Mitchell's Book: Ex. 3.1 DTs

Machine Learning 2024-25 Course Activity

Furno Francesco - francesco.furno@studenti.unipd.it - 2139507

November 21, 2024

Table of contents

1. Exercise 3.1 from Mitchell's book

Give decision trees to represent the following boolean functions:

1. $A \wedge \neg B$
2. $A \vee [B \wedge C]$
3. $A \oplus B$
4. $[A \wedge B] \vee [C \wedge D]$

1.1. $A \wedge \neg B$

$A \wedge \neg B$ is a very simple boolean function: for this reason, logical reasoning is sufficient to compute its decision tree.

The following is the truth table for this particular boolean function:

A	B	$\neg B$	$A \wedge \neg B$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Table 1: $A \wedge \neg B$ truth table

We can easily observe that the behavior of the function is strictly related to A. In fact, the function is always 0 when $A = 0$. Regarding B, we can see that the function is always 0 when $B = 1$.

Let's choose label A for the root node. The computed DT is the following:

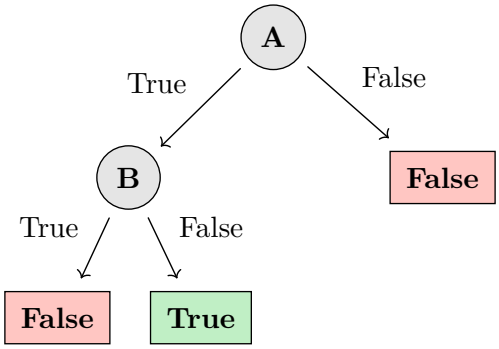

Figure 1: DT-1 of $A \wedge \neg B$

When A is False, we can immediately conclude that the value of the function is False. Otherwise, we need to check the value of B to make a final conclusion.

If we choose B as label for the root node, the computed DT is the following:

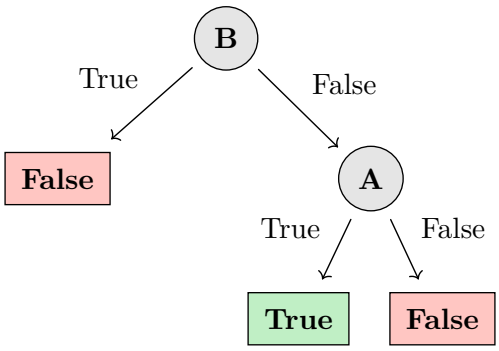

Figure 2: DT-2 of $A \wedge \neg B$

DT-1 and DT-2 have the same performance over the boolean function $A \wedge \neg B$.

1.2. $A \vee [B \wedge C]$

$A \vee [B \wedge C]$ is a simple boolean function: for this reason, logical reasoning is sufficient to compute its decision tree.

The following is the truth table for this boolean function:

A	B	C	$B \wedge C$	$A \vee [B \wedge C]$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 2: $A \vee [B \wedge C]$ truth table

Again, we can easily observe that the behavior of the function is strictly related to A. In fact, the function is always 1 when $A = 1$. Regarding B and C, we cannot make strong conclusions

independently, so we understand that the most "important" variable here is A, which will serve as the label for the root node.

The computed DTs are the following, both with the same performance:

