



CE213 Artificial Intelligence – Lecture 12

Decision Tree Induction: Part 2

An example of decision tree induction using information gain to select best attributes

Knowledge discovery by machine learning

Some issues in decision tree induction

Pseudocode for Decision Tree Induction Revisited

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** //each branch from node N

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;

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

An Example of Decision Tree Induction

Suppose we have a training sample dataset derived from weather records, containing **four attributes**:

Attributes	Possible Values
Temperature	Warm, Cool
Cloud Cover	Overcast, Cloudy, Clear
Wind	Windy, Calm
<i>Precipitation</i>	<i>Rain, Dry</i>

We want to build a decision tree based on this sample dataset, which can predict precipitation from the other **three** attributes (Attribute Precipitation is used as class in this example).

An Example ... (2)

The training sample dataset is as follows (8 samples):

Temperature	Cloud Cover	Wind	Precipitation (class)
[warm,	overcast,	windy;	rain]
[cool,	overcast,	calm;	dry]
[cool,	cloudy,	windy;	rain]
[warm,	clear,	windy;	dry]
[cool,	clear,	windy;	dry]
[cool,	overcast,	windy;	rain]
[cool,	clear,	calm;	dry]
[warm,	overcast,	calm;	dry]

What is the modal class of this dataset?

Can you guess which attribute would be the best for predicting precipitation?

(Note: For practical applications, there could be thousands or more samples and hundreds or more attributes, but the decision tree induction procedure would be the same except for more recursive loops, as described in the pseudo code, and require more computational load.)

An Example ... (3)

Initial information (or uncertainty) about the class (Precipitation):

First we consider the initial information:

$$p(\text{rain}) = 3/8; p(\text{dry}) = 5/8$$

So for the **whole set**, $\text{Inf} = -(3/8) \times \log_2(3/8) - (5/8) \times \log_2(5/8) = 0.954$ bits

Choosing the best attribute:

Next we must choose the best attribute as the root node of the decision tree and then build up its branches.

There are three to choose from:

Temperature

Cloud Cover

Wind

An Example ... (4)

Standard procedure for
calculating information
gain from an attribute

Information Gain from attribute Temperature

Cool samples: 5 out of 8

There are 5 of these; 2 rain and 3 dry. (rain and dry are precipitation values)

So for this **subset**, $p(\text{rain}) = 2/5$ and $p(\text{dry}) = 3/5$

Hence, $\text{Inf}_{\text{cool}} = -(2/5) \times \log_2(2/5) - (3/5) \times \log_2(3/5) = 0.971$ bits

Warm samples: 3 out of 8

There are 3 of these; 1 rain and 2 dry.

So for this **subset**, $p(\text{rain}) = 1/3$ and $p(\text{dry}) = 2/3$

Hence, $\text{Inf}_{\text{warm}} = -(1/3) \times \log_2(1/3) - (2/3) \times \log_2(2/3) = 0.918$ bits

Average Information about the class given value of Temperature:

$(5/8) \times \text{Inf}_{\text{cool}} + (3/8) \times \text{Inf}_{\text{warm}} = 0.625 \times 0.971 + 0.375 \times 0.918 = 0.951$ bits

Hence, Information Gain from Temperature is

Initial Information – Average Information given value of Temperature
 $= 0.954 - 0.951 = 0.003$ bits (Very small)

An Example ... (5)

Information Gain from attribute Cloud Cover

Following a similar calculation procedure gives 0.5 bits as average information about the class given value of Cloud Cover.

Hence, Information Gain from Cloud Cover is

$$\begin{aligned} & \text{Initial Information} - \text{Average Information given value of Cloud Cover} \\ &= 0.954 - 0.5 = 0.454 \text{ bits. (Large)} \end{aligned}$$

Information Gain from attribute Wind

Following a similar calculation procedure gives 0.607 bits as average information about the class given value of Wind.

