COMP3308/3608 Artificial Intelligence

Week 7 Tutorial exercises
Decision trees

***Exercise 1*. Entropy (Homework)**

What is the information content (entropy) of a message telling the outcome of the flip of:

a) an honest dice?
b) a dice that has been rigged to come up six 50% of the time?

You may find this table useful:

x	y	-(x/y)* log$_2$(x/y)	x	y	-(x/y)* log$_2$(x/y	x	y	-(x/y)* log$_2$(x/y	x	y	-(x/y)* log$_2$(x/y
1	2	0.50	4	5	0.26	6	7	0.19	5	9	0.47
1	3	0.53	1	6	0.43	1	8	0.38	7	9	0.28
2	3	0.39	5	6	0.22	3	8	0.53	8	9	0.15
1	4	0.5	1	7	0.40	5	8	0.42	1	10	0.33
3	4	0.31	2	7	0.52	7	8	0.17	3	10	0.52
1	5	0.46	3	7	0.52	1	9	0.35	7	10	0.36
2	5	0.53	4	7	0.46	2	9	0.48	9	10	0.14
3	5	0.44	5	7	0.35	4	9	0.52			

Answer:

a) H(dice toss) = I(P_{one}, P_{two}, P_{three}, P_{four}, P_{five}, P_{six}) = I(1/6, 1/6, 1/6, 1/6, 1/6, 1/6)=6*(-1/6 log$_2$(1/6))=- log$_2$(1/6) = 2.585 bits

b) P_{six}=1/2; P_{one}=P_{two}=P_{three}=P_{four}=P_{five}=1/2*1/5=1/10
H(dice toss) = -5(1/10)log$_2$(1/10)-(1/2)log$_2$(1/2)= 2.16 bits

***Exercise 2*. Information gain**

Consider the following set of training examples:

a1	a2	class
T	T	+
T	T	+
T	F	-
F	F	+
F	T	-
F	T	-

a) What is the entropy of this collection of training examples with respect to the class?
b) What is the information gain of a2?
c) Which attribute will be selected as the first attribute to split on?
d) Draw the decision tree after the selection of the attribute from the previous step. Indicate the child node(s) that: 1) need to be further expanded by adding additional tests and 2) can be converted to leafs, and show their class.

Answer:

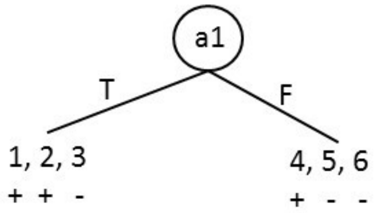
a) H(S)=I(3/6,3/6)=-3/6 log(3/6)-3/6 log(3/6) = 1 bit

b) Split on a2:
$H(S_T)=I(2/4,2/4)=-2/4 \log(2/4)-2/4 \log(2/4)=0.5 + 0.5 = 1$ bit
$H(S_F)=I(1/2,1/2)=-1/2 \log(1/2)-1/2 \log(1/2)= 0.5 + 0.5 = 1$ bit
$H(S|a2)=4/6*1 +2/6*1=1$ bit
$gain(a2)=1-1=0$ bits

c) The attribute with the highest information gain will be selected. This means that we need to calculate the information gain of all attributes. In b) we already calculated the information gain for a2. Now let's do this for a1:
$H(S_T)=I(2/3,1/3)=-2/3 \log(2/3)-1/3 \log(1/3)= 0.39 + 0.53 = 0.92$ bits
$H(S_F)=I(1/3,2/3)=-1/3 \log(1/3)-2/3 \log(2/3)= 0.53 + 0.29 = 0.92$ bits
$H(S|a1)=3/6*0.92 +3/6*0.92=0.92$ bits
$gain(a1)=1-0.92=0.08$ bits

gain(a1)>gain(a2) = > a1 will be selected as the root of the DT (the first attribute to split on)

d) The DT after the selection of the first attribute is:



It is not possible to create leaf nodes as the resulting partitions of the dataset are not "pure" – they contain examples from different classes. Hence, both child nodes need to further expanded by selecting another attribute to split on. (In the figure above the numbers correspond to the example number, i.e. 1 is example 1 from the table, etc.)

Exercise 3. ***Decision trees (Advanced only)***

In the recursive construction of DTs, it sometimes occurs that a mixed set of positive and negative examples remains at a leaf node, even after all the attributes have been used. Consider binary classification and suppose that we have p positive examples and n negative examples at a leaf node and there are no more attributes left. Show that returning the probability p/(p+n) minimizes the sum of squared errors of the set of these examples.

