UNIT 3 U3/0/2024

Machine Learning

COS4852

Year module

Department of Computer Science

School of Computing

CONTENTS

This document contains the material for UNIT 3 for COS4852 for 2024.

A decision tree is a tree-like visual representation that work similarly to a flow-chart to make or support decisions. Each node in the decision tree is an attribute that splits the data set into subsets that correspond to specific values of that attribute. Each node then becomes a single decision point, where a specific value of the attribute leads to sub-trees, until all the attributes are assigned to a node, and final decision values are reached.

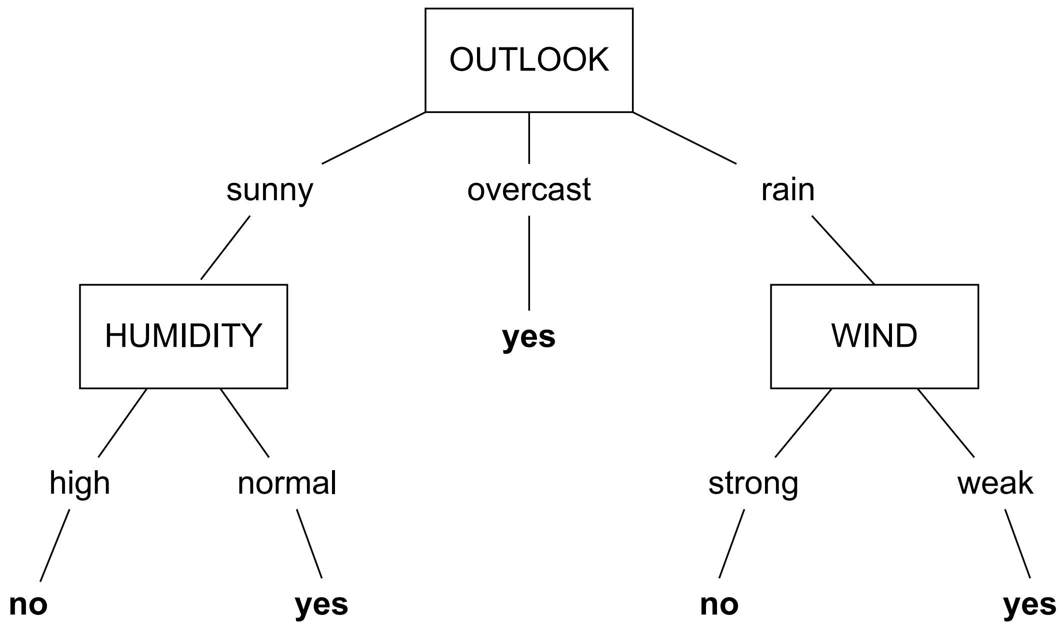

Figure 1: Example decision tree

Figure 1 shows an example of a small decision tree that could be used to determine whether to play sport based on the values of three weather variables: *Outlook*, *Humidity* and *Wind*.

1 OUTCOMES

In this Unit you will learn more about the theoretical basis of decision trees, and understand how to apply one of the algorithms used to construct a decision tree from a dataset. You will learn how to describe and solve a learning problem using decision tree learning.

You will:

1. Understand the relationship between Boolean function, binary decision trees and decision lists.
2. Learn about the theoretical basis of decision trees.
3. Understand what kinds of problems can be solved using decision trees.
4. Understand how the ID3 algorithm works.

5. Learn how to solve classification problems using ID3.

After completion of this Unit you will be able to:

1. Translate a Boolean function into a binary decision tree.
2. Convert a Boolean function into a decision list.
3. Understand and recognise appropriate learning problems that can be solved with decision tree learning.
4. Design a *Classification System* using decision trees.
5. Discuss the theoretical basis of decision trees.
6. Understand and describe how decision tree search is performed in hypothesis space, including the inductive bias implicit in decision tree learning.
7. Understand the advantages and limitations of decision trees, including overfitting of data, continuous-valued attributes, alternative methods for selecting attributes, missing attribute values and attributes with different costs.
8. Discuss what kinds of problems can be solved using decision trees.
9. Solve classification problems by implementing the ID3 algorithm on given data sets.

2 INTRODUCTION

In this Unit you will investigate the theory of decision trees and learn how to describe and solve a learning problem using decision tree learning, using the ID3 algorithm.

There are many algorithms to construct decision trees. The most famous of these is Ross Quinlan's ID3 algorithm that are used to construct a decision tree on a set of discrete and integer data values. There are variants of ID3 that can operate on continuous-valued datasets, as well as variants that use a statistical approach. There are also more complex algorithms that construct a collection of trees, called a forest-of-trees, which, although more complex, give more options for making accurate decision based on complex data.

3 PREPARATION

3.1 Online textbooks

Chapter 6 in Nilsson's book works through the ID3 algorithm for decision tree construction, using a slightly different notation from what we will be using.

