Techniques of Artificial Intelligence

Exercises – Decision Trees

Dipankar Sengupta
Dipankar.Sengupta@vub.ac.be

Roxana Rădulescu
Roxana.Radulescu@vub.ac.be

March 14, 2016

Representation and Interpretation of Boolean Functions

Symbol	**Name**
0	FALSE
1	TRUE
A	A
B	B
$!A/\neg A$	NOT B
$!B/\neg B$	NOT A
$A \wedge B$	A AND B
$A \vee B$	A OR B
$\neg A \wedge B$	NOT A AND B
$A \wedge \neg B$	A AND NOT B
$\neg A \vee B$	NOT A OR B
$A \vee \neg B$	A OR NOT B
$A \oplus B$	A XOR B
$A \bar{\vee} B$	A NOR B
A XNOR B	A XNOR B
$A \bar{\wedge} B$	A NAND B

13. **Decision Trees** (Exercise 3.1 from the course book)
Give decision trees to represent the following Boolean functions:

(a) $A \wedge \neg B$

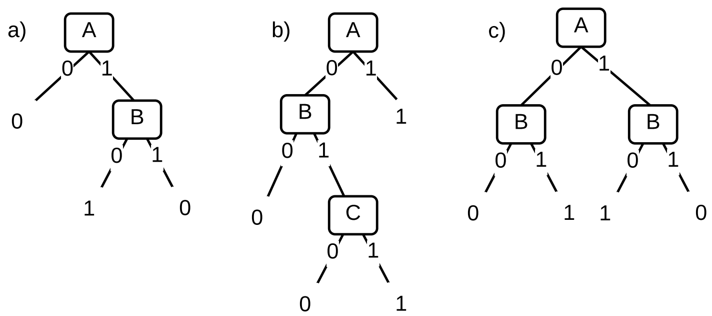
(b) $A \vee (B \wedge C)$

(c) $A \oplus B$

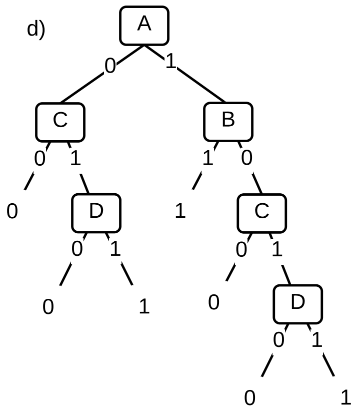
(d) $(A \wedge B) \vee (C \wedge D)$

(e) A XNOR B (hint: XNOR $= (A \wedge B) \vee (\neg A \wedge \neg B)$)

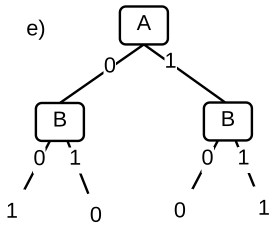
Answer:



d)



e)



14. **Decision Trees**
True or false: any Boolean function can be expressed as a decision tree. Give a proof.

Answer:

True: a proof by induction can be constructed.

If p and q are atomic logical expressions, then the trivial tree p' for p (or q' for q) is a tree consisting of a single node testing for p (or q).

If p and q are logical expressions with corresponding decision trees p' and q', then decision trees for $\neg p$, $p \wedge q$ and $p \vee q$ can be constructed as follows.

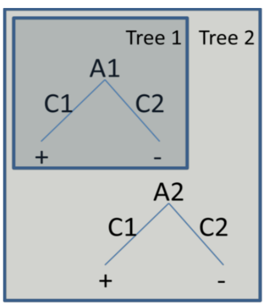
- For $\neg p$: Take the decision tree p' and change all labels $+$ to $-$ and the other way around.
- For $p \wedge q$: Take the decision tree p', replace all $+$ labels by the entire q' tree.
- For $p \vee q$: Take the decision tree p', replace all $-$ labels by the entire q' tree.

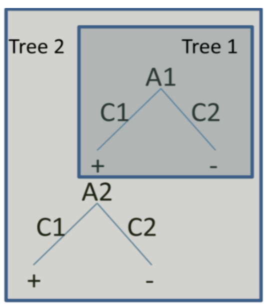
15. **More general than** (Exercise 3.3 from Mitchel)
True or False: If a decision tree D2 is an elaboration of tree D1, then D1 is more general than D2. Assume D1 and D2 are decision trees representing arbitrary Boolean functions, and that D2 is an elaboration of D1 if ID3 could extend D1 into D2. If true, give a proof; if false, give a counterexample.