Hint: The error on a given training set can be written as a mathematical expression and viewed as a function of what is returned (i.e. the class chosen - in this case a single number 0 or 1 at the leaf as the task is binary classification) and the number of positive and negative examples, i.e. :

Let's x be the returned class (x=0 or 1). The error function E depends on x, p and n: E=f(x,p,n).
The absolute error on the set of examples is: $E = p(1-x) + nx$. Let's check that it makes sense:

- if x=0 (negative class is returned), the n negative examples will be classified correctly but the p positive examples will be misclassified => E=p
- if x=1 (positive class is returned), the p positive examples will be classified correctly but the n negative examples will be misclassified => E=n

The sum of the squared errors is: $E2=p(1-x)^2+nx^2$.
Show that E2 is minimized for x=p/(n+p).

Answer:
Once we have the expression for E2, it is very easy to find the minimum value.

$E2=p(1-x)^2+nx^2$
$dE2/dx=2p(1-x)(-1)+2nx=-2p+2px+2nx=2x(n+p)-2p$
$d^2E2/dx=2(n+p)>0$ as $p>0$ and $n>0$ =>minimum
$2x(n+p)-2p=0$ => $x=p/(n+p)$
=> E2 is minimized for $x=p/(n+p)$

Exercise 4. Using Weka
1. Load iris data (iris.arff). Choose evaluation on "training set".
2. Run the decision tree (j48) classifier under "trees. In Weka there are 2 versions: id3 which is only for nominal attributes and j48 for both numeric and nominal. "

Look at the decision tree that is produced, do you understand it?

```
Test mode: evaluate on training data

=== Classifier model (full training set) ===

J48 pruned tree
------------------

petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)

Number of Leaves  :     5

Size of the tree :  9
```

This is the pruned DT. Weka uses pruning by default.

a) Draw the DT.
b) As you can see in some paths we test the same attribute more than once. Why? If the attributes are nominal, we never test the same attribute twice along one path.
c) What are the number in brackets, e.g. (48.0/1.0)?

3. Change the evaluation to "Cross validation" (10 folds). Run ZeroR, OneR, Naïve Bayes, 3-nearest neighbor (IBL-3) and the decision tree (j48.) classifiers.

a) Compare the classification accuracies. Which is the most accurate classifier?
b) What output does each of the classifiers generate (e.g. a set of rules, etc.)? Suppose that you have to explain to your client why a particular decision is taken for a new example by these classifiers. The output of which classifier(s) are easier to understand and use for decision making?

Answer:

2b) In DTs with nominal attributes, the attribute tests divide the examples according to the attribute value. Therefore, any example reaching the second test already has a known value for the attribute and the second test is redundant. In DTs with numeric values the tests split the values of the attribute into intervals, so additional tests of the same attribute can be used to redefine the interval of values (e.g.

petalwidth <= 1.7 and then petalwidth <= 1.5 redefines the interval to petalwidth <= 1.5; or petalwidth > 0.6 and then petalwidth <= 1.7 will define the interval 0.6< petalwidth <= 1.

2c) Notice that these numbers are associated with leaf nodes. The first one is the total number of examples that follow the path to the leaf and the second one is the number of these examples that were misclassified. So (48.0/1.0) means that 48 examples reach the leaf, 47 of them are correctly classified and 1 is incorrectly classified.

3b) Decision trees (which are a set of mutually exclusive rules) are considered to be easier to understand by humans and easier to use for decision making that the output of other algorithms, e.g. Naïve Bayes, nearest neighbor and especially neural networks and support vector machine classifiers that we will study later.

Additional exercises (to be done at your own time)

***Exercise 5.* – Information gain**

Consider the following data from credit history of loan application.

credit history	debt	collateral	income	risk
bad	high	none	0-15k	high
unknown	high	none	15-35k	high
unknown	low	none	15-35k	moderate
unknown	low	none	0-15k	high
unknown	low	none	over 35k	low
unknown	low	adequate	over 35k	low
bad	low	none	0-15k	high
bad	low	adequate	over 35k	moderate
good	low	none	over 35k	low
good	high	adequate	over 35k	low
good	high	none	0-15k	high
good	high	none	15-35k	moderate
good	high	none	over 35k	low
bad	high	none	15-35k	high