3.2 Textbooks

Chapter 3 of Mitchell's book goes into some depth on decision trees.

Sections 18.1 to 18.4 in Russell and Norvig's 3rd edition is also a good source in decision lists.

3.3 Online material

Here is simple explanation of Entropy and Information Gain.

The original 1986 article by Ross Quinlan describes one of the most successful algorithms to create decision trees.

This IBM article gives a detailed discussion on what a decision tree is and does, and how to do the basic ID3 calculations.

The Wikipedia page on ID3 gives a good overview of the ID3 algorithm.

4 DISCUSSION

4.1 Boolean Functions and Binary Decision Trees

Boolean function:

$$f_1(A, B) = \neg A \wedge B$$

The truth table for this Boolean function is:

A	B	$\neg A$	f_1
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0

Start by choosing A as the root node. This gives us the binary decision tree as in Figure 2. On the diagram you can see the mapping between specific parts of the truth table and the binary decision tree. Each leaf node corresponds to one row in the truth table, while the level above the leaf nodes correspond to two rows in the truth table, etc. By merging leaf nodes with the same value the tree can be simplified, as in Figure 3.

Using B as the start node results in a different binary decision tree. In this particular case the tree turns out to be as simple as the first. This is not the case for all decision trees.

The binary decision tree starting with B is shown in Figure 4.

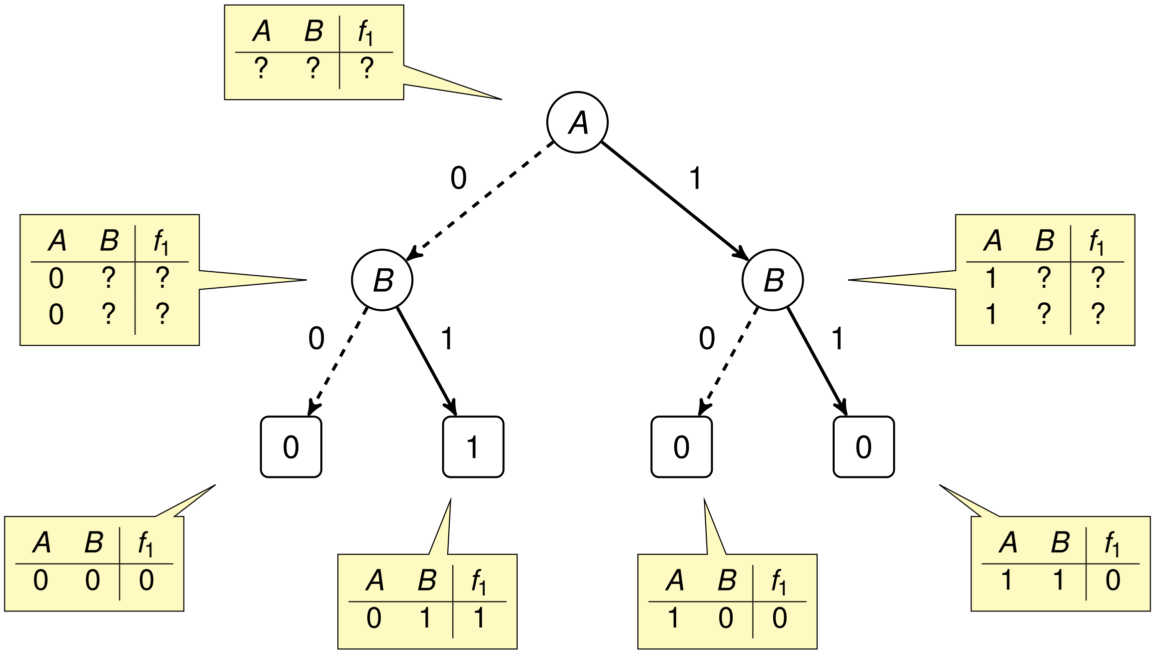

Figure 2: A binary decision tree for f_1 starting with A.

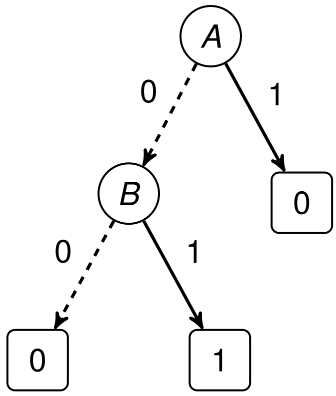

Figure 3: A simplified binary decision tree for $f_1(A, B) = \neg A \wedge B$ starting with A.

Decision lists

Rivest wrote a paper on how to create decision lists from a Boolean function. The paper goes into some depth in how to do this.

Nilsson's book summarises the concept on p.22.