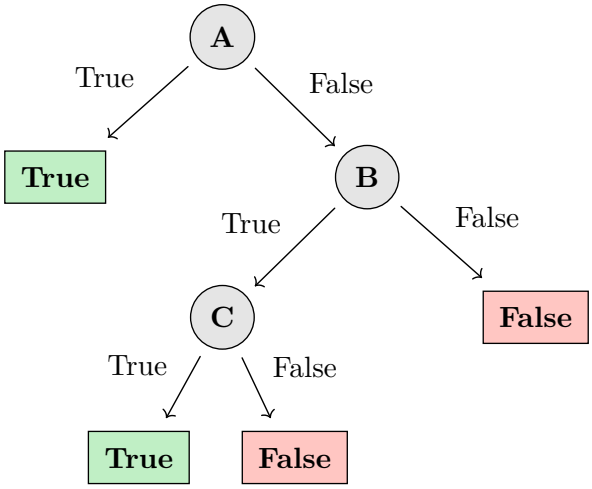

Figure 3: DT-1 of $A \vee [B \wedge C]$

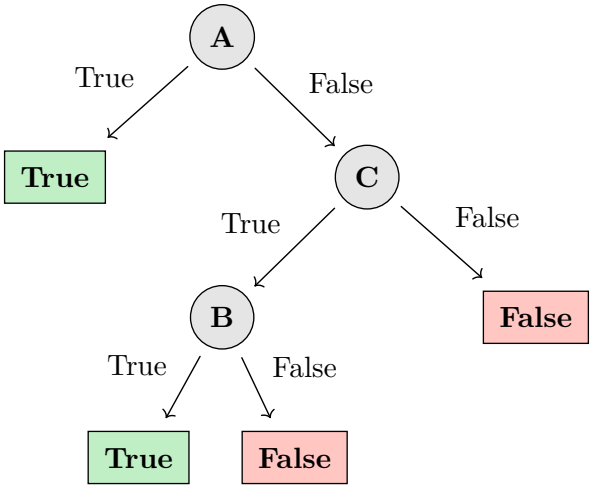

Figure 4: DT-2 of $A \vee [B \wedge C]$

As we can see, we can choose both B or C as label for the second node of the second level. In fact, when $A = 0$:

- for $B = 0$, the function produces 2 negative outputs,
- for $B = 1$, the function produces 1 positive output and 1 negative output,
- for $C = 0$, the function produces 2 negative output,
- for $C = 1$, the function produces 1 positive output and 1 negative output;

in fact, B and C have the same *Information Gain* for $A = 0$.

1.3. $A \oplus B$

$A \oplus B$ is another simple boolean function: for this reason, logical reasoning is sufficient to compute its decision tree. However, we can use also some tools as *Entropy* and *Information Gain*.

The following is the truth table for this particular boolean function:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Table 3: $A \oplus B$ truth table

Here, we can observe that the behavior of the function is related to A and B in the same way:

- for $A = 0$, the function produces 1 positive output and 1 negative output (entropy = 1),
- for $A = 1$, the function produces 1 positive output and 1 negative output (entropy = 1);

B behaves in the same way, in fact A and B have the same Information Gain.

Entropy quantifies the uncertainty of a dataset. In the case of $A \oplus B$, when we fix either A or B, the outputs are split evenly between positive and negative, leading to an entropy of 1, which indicates maximum uncertainty.

Information Gain is used to decide the best variable to split the data, aiming to reduce entropy. Since A and B provide the same Information Gain for the XOR function, either variable can be chosen as the root without affecting the decision tree's performance.

The computed DTs are the following, both with the same performance:

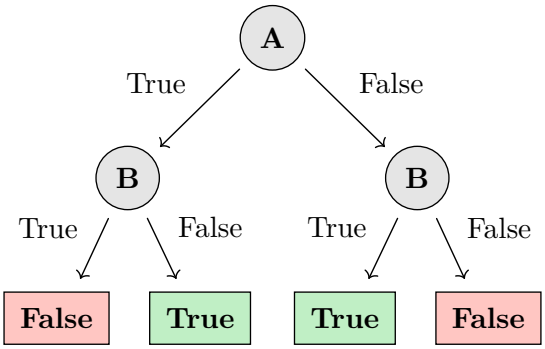

Figure 5: DT-1 of $A \oplus B$

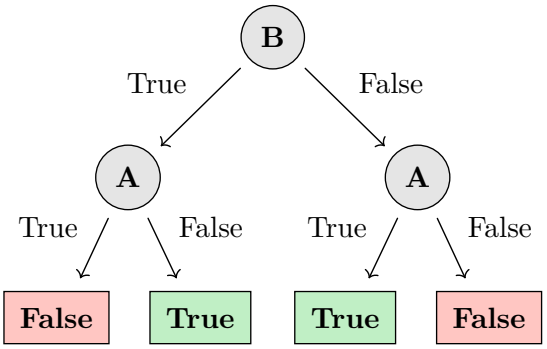

Figure 6: DT-2 of $A \oplus B$

1.4. $[A \wedge B] \vee [C \wedge D]$

$[A \wedge B] \vee [C \wedge D]$ is a simple boolean function: for this reason, logical reasoning is sufficient to compute its decision tree.

