approx 0.811$$

$$IG(Wind) = 0.971 - \left(\frac{4}{5} \times 0.811 + \frac{1}{5} \times 0\right) = 0.322$$

Water:

- Warm: 2 Yes, 2 No $\rightarrow$

$$H(S_1) = -\left(\frac{2}{4}\log_2 \frac{2}{4} + \frac{2}{4}\log_2 \frac{2}{4}\right) = 1$$

- Cool: 1 Yes $\rightarrow$

$$H(S_2) = 0$$

$$IG(Water) = 0.971 - \left(\frac{4}{5} \times 1 + \frac{1}{5} \times 0\right) = 0.171$$

Forecast:

- Same: 2 Yes, 1 No $\rightarrow$

$$H(S_1) = -\left(\frac{2}{3}\log_2 \frac{2}{3} + \frac{1}{3}\log_2 \frac{1}{3}\right) \approx 0.918$$

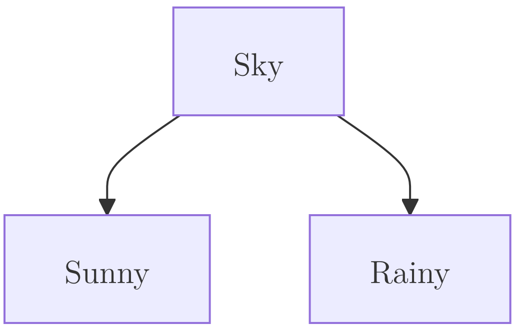
- Change: 1 Yes, 1 No $\rightarrow$

$$H(S_2) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.0$$

$$IG(Forecast) = 0.971 - \left(\frac{3}{5} \times 0.918 + \frac{2}{5} \times 1\right) = 0.0202$$

Selecting the Best Attribute for Splitting:

Since `Sky` is the first attribute with highest information gain $IG(Sky) = 0.322$, we select `Sky` as the root node.



3.2. Second Splitting

3.2.1. Sky: Sunny

The entropy of the dataset before the split:
Training examples have 3 `Yes` and 1 `No` for `EnjoySport`?, we compute:

$$H(S) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right)$$

$$H(S) \approx 0.811$$

Compute Information Gain for Each Attribute
Air-Temp:

- Warm: 3 Yes, 1 No $\rightarrow$

$$H(S_1) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) \approx 0.811$$

$$IG(AirTemp) = 0.811 - 0.811 = 0$$

Humidity:

- High: 2 Yes →

$$H(S_2) = 0$$

- Normal: 1 Yes, 1 No →

$$H(S_2) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.0$$

$$IG(Humidity) = 0.811 - \left(\frac{2}{4} \times 1 + \frac{2}{4} \times 0\right) = 0.311$$

Wind:

- Weak: 1 No →

$$H(S_1) = 0$$

- Strong: 3 Yes →

$$H(S_2) = 0$$

$$IG(Wind) = 0.811$$

Water:

- Warm: 2 Yes, 1 No →

$$H(S_1) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

- Cool: 1 Yes →

$$H(S_2) = 0$$

$$IG(Water) = 0.811 - \left(\frac{3}{4} \times 0.918 + \frac{1}{4} \times 0\right) = 0.1225$$

Forecast:

- Same: 2 Yes, 1 No →

$$H(S_1) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

- Change: 1 Yes →

$$H(S_2) = 0$$

$$IG(Forecast) = 0.811 - \left(\frac{3}{4} \times 0.918 + \frac{1}{4} \times 0\right) = 0.1225$$

