20T1: COMP9417 Machine Learning and Data Mining

Lectures: Tree Learning
Topic: Questions from lectures
Version: with answers

Introduction

Some questions and exercises from the course lectures covering aspects of supervised tree learning for classification and regression.

Expressiveness of decision trees

Question 1 Give decision trees to represent the following Boolean functions, where the variables `A`, `B`, `C` and `D` have values `t` or `f`, and the class value is either `True` or `False`:

a) $A \wedge \neg B$

b) $A \vee [B \wedge C]$

c) $A \;\; X\text{OR} \;\; B$

d) $[A \wedge B] \vee [C \wedge D]$

Can you observe any effect of the increasing complexity of the functions on the form of their expression as decision trees ?

Answer

a) $A \wedge \neg B$:

```
A = t:
|   B = f:  True
|   B = t:  False
A = f:  False
```

b) $A \vee [B \wedge C]$

```
A = t:  True
A = f:
|   B = f:  False
|   B = t:
|   |   C = t:  True
|   |   C = f:  False
```

c) A XOR B

```
A = t:
|   B = t:  False
|   B = f:  True
A = f:
|   B = t:  True
|   B = f:  False
```

d) $[A \wedge B] \vee [C \wedge D]$

```
A = t:
|   B = t:  True
|   B = f:
|   |   C = t:
|   |   |   D = t:  True
|   |   |   D = f:  False
|   |   C = f:  False
A = f:
|   C = t:
|   |   D = t:  True
|   |   D = f:  False
|   C = f:  False
```

Notice the *replication* effect of repeated subtrees as the target expression becomes more complex, for example, in the tree for *d*. This is a situation where the hypothesis class of models (here, decision trees) can fit any Boolean function, but in order to represent the function the tree may need to be very complex. This makes it hard to learn, and will require a lot data !

Decision tree learning

Question 2 Here is small dataset for a two-class prediction task. There are 4 attributes, and the class is in the rightmost column. Look at the examples. Can you guess which attribute(s) will be most predictive of the class ?

species	**rebel**	**age**	**ability**	**homeworld**
pearl	yes	6000	regeneration	no
bismuth	yes	8000	regeneration	no
pearl	no	6000	weapon-summoning	no
garnet	yes	5000	regeneration	no
amethyst	no	6000	shapeshifting	no
amethyst	yes	5000	shapeshifting	no
garnet	yes	6000	weapon-summoning	no
diamond	no	6000	regeneration	yes
diamond	no	8000	regeneration	yes
amethyst	no	5000	shapeshifting	yes
pearl	no	8000	shapeshifting	yes
jasper	no	6000	weapon-summoning	yes

You probably guessed that attributes 3 and 4 were not very predictive of the class, which is true. However, you might be surprised to learn that attribute "species" has higher information gain than attribute "rebel". Why is this ? Refer to slides 37-38 on "Attributes with Many Values" in the lecture notes.

Suppose you are told the following: for attribute "species" the Information Gain is 0.52 and *Split Information* is 2.46, whereas for attribute "rebel" the Information Gain is 0.48 and *Split Information* is 0.98.

Which attribute would the decision-tree learning algorithm select as the split when using the *Gain Ratio* criterion instead of Information Gain ? Is Gain Ratio a better criterion than Information Gain in this case ?

Answer

Gain Ratio corrects for a bias in Information Gain that favours attributes with many values by dividing it by the Split Information, which is the information of the partition of the data according to the values of the attribute in question. This is shown in the formula on slide 38 for Split Information of some attribute A on a sample S, where c is the number of values for A.

For attribute "species" Gain Ratio is $\frac{0.52}{2.46} \simeq 0.21$, whereas for attribute "rebel" Gain Ratio is $\frac{0.48}{0.98} \simeq 0.49$.

So attribute "rebel" would be selected under the Gain Ratio criterion.

In this case, it is a better choice, since the "species" attribute has higher information gain simply by having many more values in this dataset. Note however that other corrections to information gain have been proposed that have other advantages in different settings, and in general there is no "best" splitting criterion for all settings.

Question 3 Assume we learn a decision tree to predict class Y given attributes A, B and C from the following training set, with no pruning.

A	B	C	Y
0	0	0	0
0	0	1	0
0	0	1	0
0	1	0	0
0	1	1	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	0	1
1	1	1	0
1	1	1	1

What would be the training set error for this dataset ? Express your answer as the number of examples out of twelve that would be misclassified.

Answer

2. There are two pairs of examples with the same values for attributes A, B and C but a different (contradictory) value for class Y. One example from each of these pairs will always be misclassified (noise).

Question 4 One nice feature of decision tree learners is that they can learn trees to do *multi-class* classification, i.e., where the problem is to learn to classify each instance into exactly one of $k > 2$ classes.

Suppose a decision tree is to be learned on an arbitrary set of data where each instance has a discrete class value in one of $k > 2$ classes. What is the maximum training set error, expressed as a fraction, that any dataset could have ?

Answer

First consider the case where all k class values are evenly distributed, so there are $\frac{1}{k}$ examples of each class in the training set. In the worst case a decision tree can be learned that predicts one of the k classes for all examples. This will get $\frac{1}{k}$ of the training set correct, and make $k - 1 \times \frac{1}{k}$ mistakes, or $\frac{k-1}{k}$ as a fraction of the training set.

If any one class has more than $\frac{1}{k}$ examples then the worst case decision tree is guaranteed to predict that class, which will reduce the error since that class now represents more than $\frac{1}{k}$ of the training set.