The following is the truth table for this boolean function:

A	B	C	D	$A \wedge B$	$C \wedge D$	$[A \wedge B] \vee [C \wedge D]$
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1

A	B	C	D	$A \wedge B$	$C \wedge D$	$[A \wedge B] \vee [C \wedge D]$
1	1	1	0	1	0	1
1	1	1	1	1	1	1

Table 4: $[A \wedge B] \vee [C \wedge D]$ truth table

First of all we should check A:

- if $A = 0$, we can skip B and evaluate $C \wedge D$,
- if $A = 1$, we should check B:
  - if $B = 0$, we need to evaluate $C \wedge D$,
  - if $B = 1$, we conclude that the function outputs True whatever C and D are.

If $B = 0$ or $A = 0$, we need to evaluate C and D. We can start from C, again:

- if $C = 0$, we can skip D and conclude that the function outputs False whatever D is,
- if $C = 1$, we need to evaluate D:
  - if $D = 0$, we conclude that the function outputs False,
  - if $D = 1$, we conclude that the function outputs True.

The computed DT is the following:

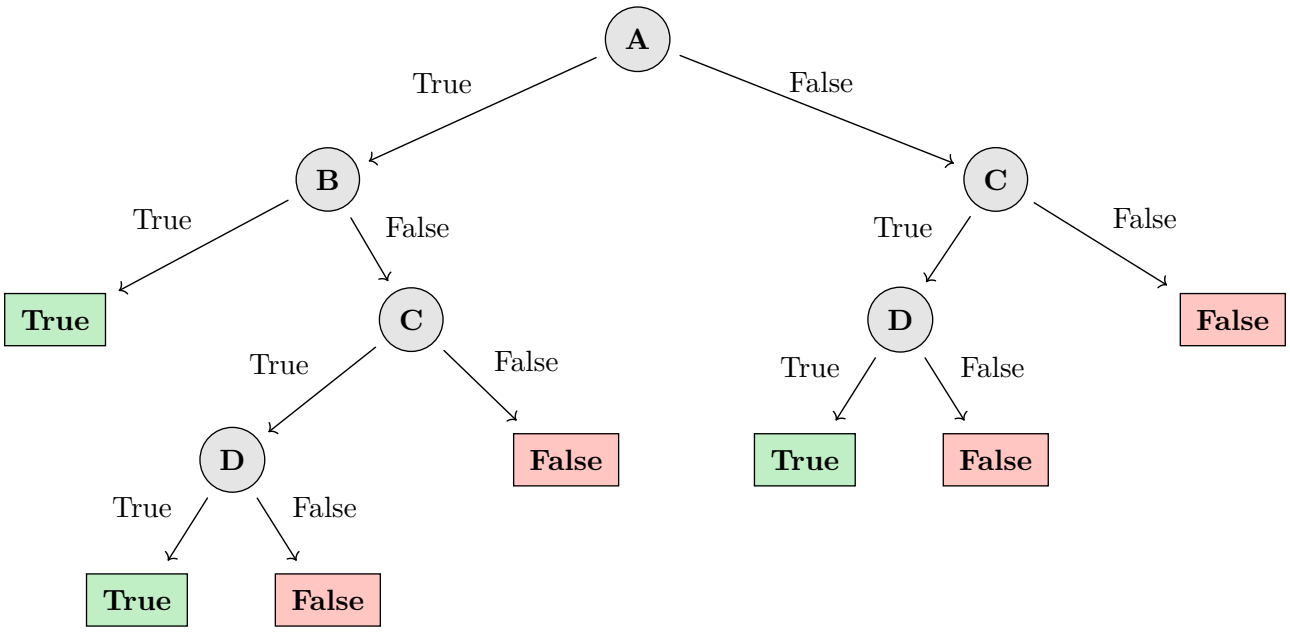

Figure 7: DT of $[A \wedge B] \vee [C \wedge D]$