More-general-than was defined in chapter two: Let h_j and h_k be Boolean-valued functions defined over X. Then h_j is more general than or equal to h_k if and only if $(\forall x \in X)[(h_k(x) = 1 \rightarrow (h_j(x) = 1)]$

Answer:

False: Consider the case shown at the left: Tree 2 is an elaboration of Tree 1. Tree 1 is extended by extending the negative branch. Tree 2 is more general than Tree 1 (all examples classified positive by Tree 1 are also classified positive by Tree 2, but not the other way around).





Consider the case shown at the right: Tree 2 is an elaboration of Tree 1. Tree 1 is extended by extending the positive branch. Tree 2 is more specific than Tree 1 (all examples classified positive by Tree 2 are also classified positive by Tree 1, but not the other way around).

The statement is not true as any tree extending a positive branch will make the original tree more specific.

16. **ID3**

In order to evaluate the quality of the tree which is grown by ID3, one could compare its performance to a baseline performance. The baseline performance is often the performance of a very simple machine learning algorithm. Consider the following approach:

You have a dataset consisting of 25 examples of two classes. You plan to use leave-one-out cross validation. As a baseline, you use a simple majority classifier (a majority classifier is given a set of training data and then always outputs the class that is in the majority in the training set, regardless the input). Such a majority classifier is expected to score about 50%, but with this example of leave-one-out cross-validation, it does not. What will be its performance and why?

Answer: The entire set contains 25 positive examples and 25 negative examples. With leave one out cross validation, you randomly select 49 examples for training and one for testing. Suppose the test instance should be classified positive. In that case the training set contains 24 positive and 25 negative examples. The majority voter hence classifies the instance as negative, which is wrong. A similar reasoning holds for a negative test instance. Hence, the performance of the classifier is always zero in this case.

17. **Decision Trees with ID3**

Consider the following data on the hair color, body weight, body height and the usage of lotion of eight different people. The table shows whether the people got sunburned after an afternoon in the sun.

Hair	**Height**	**Weight**	**Lotion**	**Sunburned?**
Blonde	Average	Light	No	Yes
Blonde	Tall	Average	Yes	No
Brown	Short	Average	Yes	No
Blonde	Short	Average	No	Yes
Red	Average	Heavy	No	Yes
Brown	Tall	Heavy	No	No
Brown	Average	Heavy	No	No
Blonde	Short	Light	Yes	No

(a) Perform average entropy calculations on the following complete dataset for each of the four attributes. Select the attribute which minimizes the entropy; draw the first level of the decision tree.

(b) Grow the tree until you reach the proper identification of all the samples.

(c) Establish the rules from the tree found in the previous question.

(d) The factual value of the training instance, in the dataset is: **Sunburned?** – **Yes**. Consider the following additional instance in **your** dataset:

Hair	**Height**	**Weight**	**Lotion**	**Sunburned?**
Red	Tall	Average	Yes	No

Will the inclusion of this new instance in your training dataset impact the structure of the decision tree? What are such instances considered to be? Does their presence or absence in the training dataset impact the outcome of the built model?

(e) Consider the herewith provided alternative dataset:

Hair	Height	Weight	Lotion	Sunburned?
?	Average	Light	No	Yes
Blonde	Tall	Average	Yes	No
Brown	Short	Average	Yes	No
Blonde	Short	Average	No	Yes
Red	Average	Heavy	No	Yes
Brown	Tall	Heavy	No	No
Brown	Average	?	No	No
Blonde	Short	Light	?	No

Is this dataset suitable for training your model and will you be able to build an optimal decision tree? Explain briefly how you can deal with this situation.

Additional question Consider the data set given below. Explain how one can handle a continuous-valued attribute when building a decision tree and demonstrate the approach on the given data.

