CE213 Artificial Intelligence – Lecture 11

Decision Tree Induction: Part 1

What is induction?

What is a decision tree?

How to do decision tree induction?
(Information and information gain will be used for decision tree induction)

Is it structural learning or parametric learning?

What is Induction?

Induction is learning (or drawing conclusions) by generalising from samples or experiences. It may be contrasted with ***deduction***.

Consider the following proposition:

The sum of any two odd numbers is even

You could repeatedly add pairs of odd numbers and notice that the result is always even.

That is ***induction***

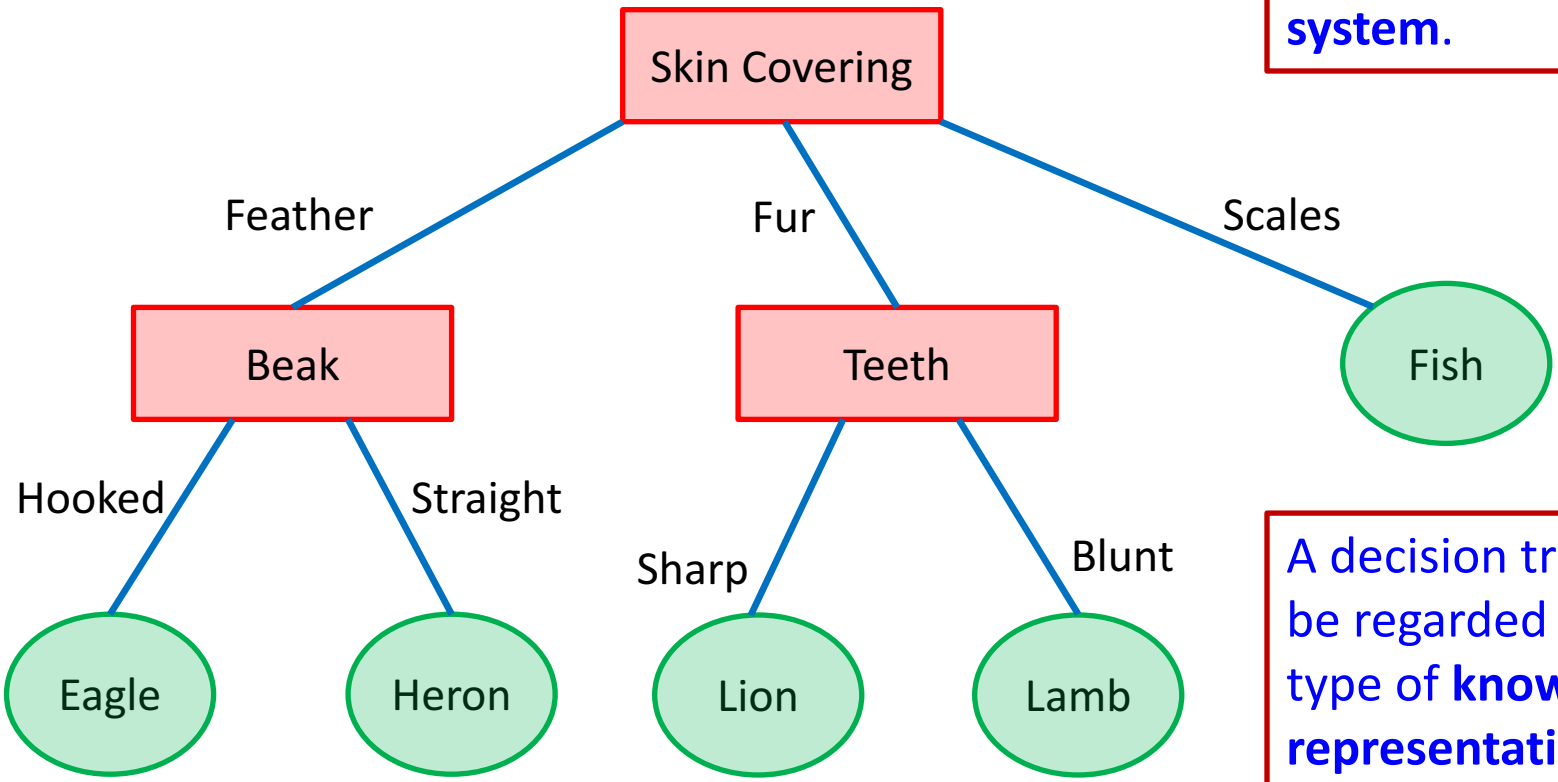
Or you could construct a mathematical proof that the result will necessarily be even.