Hence, Information Gain from Wind is

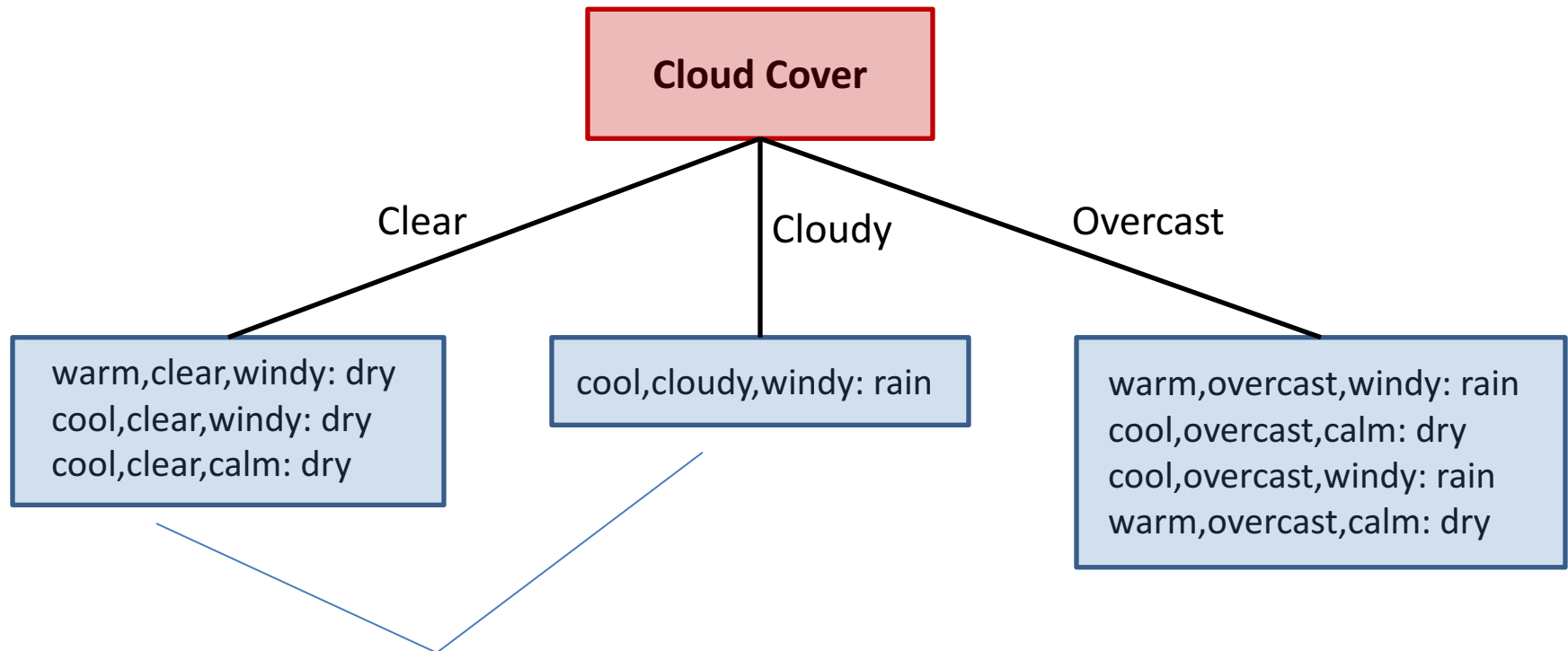
$$\begin{aligned} & \text{Initial Information} - \text{Average Information given value of Wind} \\ &= 0.954 - 0.607 = 0.347 \text{ bits. (Quite large)} \end{aligned}$$

Conclusion

Cloud Cover gives the greatest information gain and is therefore the best attribute for predicting precipitation.