You can think of a decision list as a binary decision tree where each node divides the data set into two so that one branch has a binary value ((0, 1) or $\{T, F\}$ as output, and the other branch leads leads to further subdivision of the dataset. By writing a Boolean function in a DNF form, this becomes reasonably obvious. Another method that works well is to draw a Karnaugh diagram of the function and reduce the function through the diagram using the same process that would be used to create a DNF form.

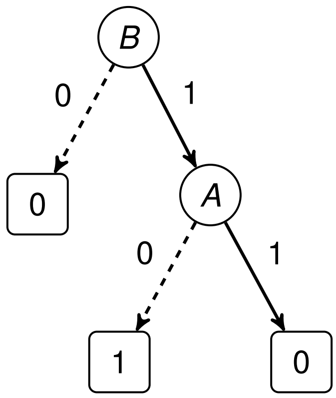

Figure 4: A binary decision tree for f_1 starting with B.

4.2 The ID3 algorithm

The ID3 algorithm can be described by the following pseudocode:

Require: ID3(node, instances, targets values, attributes)
  Root $\leftarrow$ node
  $V \leftarrow \{$instances$\}$
  $T \leftarrow \{$target values$\}$
  $A \leftarrow \{$attributes$\}$
  if all $v_i \in V = \oplus$ **then return** *Root* with label $\oplus$
  end if
  if all $v_i \in V = \ominus$ **then return** *Root* with label $\ominus$
  end if
  if $A = \emptyset$ **then return** single node tree *Root* with label = majority value of t in A
  else
    $A \leftarrow$ the attribute that best classifies instances in subset
    $Root = A$
    while $v_i \in A$ **do**
      add new branch where $A = v_i$
      $V(v_i) \leftarrow$ subset of instances of A that have value v_i
      if $V(v_i) \in \emptyset$ **then**
        add leaf node with label = majority value of T in A
      else
        add subtree ID3(node, $V(v_i)$, T, A)
      end if
    end while
  end if
  return *Root*

Constructing a decision tree is a recursive process to decide which attribute to use at each node of the decision tree. We want to choose the attribute that is the “best” at classifying the instances in the data set. “Best” here is a quantitative measure (a number). One such measure is a statistical measure called *Information Gain*. This determines how well a given attribute separates the data set as measured against the target classification.

ID3 uses the attribute with the highest *Information Gain* as the next node in constructing the tree. The ID3 algorithm is a recursive algorithm that constructs sub-trees for attribute values of each node, using the sub-sets of the data matching the attribute value of the sub-tree.

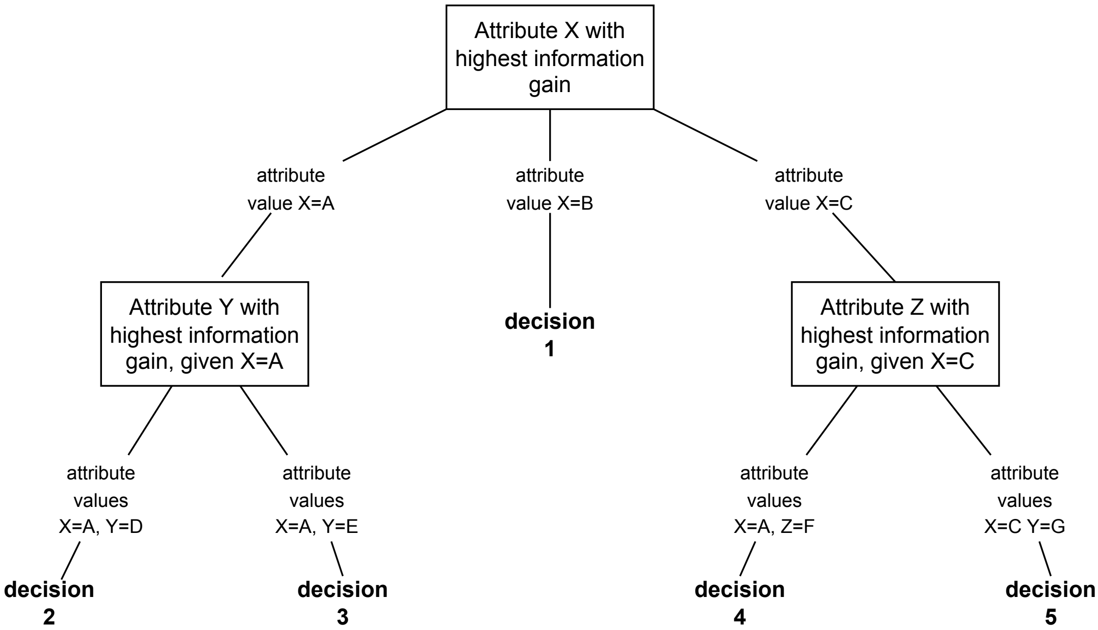

Figure 5: Decision tree showing how nodes are selected in the ID3 algorithm

Figure 5 shows a decision tree with labels indicating how ID3 selects nodes in the construction of the tree.