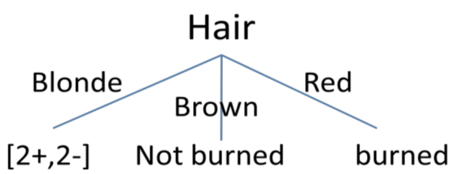
Temperature	25	17	22	10	20	5	13
PlayTennis	Yes	No	Yes	No	Yes	No	No

Answer:

(a) The entropy of the entire example set is given by:
$Entropy(S) = -\frac{3}{8}log_2\frac{3}{8} - \frac{5}{8}log_2\frac{5}{8} = 0.954$
For constructing the first level of the tree, we seek for the attribute with the biggest gain.

- For Hair:
  - $Entropy(Hair, blonde) = [2+, 2-]$(2 negative examples, two positive examples) $= -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$
  - $Entropy(Hair, brown) = [0+, 3-] = 0$
  - $Entropy(Hair, red) = [1+, 0-] = 0$
  - $Gain(S, Hair) = 0.954 - \frac{4}{8} \times 1 - \frac{3}{8} \times 0 - \frac{1}{8} \times 0 = 0.454$
- For Height:
  - $Entropy(Height, average) = [2+, 1-] = -\frac{2}{3}log_2\frac{2}{3} - -\frac{1}{3}log_2\frac{1}{3} = 0.918$
  - $Entropy(Height, tall) = [2+, 0-] = 0$
  - $Entropy(Height, short) = [2+, 1-] = 0.918$
  - $Gain(S, Height) = 0.954 - \frac{3}{8} \times 0.918 - \frac{2}{8} \times 0 - \frac{3}{8} \times 0.918 = 0.266$
- For Weight:
  - $Entropy(Weight, light) = [1+, 1-] = 1$
  - $Entropy(Weight, average) = [2-, 1+] = 0.918$
  - $Entropy(Weight, heavy) = [2+, 1-] = 0.918$
  - $Gain(S, Weight) = 0.954 - \frac{2}{8} \times 1 - \frac{3}{8} \times 0.918 - \frac{3}{8} \times 0.918 = 0.016$
- For Lotion:
  - $Entropy(Lotion, no) = [3+, 2-] = 0.971$
  - $Entropy(Lotion, yes) = [3-, 0+] = 0$
  - $Gain(S, Lotion) = 0.954 - \frac{5}{8} \times 0.971 - \frac{3}{8} \times 0 = 0.347$

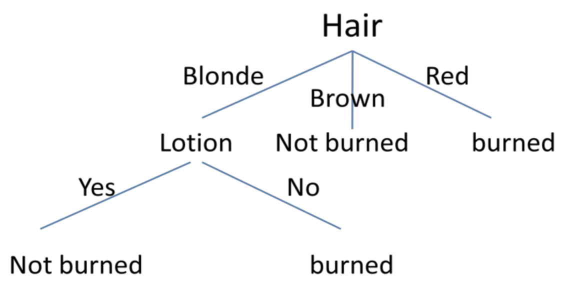
Hence, the attribute with the biggest gain is "Hair", the first node in the tree tests for Hair:



(b) So, in case the color of the hair is blonde, the tree must be elaborated further. S is now restricted to the four examples where the color of the hair is blonde.
$Entropy(S) = [2+, 2-] = 1$

- For Height:
  - $Entropy(Height, average) = [1+, 0-] = 0$
  - $Entropy(Height, tall) = [1-, 0+] = 0$
  - $Entropy(Height, short) = [1+, 1-] = 1$
  - $Gain(S, Height) = 1 - \frac{2}{4} \times 1 = 0.5$
- For Weight:
  - $Entropy(Weight, light) = [1+, 1-] = 1$
  - $Entropy(Weight, average) = [1-, 1+] = 1$
  - $Entropy(Weight, heavy) = [0+, 0-] = 0$
  - $Gain(S, Weight) = 1 - \frac{1}{2} \times 1 - \frac{1}{2} \times 1 = 0$
- For Lotion:
  - $Entropy(Lotion, no) = [0+, 2-] = 0$
  - $Entropy(Lotion, yes) = [2+, 0-] = 0$
  - $Gain(S, Lotion) = 1 - 0 - 0 = 1$

The attribute with the biggest gain is "Lotion", so we grow the tree as follows:



(c) Rules:
- If (Hair = blonde and lotion = yes) then burned = false
- If (Hair = blonde and lotion = no) then burned = true
- If (Hair = brown) then burned = false
- If (Hair = red) then burned = true

(d) Dealing with noisy data

(e) Dealing with missing attribute values

(f) Handling continuous-valued attributes