An Example ... (6)

Building up the decision tree with the first best attribute as root node:



In each subset, samples are
all in same class

Pseudocode for Decision Tree Induction Revisited

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;

bestAtt = **chooseBestAtt**(samples,atts);

label N with bestAtt;

FOR each value a_i of bestAtt //each branch from node N

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;

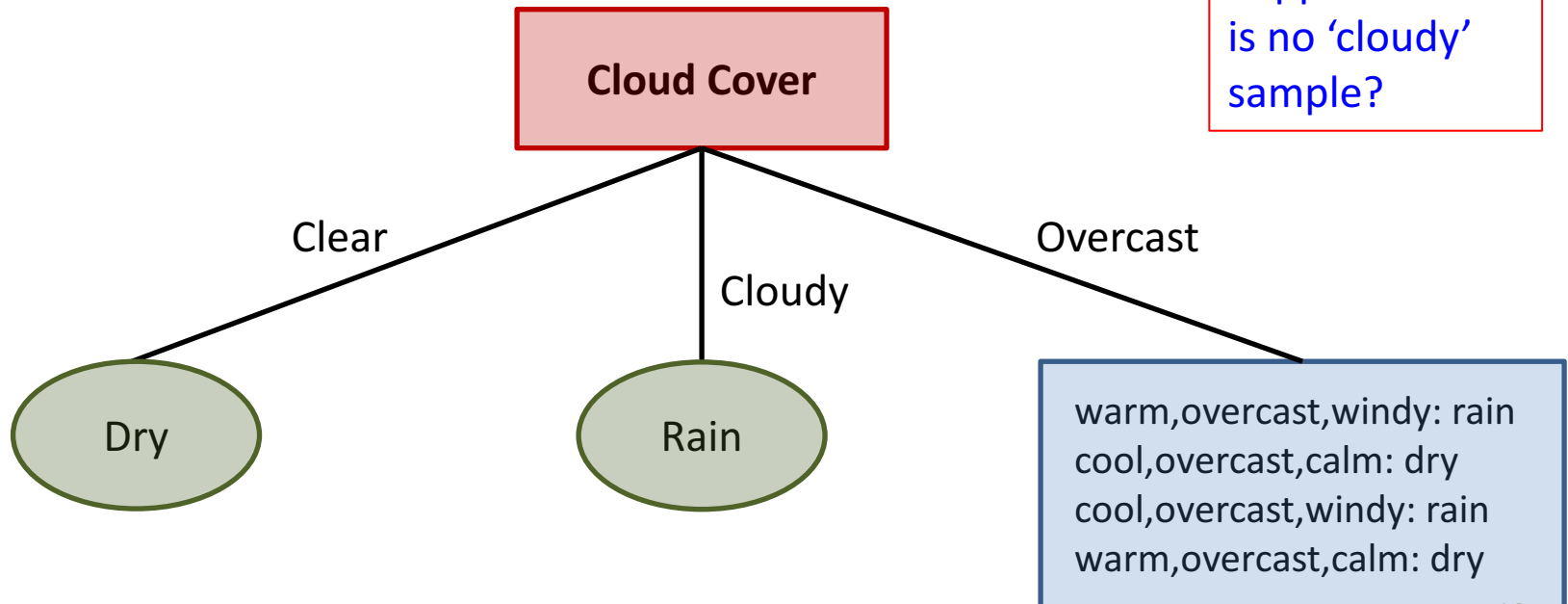
An Example ... (7)

All the samples on the “Clear” branch or subset belong to the same class, i.e., dry, so no further expansion is needed.

It can be terminated with a leaf node labelled “dry” (see pseudo code)

Similarly, the single sample on the “Cloudy” branch or subset necessarily belongs to one class, i.e., rain.

It can be terminated with a leaf node labelled “rain”.



An Example ... (8)

Extending the Overcast subtree (see pseudo code)

The “Overcast” branch has both rain and dry samples.

So we must attempt to extend the tree from this node.

There are **4 samples in this subset**: 2 rain and 2 dry

So for this subset, $p(\text{rain}) = p(\text{dry}) = 0.5$ and the initial information is 1 bit.

There are two remaining attributes: temperature and wind. Which is the best?