To understand *Information Gain* we need to first look at the concept of *Entropy*.

Entropy

Entropy is an important concept in thermodynamics. Claude Shannon saw that the concept could be used to describe how much information there is in the outcome of a random discrete variable (such as determining if a coin will land heads up or not, or to make sure that communication over a network does not lose information). We can use the concept to measure the “usefulness” of a variable in terms of its information content. This idea forms the core of the decision tree construction process in ID3.

Given a discrete random variable X, which takes values in the alphabet $\mathcal{X}$ and is distributed according to $p : \mathcal{X} \to [0, 1]$:

$$\mathrm{H}(X) := -\sum_{x\in\mathcal{X}} p(x) \log p(x) = \mathbb{E}[-\log p(X)]$$

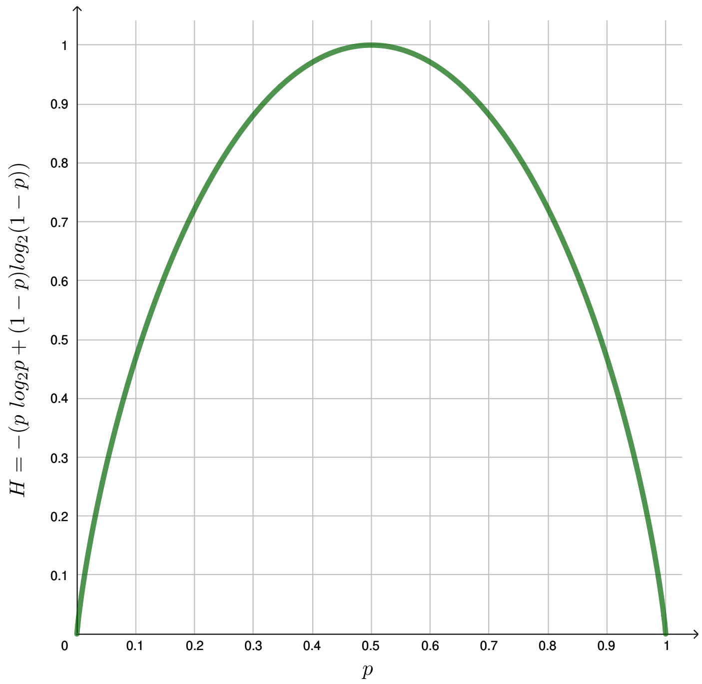

Figure 6: Entropy of a single variable

where the sum is calculated over all possible alphabet values $\mathcal{X}$. The base of the log matches the distribution of *p*. For example $\log_2$ is used when the target values in the data is binary (yes/no or T/F).

Figure 6 shows the Entropy for a single variable. Here you can see that Entropy is always positive and can never be larger than 1.

Information Gain

Entropy can be viewed as a measure of the impurity of a collection of instances (a data set). In order to construct a decision tree we want to repeatedly sub-divide our data set in such a way that we create the largest reduction in entropy with each sub-division. The *Information Gain* of an attribute *A* relative to a dataset *S* is defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where $Values(A)$ is the set of all possible values attribute A can have, $|S_v|$ is the subset of S where A has the values v.

ID3 uses *Information Gain* to find the attribute that splits the dataset in such a way that we have the highest reduction in entropy, or, as calculated above the highest *Information Gain*. The worked example in Subsection 4.3 below will illustrate this in more detail.

4.3 Worked example

We will use the data set in Table 1 to work through an example of the ID3 algorithm.

Table 1: A set of object, their attributes and classes (positive or negative)

Colour	Form	Hollow	Transparent	Class
RED	cube	yes	yes	$\oplus$
BLUE	sphere	no	yes	$\ominus$
GREEN	pyramid	no	yes	$\ominus$
RED	sphere	no	no	$\ominus$
GREEN	pyramid	yes	no	$\oplus$
GREEN	cube	no	no	$\ominus$
BLUE	cube	yes	no	$\ominus$
BLUE	pyramid	yes	yes	$\oplus$
RED	cube	yes	no	$\ominus$
BLUE	pyramid	no	no	$\ominus$
GREEN	cube	no	yes	$\oplus$
RED	pyramid	yes	no	$\oplus$
GREEN	cube	yes	no	$\ominus$
GREEN	sphere	no	yes	$\ominus$

First, we calculate the Entropy for the entire data set. We do this as a baseline against which to compare which attribute will become our root note. This is in turn is done by calculating the *Information Gain* for each attribute.

This is a binary classification problem. There are 14 instances, of which 5 result in **Class** = $\oplus$ and 9 gives **Class** = $\ominus$. In other words:

$$\text{Entropy}(S) \equiv \text{Entropy}([5_\oplus, 9_\ominus])$$

There are four attributes, which we can shorten to (C, F, H, T) whose combination of values determine the value of the target attribute, **Class**.