Selecting the Best Attribute for Splitting

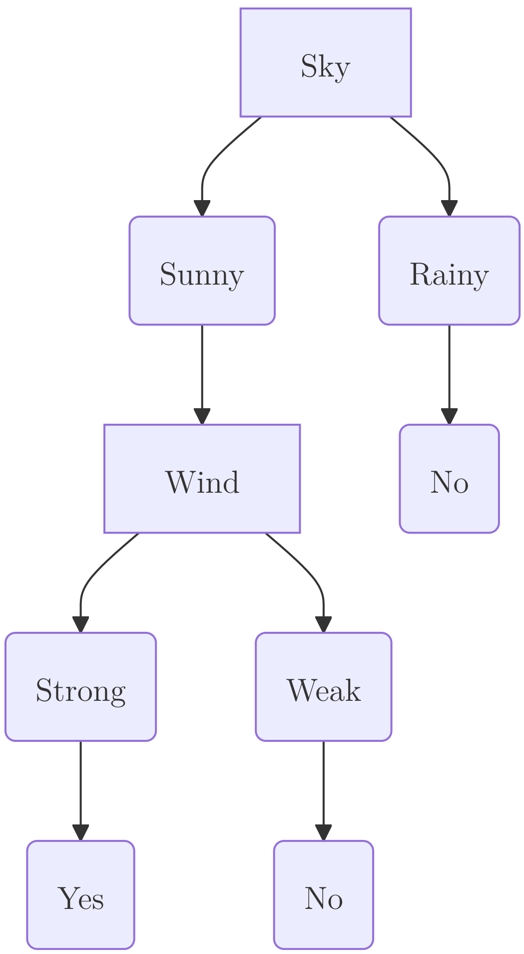
Since `Wind` is the attribute with highest information gain $IG(Wind) = 0.811$, we select `Wind` as the root node.

3.2.2. Sky: Rainy

Because there is only **one training example** for this case, we set the label to be the same as the example, which is `No`.

3.3. Third Splitting

Because `Wind` can perfectly identify `EnjoySport?` (`Wind` : `Weak` => `No` ; `Wind` : `Strong` => `Yes`). We can now draw the decision tree without further calculating the Information Gain.



4. Problem (d)

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?

ANSWER:

We will illustrate how to apply the Candidate-Elimination algorithm in the hypothesis space of decision trees using the data from Table 2.1 (reproduced below for reference):

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`

The Candidate-Elimination algorithm maintains a general boundary G (the set of maximally general hypotheses consistent with the data) and a specific boundary S (the set of maximally specific hypotheses consistent with the data). When a new example arrives, both S and G are updated to ensure consistency with all observed examples while preserving their boundary roles.

4.1. After the First Training Example

- **Example 1:** (`Sky` = `Sunny` , `AirTemp` = `Warm` , `Humidity` = `Normal` , `Wind` = `Strong` , `Water` = `Warm` , `Forecast` = `Same`) with `EnjoySport` = `Yes` .
  Specific boundary S: Because we have only one positive example and no negative examples, the most specific decision tree that classifies this instance as `Yes` and everything else as `No` can be visualized as:

$$\begin{cases} \text{If } (\mathit{Sky} = \mathit{Sunny} \land \mathit{AirTemp} = \mathit{Warm} \land \mathit{Humidity} = \mathit{Normal} \land \ldots) \\ \qquad \text{then Yes} \\ \\ \text{Else No} \end{cases}$$

```
This tree is as specific as possible while remaining consistent with the
single positive example.
**General boundary $G$:** With no negative examples, the most general
hypothesis that remains consistent simply classifies all instances as
`Yes`. In tree form, this might be: $$\text{Tree}_G: \text{(No tests) } \to
\text{Yes}$$
```

```
Thus, after the first example: $$ S = \{\text{Tree}_S\}, \quad G = \
{\text{Tree}_G\} $$
```

4.2. After the Second Training Example

- **Example 2:** (`Sky` = `Sunny` , `AirTemp` = `Warm` , `Humidity` = `High` , `Wind` = `Strong` , `Water` = `Warm` , `Forecast` = `Same`) with `EnjoySport` = `Yes` . Since this second example is also positive, we check how our current S and G classify it.

  ```
  **Refining $S$:** - The current Tree$_S$ (from **Example 1**) specifies
  “`Humidity` = `Normal`” as part of the condition. But now we see a positive
  example with `Humidity` = `High`.
  - To remain consistent with both positive examples, we must *generalize*
  Tree$_S$ minimally.
  ```

$$\begin{cases} \text{If } (\mathit{Sky} = \mathit{Sunny} \wedge \mathit{AirTemp} = \mathit{Warm} \wedge \mathit{Humidity} = ? \wedge \ldots) \\ \qquad \text{then Yes} \\ \\ \text{Else No} \end{cases}$$

$$S = \{\text{Tree}_S\}$$

Refining G: - The old Tree_G simply classifies everything as `Yes` . It is still consistent with these two *positive* examples (and we have no negative examples yet), so no change is required. - Hence,