An Example ... (9)

Information Gain from attribute Temperature:

Cool samples: 2 out of 4

There are 2 of these; 1 rain and 1 dry.

So for this **sub-subset**, $p(\text{rain}) = 1/2$ and $p(\text{dry}) = 1/2$

Hence, $\text{Inf}_{\text{cool}} = -(1/2) \times \log_2(1/2) - (1/2) \times \log_2(1/2) = 1 \text{ bit}$

Warm samples: 2 out of 4

There are also 2 of these; 1 rain and 1 dry.

So again for this **sub-subset**, $p(\text{rain}) = 1/2$ and $p(\text{dry}) = 1/2$

Hence, $\text{Inf}_{\text{warm}} = -(1/2) \times \log_2(1/2) - (1/2) \times \log_2(1/2) = 1 \text{ bit}$

Average **Information about the class given value of Temperature:**

$$(1/2) \times \text{Inf}_{\text{cool}} + (1/2) \times \text{Inf}_{\text{warm}} = 0.5 \times 1.0 + 0.5 \times 1.0 = 1 \text{ bit}$$

Hence, Information Gain from Temperature is zero!

(Does it make good sense? – examine the 4 samples)

An Example ... (10)

Information Gain from attribute Wind:

Windy samples: 2 out of 4

There are 2 of these; 2 rain and 0 dry.

So for this **sub-subset**, $p(\text{rain}) = 1$ and $p(\text{dry}) = 0$

Hence, $\text{Inf}_{\text{windy}} = -1 \times \log_2(1) - 0 \times \log_2(0) = 0$

$$0 \times \log_2(0) = 0$$

Calm samples: 2 out of 4

There are also 2 of these; 0 rain and 2 dry.

So again $\text{Inf}_{\text{calm}} = -1 \times \log_2(1) - 0 \times \log_2(0) = 0$

Average Information about the class given value of Wind:

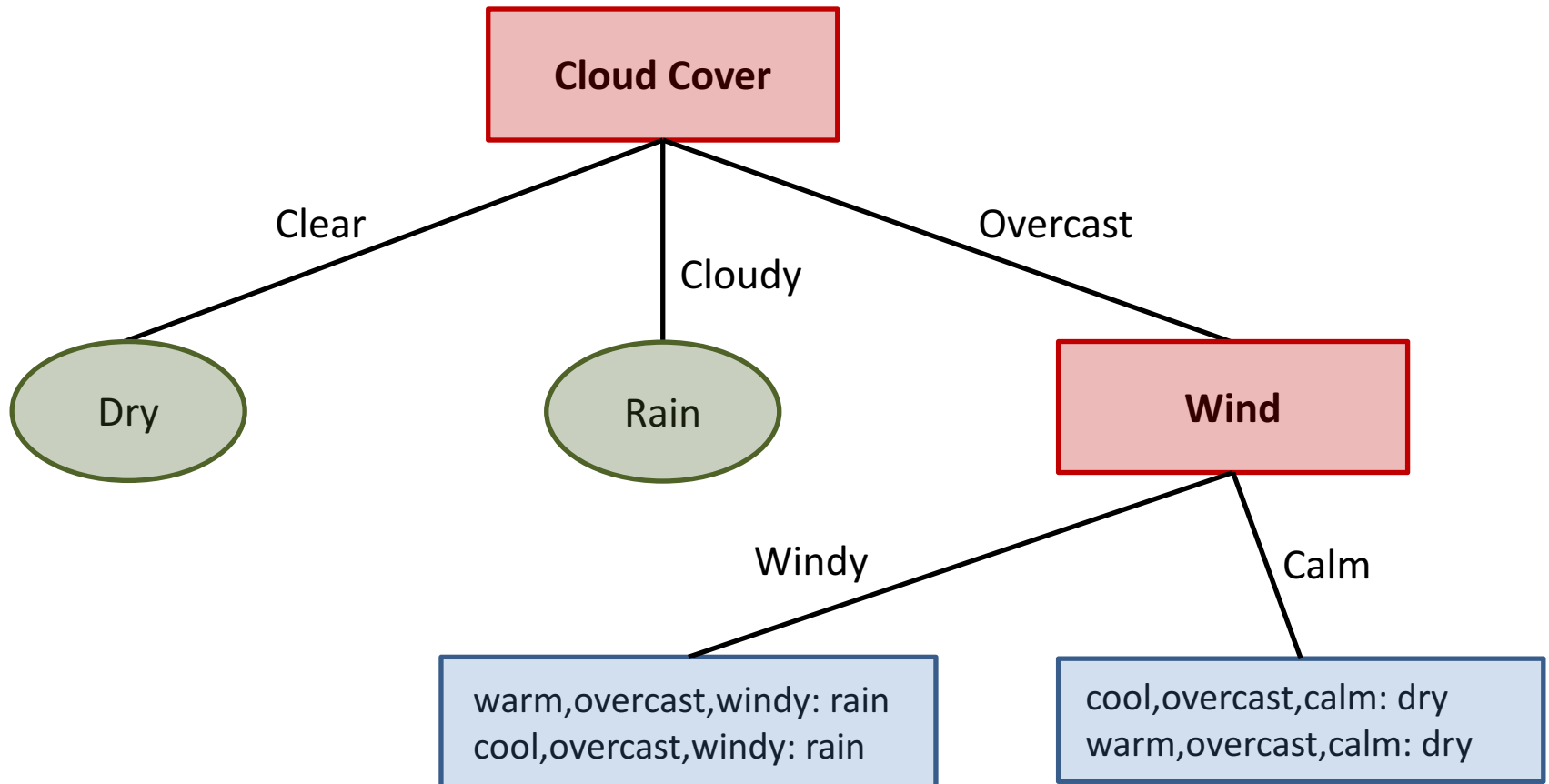
$$(1/2) \times \text{Inf}_{\text{windy}} + (1/2) \times \text{Inf}_{\text{calm}} = 0.5 \times 0 + 0.5 \times 0 = 0$$