That is ***deduction***

What is a Decision Tree?

Let's start with an example:

a very simple decision tree (or subtree):



You may compare it with a **search tree**, or a **production system**.

A decision tree can be regarded as a type of **knowledge representation**.

What is a Decision Tree? (2)

It is a tree for making decisions based on attribute values, in which:

- Each leaf node is associated with a ***class or decision***.
- Each non-leaf node is associated with one of the ***attributes*** that objects possess.
- Each branch is associated with a particular ***value*** that the attribute of the connected parent node can take.

So what is decision tree induction?

It is a procedure that, based on a given set of training samples, attempts to build a decision tree capable of predicting the class of any new sample (generalising from training samples).

The Basic Decision Tree Induction Procedure

```
METHOD buildDecTree(samples,atts)
Create node N if necessary; //starting as a node, ending as a tree
IF samples are all in same class
THEN RETURN N labelled with that class;
IF atts is empty
THEN RETURN N labelled with modal class [1];
bestAtt = chooseBestAtt(samples,atts);
label N with bestAtt;
FOR each value a_i of bestAtt
    s_i = subset of samples with bestAtt = a_i;
    IF s_i is not empty
    THEN
        newAtts = atts - bestAtt;
        subtree = buildDecTree(s_i,newAtts); //recursive
        attach subtree as child of N;
    ELSE
        Create leaf node L;
        Label L with modal class;
        attach L as child of N;
RETURN N;
```

Most important

Will come back to this pseudo code later with an example.

{Note 1: Model class is the class of the group with the maximum number of samples or highest frequency}

What is a Training Sample Set?

What is a training sample set?

It is a set of labelled samples, drawn from some population of possible samples.

The training set is almost always a very small fraction of the population.

What is a sample?

Typically decision trees operate using samples that take the form of *feature or attribute vector*.

An ***attribute vector*** *is* simply a vector whose elements are the *values* taken by the *attributes* of objects in the sample set.

e.g., a heron might be represented as follows:

Skin Covering	*Beak*	*Teeth*	*Class*
Feather	Straight	None	Heron

Choosing the Best Attribute

What is "the best attribute"?

Many possible definitions.

A reasonable answer:

The attribute that best discriminates the samples with respect to their classes.

So what does "best discriminates" mean?

Still many possible answers.

Many different criteria have been used.

The most popular is ***information gain***.

We need to know how to calculate information in order to calculate information gain.

Shannon's Information Formula

Given a situation in which there are N unknown outcomes:

How much information have you acquired once you know what the outcome is?

Let's begin by considering the simplest possible situation:

2 outcomes, each equally likely

e.g., A coin toss

We can ***define*** the amount of ***information*** you acquire when you learn the outcome of such an event as **1 bit.** (0 for one outcome and 1 for the other outcome)

Shannon's Information Formula (2)

Now consider picking 1 card at random from a pack of 8.

i.e., 8 equiprobable outcomes (8 different cards)

One way to make the random choice would be to toss a coin 3 times.

This would provide a binary number in the range 0~7

The number could then be used to choose the card

So when you learn which of 8 equiprobable outcomes has occurred you would have acquired 3 bits of information.