You may find this table useful:

x	y	-(x/y)* log$_2$(x/y)	x	y	-(x/y)* log$_2$(x/y	x	y	-(x/y)* log$_2$(x/y	x	y	-(x/y)* log$_2$(x/y	x	y	-(x/y)* log$_2$(x/y
1	2	0.50	4	5	0.26	6	7	0.19	5	9	0.47	2	11	0.45
1	3	0.53	1	6	0.43	1	8	0.38	7	9	0.28	3	11	0.51
2	3	0.39	5	6	0.22	3	8	0.53	8	9	0.15	4	11	0.53
1	4	0.5	1	7	0.40	5	8	0.42	1	10	0.33	5	11	0.52
3	4	0.31	2	7	0.52	7	8	0.17	3	10	0.52	6	11	0.48
1	5	0.46	3	7	0.52	1	9	0.35	7	10	0.36	7	11	0.42
2	5	0.53	4	7	0.46	2	9	0.48	9	10	0.14	8	11	0.33
3	5	0.44	5	7	0.35	4	9	0.52	1	11	0.31	9	11	0.24

x	y	-(x/y)* log$_2$(x/y)	x	y	-(x/y)* log$_2$(x/y	x	y	-(x/y)* log$_2$(x/y
10	11	0.13	4	13	0.52	12	13	0.11
1	12	0.30	5	13	0.53	1	14	0.27
5	12	0.53	6	13	0.51	3	14	0.48

7	12	0.45	7	13	0.48	5	14	0.53
11	12	0.12	8	13	0.43	9	14	0.41
1	13	0.28	9	13	0.37	11	14	0.27
2	13	0.42	10	13	0.29	13	14	0.10
3	13	0.49	11	13	0.20			

a) What is the amount of information (entropy) in the above set of examples with respect to the target classification?

b) What is the information gain when using the attribute credit history to predict the risk?

c) Which attribute will be selected as the first attribute to split on? Why?

Solution:
a) H(S)=I(6/14, 3/14, 5/14)=-6/14log(6/14)-3/14log(3/14)-5/14(log5/14)=1.52 bits
where S is the given set of examples

b) split based on *credit history*:

H(S$_{bad}$)=I(0/4, 1/4, 3/4)=-0/4log$_2$(0/4)-1/4log$_2$(1/4)-3/4log$_2$(3/4)=0.81 bit
H(S$_{unknown}$)=I(2/5, 1/5, 2/5)=-2/5log$_2$(2/5)-1/5log$_2$(1/5)-2/5log$_2$(2/5)=1.5 bits
H(S$_{good}$)=I(3/5, 1/5, 1/5)=-3/5log$_2$(3/5)-1/5log$_2$(1/5)-1/5log$_2$(1/5)=1.34 bits
H(S|credit history)=4/14*0.81+5/14*1.5+5/14*1.34=1.24 bits
gain(credit history)=1.52-1.24=0.27 bits

c) split based on *debt*:
H(S$_{high}$)=I(2/7, 1/7, 4/7)=-2/7log$_2$(2/7)-1/7log$_2$(1/7)-4/7log$_2$(4/7)=1.38 bit
H(S$_{low}$)=I(3/7, 2/7, 2/7)=-3/7log$_2$(3/7)-2/7log$_2$(2/7)-2/7log$_2$(2/7)=1.56 bits
H(S|debt)=7/14*1.38+7/14*1.56=1.47 bits
gain(debt)=1.52-1.47=0.05 bits

split based on *collateral*:
H(S$_{none}$)=I(6/11, 2/11, 3/11)=-6/11 log$_2$(6/11)-2/11 log$_2$(2/11)-3/11 log$_2$(3/11)=1.4bit
H(S$_{adequate}$)=I(0/3, 1/3, 2/3)=-0/3log$_2$(0/3)-1/3log$_2$(1/3)-2/3log$_2$(2/3)=0.9 bits
H(S|collateral)=11/14*1.4+3/14*0.9=1.33 bits
gain(collateral)=1.52-1.33=0.19 bits

split based on *income*:
H(S$_{0-15}$)=I(4/4, 0/4, 0/4)=-4/4 log$_2$(4/4)-0/4 log$_2$(0/4)-0/4 log$_2$(0/4)=0 bit
H(S$_{15-35}$)=I(2/4, 2/4, 0/4)=-2/4log$_2$(2/4)-2/4log$_2$(2/4)-0/4log$_2$(0/4)=1 bit
H(S$_{over35}$)=I(0/6, 1/6, 5/6)=-0/6log$_2$(0/6)-1/6log$_2$(1/6)-5/6log$_2$(5/6)=0.65 bits
H(S|income)=4/14*0+4/14*1+6/14*0.65=0.56 bits
gain(income)=1.52-0.56=0.96 bits

Hence, the most informative attribute is *income*, followed by *credit history*, *collateral* and *dept.* The split will be on *income* as it has the highest information gain.