Calculate the Entropy of the data set:

$$\begin{aligned}
\text{Entropy}(S) &\equiv \sum_{i=1}^{c} -p_i log_2(p_i) \\
&= -p_\oplus\ log_2(p_\oplus) - p_\ominus\ log_2(p_\ominus) \\
&= -{}^5\!/_{14}\ log_2({}^5\!/_{14}) - {}^9\!/_{14}\ log_2({}^9\!/_{14}) \\
&= (-0.3571 \times -1.4854) + (-0.6429 \times -0.6374) \\
&= 0.9403
\end{aligned}$$

Attribute C can take on three values (shortened here):

$$\begin{aligned}
\text{Values}(C) &= R, G, B \\
\text{S}_C &= [5_\oplus, 9_\ominus] \\
\text{S}_{C=R} &\leftarrow [2_\oplus, 2_\ominus] \\
\text{S}_{C=G} &\leftarrow [2_\oplus, 4_\ominus] \\
\text{S}_{C=B} &\leftarrow [1_\oplus, 3_\ominus]
\end{aligned}$$

Calculate the Entropy values of the three subsets of the data associated with the values of the attribute C:

$$\begin{aligned}
\text{Entropy}(\text{S}_{C=R}) &= -{}^2\!/\!_4\, log_2({}^2\!/\!_4) - {}^2\!/\!_4\, log_2({}^2\!/\!_4) \\
&= 1.0000 \\
\text{Entropy}(\text{S}_{C=G}) &= -{}^2\!/\!_6\, log_2({}^2\!/\!_6) - {}^4\!/\!_6\, log_2({}^4\!/\!_6) \\
&= 0.9183 \\
\text{Entropy}(\text{S}_{C=B}) &= -{}^1\!/\!_4\, log_2({}^1\!/\!_4) - {}^3\!/\!_4\, log_2({}^3\!/\!_4) \\
&= 0.8112
\end{aligned}$$

Calculate the *Information Gain* for attribute C:

$$\begin{aligned}
\text{Gain(S, C)} &= \text{Entropy(S)} - \sum_{v \in \{R,G,B\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \\
&= \text{Entropy(S)} - {}^4\!/\!_{14}\, \text{Entropy}(S_{C=R}) - {}^6\!/\!_{14}\, \text{Entropy}(S_{C=G}) - {}^4\!/\!_{14}\, \text{Entropy}(S_{C=B}) \\
&= 0.9403 - {}^4\!/\!_{14} \times 1.0000 - {}^6\!/\!_{14} \times 0.9183 - {}^4\!/\!_{14} \times 0.8112 \\
&= 0.0292
\end{aligned}$$

Repeat these calculations for the other three attributes as well. We now get all the *Information Gain* values:

$$\begin{aligned}
\text{Gain(S, C)} &= 0.0292 \\
\text{Gain(S, F)} &= 0.2000 \\
\text{Gain(S, H)} &= 0.1518 \\
\text{Gain(S, T)} &= 0.0481
\end{aligned}$$

The attribute with the highest *Information Gain* causes the highest reduction in entropy. This is the attribute **Form** with Gain(S, F) = 0.2000, which then becomes the root node of the decision tree, as shown in Figure 7.

The ID3 algorithm now recurses over the subsets of the data associated with the three branches of the root node of the decision tree.

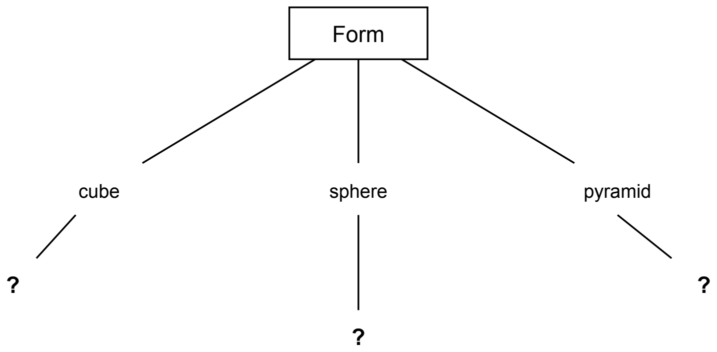

Figure 7: Decision tree after the first set of calculations

Table 2: Subset of the data with **Form**=cube

Colour	**Form**	**Hollow**	**Transparent**	**Class**
RED	cube	yes	yes	$\oplus$
GREEN	cube	no	no	$\ominus$
BLUE	cube	yes	no	$\ominus$
RED	cube	yes	no	$\ominus$
GREEN	cube	no	yes	$\oplus$
GREEN	cube	yes	no	$\ominus$