Shannon's Information Formula (3)

We can extend this to other situations with equiprobable outcomes:

Toss a coin	2 outcomes	1 bit of information
Pick 1 card from 8	8 outcomes	3 bits of information
Pick 1 card from 32	32 outcomes	5 bits of information
Pick 1 card from 128	128 outcomes	7 bits of information

Or in general, for a situation or event with N equiprobable outcomes:

$$\text{Information} = \log_2(N) \quad \text{bits}$$

Since the probability of each outcome p = 1/N, we can also express this as

$$\text{Information} = -\log_2(p) \quad \text{bits}$$

For example, being told the outcome when there are 5 equiprobable outcomes:

$$\text{Information} = \log_2(5) \text{ bits} = 2.322 \text{ bits}$$

Shannon's Information Formula (4)

Non-equiprobable outcomes:

Consider picking 1 card from a pack containing 127 red and 1 black.

> There are 2 possible outcomes, red card or black card, but you would be almost certain that the result would be red.

Thus being told the outcome in this situation usually gives you less information than being told the outcome of an event with two equiprobable outcomes.

We need to modify the definition of information to reflect the fact that there is **less information to be gained when we already know that some outcomes are more likely than others**.

Shannon's Information Formula (5)

Shannon proposed the following information formula:

$$Information = -\sum_{i=1}^{N} p_i log_2(p_i) \quad \text{bits}$$

where

N is the number of alternative outcomes

p_i is the probability of the ith outcome

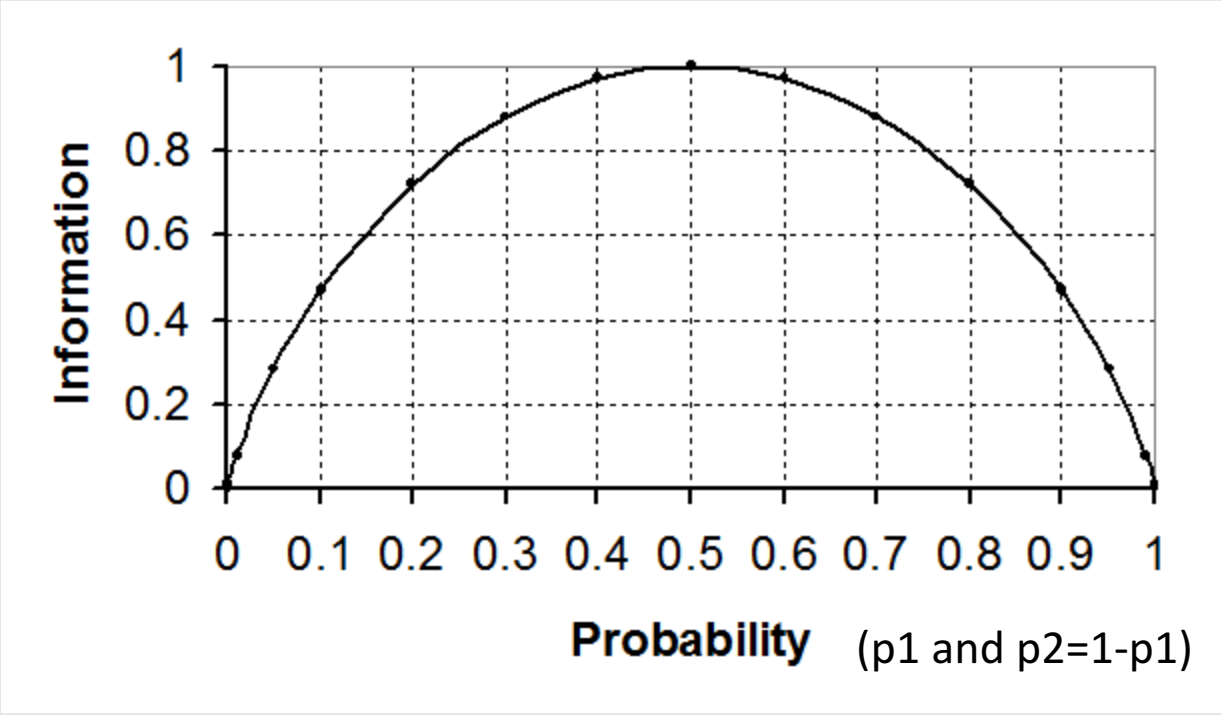
Notice that this formula reduces to $-log_2(p)$ when the outcomes are all equiprobable.