Hence, Information Gain from Wind is 1.

Note: This reflects the fact that wind is a perfect predictor of precipitation for this sub-subset of samples.

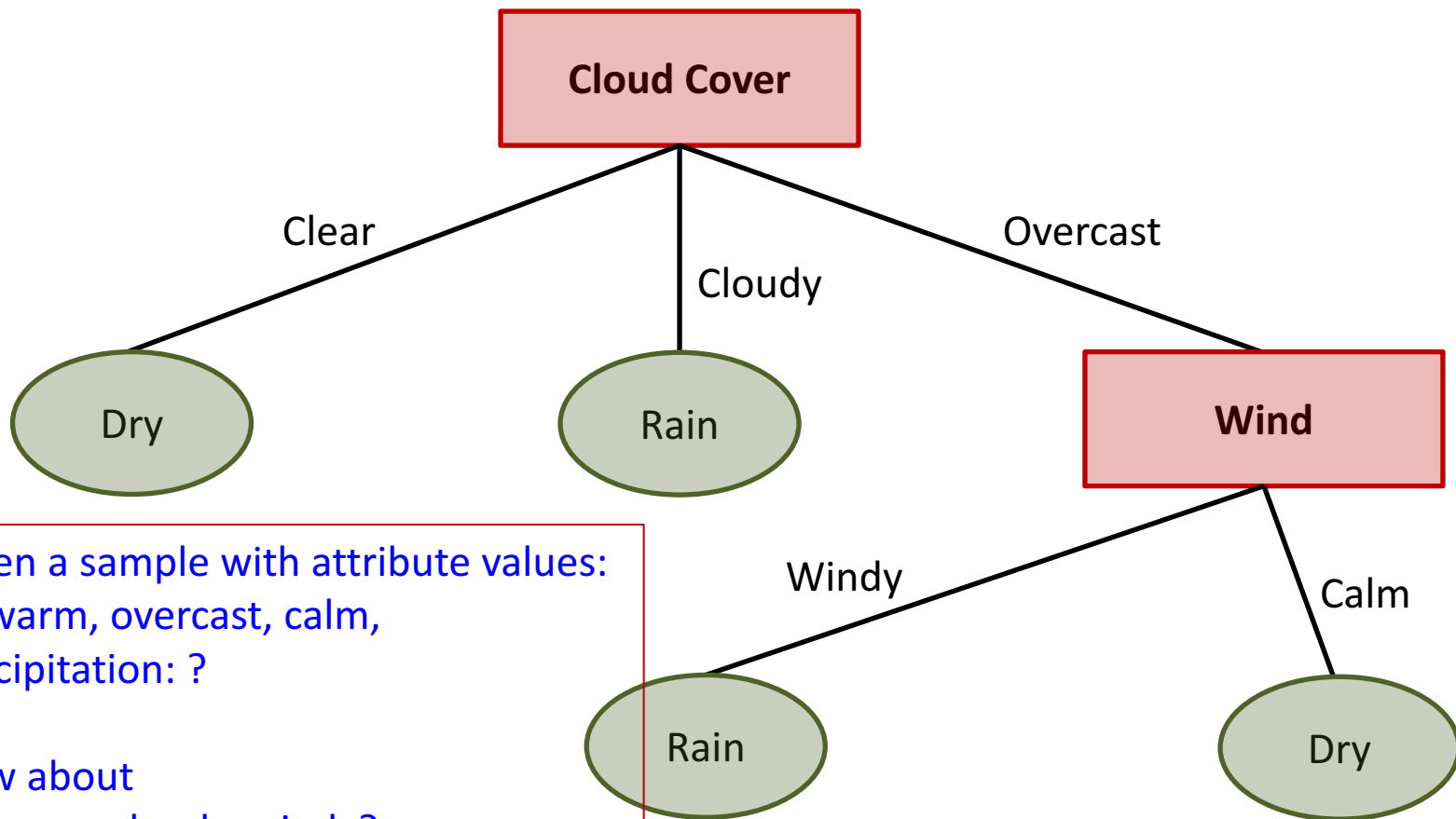
An Example ... (11)

Obviously wind is the best attribute, so we can now extend the tree as follows:



An Example ... (12)

All the samples on both the new branches belong to the same class, so they can be terminated with appropriately labelled leaf nodes.




Given a sample with attribute values:
warm, overcast, calm,
Precipitation: ?

How about
warm, cloudy, windy?

From Decision Trees To Production Rules

Decision trees can easily be converted into sets of IF-THEN rules as follows:



```
IF clear THEN dry
IF cloudy THEN rain
IF overcast AND calm THEN dry
IF overcast AND windy THEN rain
```

(knowledge discovery)

Such rules are usually easier to understand than the corresponding decision tree.

Large trees produce large sets of rules.

It is often possible to simplify these rule sets considerably.

In some cases these simplified rule sets are more accurate than the original tree because they reduce the effect of overfitting – a topic we will discuss later.

Is it necessary to have conflict resolution strategies here?

Which rule interpreter is better, forward chaining or backward chaining?

Some Issues in Decision Tree Induction

1) Inconsistent / Contradictory Training Data

It is possible (and not unusual) to arrive at a situation in which the samples associated with a leaf node belong to **more than one class**, because there are ***no more attributes*** available or useful to further subdivide the samples.

A simple method for handling this is to label the leaf node with the modal class. This means that some training samples will be misclassified by the induced decision tree.

Some Issues in Decision Tree Induction (2)

2) Numeric Attributes

The number of possible attribute values can be very large, creating too many branches in the decision tree.

A simple solution is to partition the value into a small number of contiguous subranges and then treat the membership of each subrange as a **categorical variable**, e.g., small, medium, large.

Some Issues in Decision Tree Induction (3)

3) Overfitting

Question

What would happen if a set of completely random data were used for a decision tree induction program?