In Table 2 are 6 instances, of which 2 result in **Class** = $\oplus$ and 4 gives **Class** = $\ominus$. Therefore:

$$\text{Entropy}(S_{\text{F=c}}) \equiv \text{Entropy}([2_\oplus, 4_\ominus])$$

Calculate the Entropy of this sub-set of the data:

$$\begin{aligned}
\text{Entropy}(S_{\text{F=c}}) &\equiv \sum_{i=1}^{c} -p_i log_2(p_i) \\
&= -p_\oplus\ log_2(p_\oplus) - p_\ominus\ log_2(p_\ominus) \\
&= -{}^2\!/\!_6\ log_2({}^2\!/\!_6) - {}^4\!/\!_6\ log_2({}^4\!/\!_6) \\
&= (-0.3333 \times -1.5850) + (-0.6667 \times -0.5850) \\
&= 0.9183
\end{aligned}$$

Calculate the Entropy values of the three subsets of the data associated with the values of the attribute C:

$$\begin{aligned}
\text{Entropy}(S_{\text{F=c,C=R}}) &= -{}^1\!/\!_2\ log_2({}^1\!/\!_2) - {}^1\!/\!_2\ log_2({}^1\!/\!_2) \\
&= 1.0000 \\
\text{Entropy}(S_{\text{F=c,C=G}}) &= -{}^1\!/\!_3\ log_2({}^1\!/\!_3) - {}^2\!/\!_3\ log_2({}^2\!/\!_3) \\
&= 0.9183 \\
\text{Entropy}(S_{\text{F=c,C=B}}) &= -{}^0\!/\!_1\ log_2({}^0\!/\!_1) - {}^1\!/\!_1\ log_2({}^1\!/\!_1) \\
&= 0.0000
\end{aligned}$$

Table 3: Subset of the data with **Form**=sphere

Colour	Form	Hollow	Transparent	Class
BLUE	sphere	no	yes	$\ominus$
RED	sphere	no	no	$\ominus$
GREEN	sphere	no	yes	$\ominus$

Table 4: Subset of the data with **Form**=pyramid

Colour	Form	Hollow	Transparent	Class
GREEN	pyramid	no	yes	$\ominus$
GREEN	pyramid	yes	no	$\oplus$
BLUE	pyramid	yes	yes	$\oplus$
BLUE	pyramid	no	no	$\ominus$
RED	pyramid	yes	no	$\oplus$

Calculate the *Information Gain* for attribute *C*, where **Form**=cube:

$$\begin{aligned}
\mathrm{Gain(S, C}_{F=c}) &= \mathrm{Entropy(S)} - \sum_{v \in \{R,G,B\}} \frac{|S_v|}{|S|} \mathrm{Entropy}(S_v) \\
&= \mathrm{Entropy(S)} - {}^2\!/\!_6\ \mathrm{Entropy}(S_{C=R}) - {}^3\!/\!_6\ \mathrm{Entropy}(S_{C=G}) - {}^1\!/\!_6\ \mathrm{Entropy}(S_{C=B}) \\
&= 0.9183 - {}^2\!/\!_6 \times 1.0000 - {}^3\!/\!_6 \times 0.9183 - {}^1\!/\!_6 \times 0.0000 \\
&= 0.1258
\end{aligned}$$

We do similar calculations for the rest of the subset to get:

$$\begin{aligned}
\mathrm{Gain(S, C}_{F=c}) &= 0.1258 \\
\mathrm{Gain(S, H}_{F=c}) &= 0.0441 \\
\mathrm{Gain(S, T}_{F=c}) &= 0.9183
\end{aligned}$$

The attribute with the highest *Information Gain* is **Transparent**, which then becomes the next node in the decision tree, under the branch with the value **Form**=cube. The data in Table 3 show that all the instances have output $\ominus$, which means that we can define a leaf node under **Form**=sphere. The result of these calculations gives the decision tree as in Figure 8.

In Table 4, where **Form**=pyramid, are 5 instances, of which 3 result in **Class** = $\oplus$ and 2 gives **Class** = $\ominus$. Therefore:

$$\mathrm{Entropy}(S_{\mathrm{F=p}}) \equiv \mathrm{Entropy}([3_\oplus, 2_\ominus])$$

We do the same calculations are above to get:

$$\mathrm{Gain(S, C}_{F=c}) = 0.9710$$

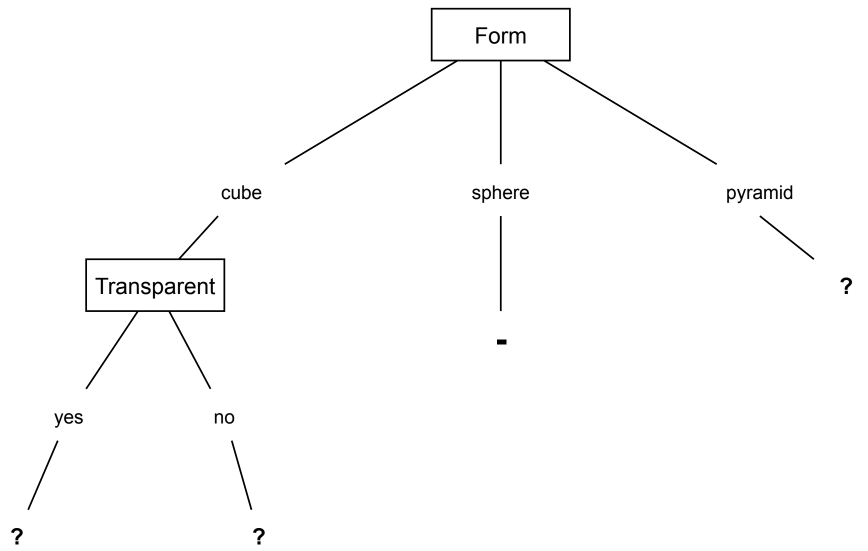

Figure 8: Decision tree after the second set of calculations and the observation for **Form**=sphere

and

$$
\begin{aligned}
\text{Gain}(S, C_{F=p}) &= 0.1710 \\
\text{Gain}(S, H_{F=p}) &= 0.9710 \\
\text{Gain}(S, T_{F=p}) &= 0.9710
\end{aligned}
$$

We now see an interesting phenomenon. The highest *Information Gain* values are the same for two possible branches. We can choose either, as they have the same effect in reducing Entropy. We have already used **Transparent** in another branch, so choosing **Hollow** will result in a simpler tree (Occam's razor) to get the decision tree as in Figure 9.

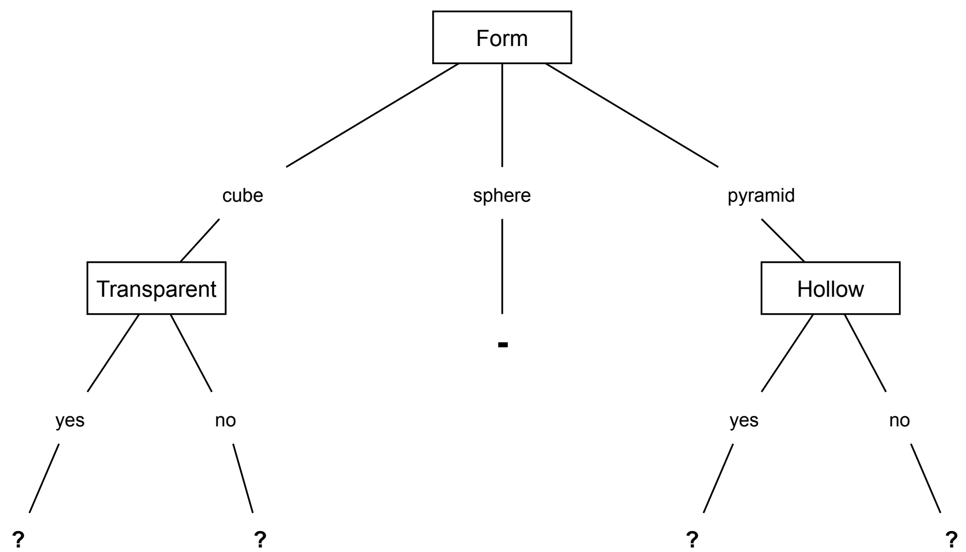

Figure 9: Decision tree after the 4th set of calculations

We now have four branches of the tree to investigate, and possibly repeat the calculations. These branches correspond to sub-sets of the data. Tables 5 and 6 are the subsets under the branches of **Transparent**.

Table 5: Subset of the data with **Form**=cube and **Transparent**=yes

Colour	**Form**	**Hollow**	**Transparent**	**Class**
RED	cube	yes	yes	$\oplus$
GREEN	cube	no	yes	$\oplus$

Table 6: Subset of the data with **Form**=cube and **Transparent**=no

Colour	**Form**	**Hollow**	**Transparent**	**Class**
GREEN	cube	no	no	$\ominus$
BLUE	cube	yes	no	$\ominus$
RED	cube	yes	no	$\ominus$
GREEN	cube	yes	no	$\ominus$

In both of these we see that there is only one output class in each. This means that we have two more leaf nodes, as in Figure 10.