$$G = \{\text{Tree}_G\}$$

4.3. After the Third Training Example

- **Example 3:** (`Sky` = `Rainy` , `AirTemp` = `Cold` , `Humidity` = `High` , `Wind` = `Strong` , `Water` = `Warm` , `Forecast` = `Change`) with `EnjoySport` = `No` . Since this third example is *negative*, we check how our current S and G classify it.

  ```
  **Refining $S$:** - The current Tree$_S$ (from Example 2) is still
  consistent with these two *positive* examples and does not match the
  negative example, so no change is required.

  **Refining $G$:** - The old Tree$_G$, with the introduction of a negative
  example, is now overly generalized and need to be more specific.
  ```

$$\begin{cases} \text{If } (Sky = Sunny \wedge AirTemp =? \wedge Humidity =? \wedge \ldots) \\ \text{Or } (Sky =? \wedge AirTemp = Warm \wedge Humidity =? \wedge \ldots) \\ \text{Or } (\ldots \wedge Wind =? \wedge Water =? \wedge Forecast = Same) \\ \qquad \text{then Yes} \\ \\ \text{Else No} \end{cases}$$

This tree is as general as possible while remaining consistent with the examples.

$$G = \{\text{Tree}_G\}$$

4.4. After the Fourth Training Example (Final S and G)

- **Example 4:** (`Sky` = `Sunny` , `AirTemp` = `Warm` , `Humidity` = `High` , `Wind` = `Strong` , `Water` = `Cool` , `Forecast` = `Change`) with `EnjoySport` = `Yes` . This fourth example is *positive*, we now check how our current S and G classify it.

```
**Refining $S$:** - The current Tree$_S$ (from Example 2) specifies
`Forecast` = `Same` and
`Water` = `Warm` as part of the condition. But now we see a positive
example with
`Forecast` = `Change` and `Water` = `Cool`. - To remain consistent with the
positive
examples, we must *generalize* Tree$_S$ minimally.
```

$$\text{Tree}_S : \begin{cases} \text{If } (Sky = Sunny \wedge AirTemp = Warm \wedge Humidity =? \wedge \\ Wind = Strong \wedge Water =? \wedge Forecast =?) \\ \qquad \text{then Yes} \\ \\ \text{Else No} \end{cases}$$

$$S = \{\text{Tree}_S\}$$

Refining G: - Tree$_G$, also have one case remove (`Forecast` = `Same`) due to being too specific.

$$\text{Tree}_G : \begin{cases} \text{If } (Sky = Sunny \wedge AirTemp =? \wedge Humidity =? \wedge \ldots) \\ \text{Or } (Sky =? \wedge AirTemp = Warm \wedge Humidity =? \wedge \ldots) \\ \qquad \text{then Yes} \\ \\ \text{Else No} \end{cases}$$

```
This tree is as general as possible while remaining consistent with the
examples. $$G = \{\text{Tree}_G\}$$
```

4.5. Difficulties in Applying Candidate-Elimination to Decision Trees

When using decision trees as hypotheses, several practical issues arise:

1. **Complexity of the Hypothesis Space:** The space of all possible decision trees can be extremely large or even infinite (for continuous attributes). Enumerating minimal generalizations or specializations can be very expensive.
2. **Defining Generality:** Comparing two decision trees to decide which is `more general'' or` more specific'' is not straightforward, because different tree structures may represent the same function.
3. **Multiple Minimal Refinements:** When encountering a negative example, there can be many ways to refine a tree to exclude it, leading to a large or fragmented G set.
4. **Lack of Canonical Forms:** Decision trees can represent the same concept with different structures. Maintaining canonical (unique) representations is non-trivial.
5. **Overfitting and Pruning:** Standard decision-tree learning (\textsc{ID3}, \textsc{C4.5}) uses heuristic splitting and pruning to control complexity. The pure Candidate-Elimination framework does not incorporate such heuristics, making it challenging to manage overfitting or large hypothesis spaces.

In conclusion, although it is ***theoretically*** possible to extend the Candidate-Elimination algorithm to any well-defined hypothesis space (including decision trees), the practical obstacles outlined above make it difficult to implement effectively for real-world decision-tree learning.