Answer

The program would build a decision tree.

If there were many variables and plenty of data, it could be a large tree.

Question

Would the decision tree be any good as a classifier?

Would it do better than the simple strategy of always picking the modal class?

Answer

No, because random dataset contains no useful information for classification.

Note that if a new set of random data were used we would get an entirely different decision tree.

Some Issues in Decision Tree Induction (4)

3) Overfitting (continued)

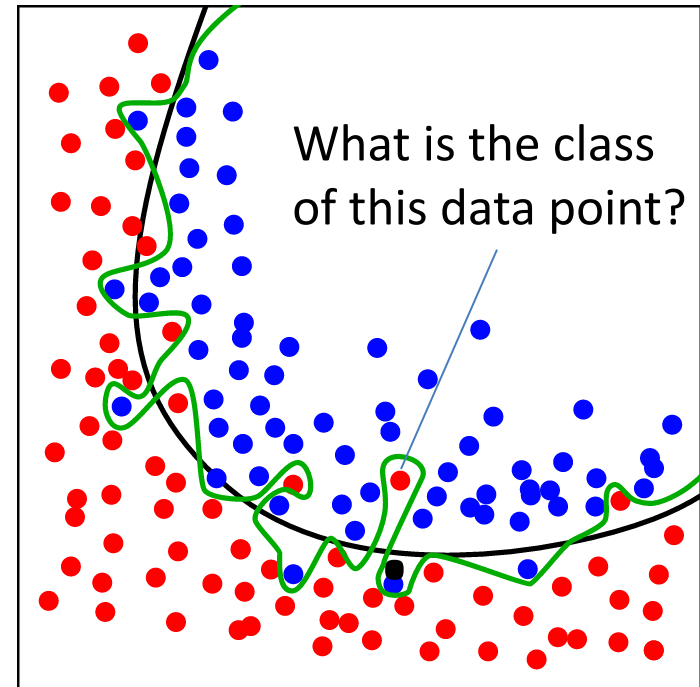
Question

Could the same sort of thing happen with non-random data?

Answer

Yes, when the training data is small, noisy, or corrupted, the deduced decision tree will fit the noise.

An example of overfitting with non-random data: see the figure.



The samples nearby separation boundary could be mislabelled due to noise.

The separation boundary in green is bad, which is a good example of overfitting.

Some Issues in Decision Tree Induction (5)

3) Overfitting (continued)

What are the reasons for overfitting?

A decision tree is a mathematical model of some *population* of samples, but the tree is built on the basis of *a small fraction of the population* – the training dataset. What a decision tree induction program really does is building *a model of the training dataset*.

The induced decision tree could reflect relationships that are *true for the population as a whole*, when the training dataset is representative, but it may only reflect relationships that *are peculiar to the particular training dataset*, when the training dataset is too small or of poor quality (e.g., the example in this lecture, with one ‘cloudy’ sample only).

This phenomenon of modelling the training data rather than the population it represents is called *overfitting*. The reasons for overfitting include that the training data is not representative or the model is too flexible or powerful.

Some Issues in Decision Tree Induction (6)

3) Overfitting (continued)

How to combat overfitting?

- by better model and/or better training data

There are two basic ways of reducing or preventing overfitting in decision tree induction:

- Stop tree growth before it happens: ***Pre-pruning***.
- Remove parts of the tree due to overfitting after it has been constructed: ***Post-pruning***.

The pruning methods are beyond the scope of this module.

Of course, increasing the number of high-quality training samples (if available/possible) is a more direct approach to overfitting prevention.

Summary

The Weather Data Example of Decision Tree Induction

- Steps in the pseudo code

- Procedure for choosing best attributes using information gain

From Decision Trees to Production Rules

- Knowledge discovery from data by machine learning

Some Issues in Decision Tree Induction

- Inconsistent Data

- Numeric Attributes

- Overfitting (to be addressed again in Neural Networks)