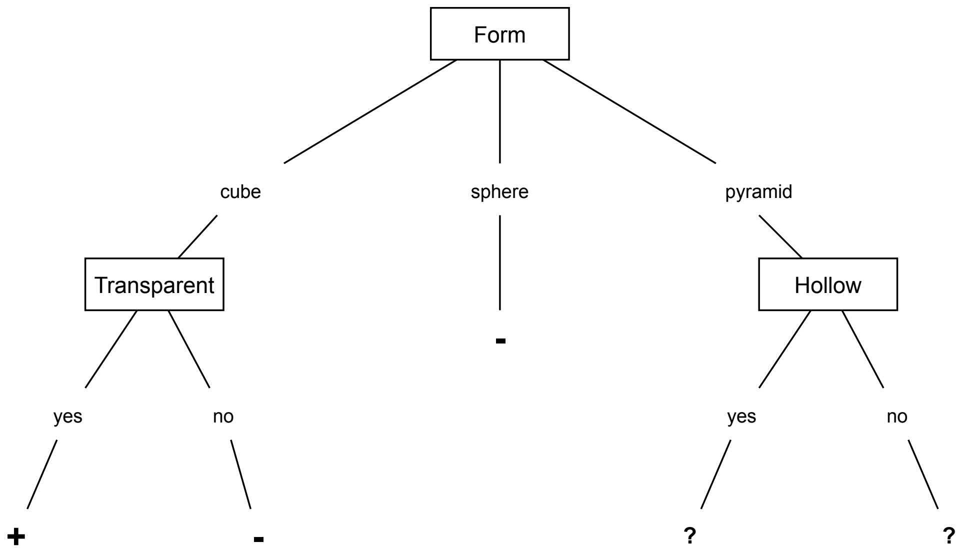

Figure 10: Decision tree after the 5th set of observations

We are now left with two more subsets to investigate - those for the branches of **Hollow**. Tables 7 and 8 show these sub-sets.

Again, we observe a similar phenomenon as with the previous two subsets, namely that there is only a single class in each. This means that we have our final two leaf nodes, as in Figure 11.

Table 7: Subset of the data with **Form**=pyramid and **Hollow**=yes

Colour	**Form**	**Hollow**	**Transparent**	**Class**
GREEN	pyramid	yes	no	$\oplus$
BLUE	pyramid	yes	yes	$\oplus$
RED	pyramid	yes	no	$\oplus$

Table 8: Subset of the data with **Form**=pyramid and **Hollow**=no

Colour	**Form**	**Hollow**	**Transparent**	**Class**
GREEN	pyramid	no	yes	$\ominus$
BLUE	pyramid	no	no	$\ominus$

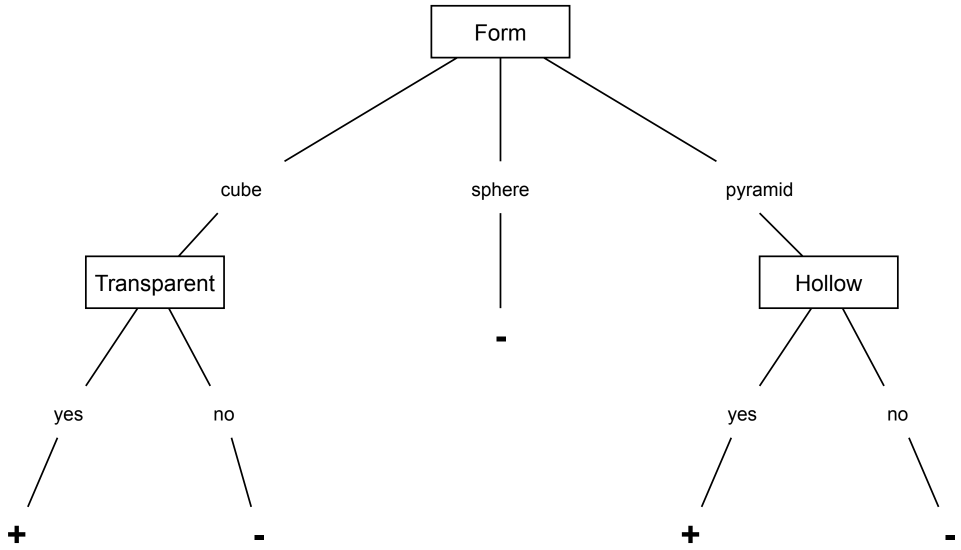

Figure 11: Decision tree after the final set of observations

5 ACTIVITIES

5.1 TASK 1 - STUDY THE MATERIAL

Find and read all the online material shown earlier in this document. Study the relevant concepts carefully and thoroughly.

5.2 TASK 2 - OTHER DECISION TREE ALGORITHMS

Find resources (some of this will be in the textbooks and material you have already studied in the first task) on other algorithms for contructing decision trees. Some of these algorithms include ID3 (what you've studied here), C4.5, and CART.

Study these algorithms so that you understand how they work, and on what kinds of data sets they can be applied. What are the differences? What are the advantages and shortcomings of these algorithms. What would you do with missing or incorrect data? How would you handle non-categorical or continuous data? Can you use other costs functions? Can you use different cost functions in different parts of the data set? Why and when would you do so?

5.3 TASK 3

Find resources on more advanced extentions of decision-tree learning. Look specifically at ensemble methods, such as bagging an boosting, and their further extension into random forests.