So Shannon's formula is a generalization of the formula we have already derived.

In order to calculate the information of an event, you need to know the number of outcomes and the probability of each outcome. An event may be represented as a dataset, from which the probability of each outcome can be calculated.

Shannon's Information Formula (6)

If there are only **two outcomes**, we can use the following graph to show the change of Shannon's information with the probabilities of the 2 outcomes:



Information is also sometimes called ***uncertainty*** or ***entropy***.

It is clear that higher information means higher uncertainty.

Using Information Gain to Evaluate Attributes

Information gain = Information before knowing attribute value
– Information after knowing attribute value

An example to show how knowing the value of an attribute affects the information or uncertainty about the class or outcome:

Suppose

You have a set of 100 samples (e.g., a set of colourful objects).

These samples fall in two classes, c_1 and c_2 (e.g., hot, cold):

70 samples are in c_1 and 30 samples are in c_2

How uncertain are you about the class that a sample belongs to?

$$\text{Information}_{\text{class}} = -p(c_1) \times \log_2(p(c_1)) - p(c_2) \times \log_2(p(c_2))$$
$$= -0.7 \times \log_2(0.7) - 0.3 \times \log_2(0.3)$$
$$= -0.7 \times (-0.51) - 0.3 \times (-1.74) = 0.88 \text{ bits}$$

Using Information Gain to Evaluate Attributes (2)

Now suppose *Colour* is one of the attributes with values *red* and *blue.*

The 100 samples are distributed as follows in terms of colour:

	red	*blue*
c_1	63	7
c_2	6	24

What is the information about the class of the samples whose *Colour* value is *red*?

There are 69 of them: 63 in c_1 and 6 in c_2, which form a sample subset.

So for this *subset*, $p(c_1) = 63/69 = 0.913$

and $p(c_2) = 6/69 = 0.087$

Therefore

$$\text{Information}_{\text{class|colour\_red}} = -0.913 \times \log_2(0.913) - 0.087 \times \log_2(0.087)$$

$$= 0.43 \text{ bits}$$

Using Information Gain to Evaluate Attributes (3)

Similarly, for the samples whose *Colour* value is *blue*:

There are 31 of them; 7 in c_1 and 24 in c_2, which form another sample subset.

So for this *subset*, $p(c_1) = 7/31 = 0.226$

and $p(c_2) = 24/31 = 0.774$

Hence

$$\text{Information}_{\text{class|colour\_blue}} = -0.226 \times \log_2(0.226) - 0.774 \times \log_2(0.774)$$

$$= 0.77 \text{ bits}$$

So

If we know the *Colour* is *red*, the remaining uncertainty is 0.43 bits

If we know the *Colour* is *blue*, the remaining uncertainty is 0.77 bits

How much information or uncertainty about the class will remain if we are told the *Colour*?

Using Information Gain to Assess Attributes (4)

69% of samples are *red* and 31% of samples are *blue*.

So, if we are told the Colour:

69% of the time we will be told *red*

31% of the time we will be told *blue*

Hence, the average information about the class if we are told the value of the *Colour* attribute will be

$$\text{Information}_{\text{class|colour}} = 0.69 \times 0.43 + 0.31 \times 0.77 = 0.54 \text{ bits}$$

Compare this with the information about the class if we don't know the value of *Colour,* which we calculated earlier as 0.88 bits.

Therefore, the Colour attribute provides an ***information gain***:

$$\text{Information}_{\text{class}} - \text{Information}_{\text{class|colour}} = 0.88 - 0.54 = 0.34 \text{ bits}$$

You will learn how to use **information gain** to select best attributes and thus construct decision tree in the next lecture.

Summary

Decision Tree

Non-leaf nodes represent attributes;
Leaf nodes represent classes;
Branches represent attribute values.

Decision Tree Induction

The Basic Decision Tree Induction Procedure (pseudo code)
It is structural learning.
It is recursive.

Choosing the Best Attribute

Shannon's Information Formula
Using **Information Gain** to Evaluate an Attribute

(***Example of decision tree induction using information gain: next lecture***)