Completed

# Principles of Machine Learning

# Exercises

Victor Verreet *victor.verreet@cs.kuleuven.be*
Laurens Devos *laurens.devos@cs.kuleuven.be*

Fall, 2021

## Exercise Session 1: Decision Trees and $k$-NN

### 1.1 Logical Formulas as Decision Trees

Let the left subtree be the *true branch*. Draw decision trees for the following formulas.

a. $A \wedge \neg B$

b. $\neg A \vee B$

c. $A \vee (B \wedge C)$

d. $(\neg A \wedge B) \vee (A \wedge \neg B)$

e. $(A \vee B) \wedge (C \vee D \vee \neg E)$

f. $(A \vee B \vee C) \wedge (D \vee E \vee F)$

### 1.2 The Decision Surface of a Tree

Consider a data set with two *numeric* attributes $a_1$ and $a_2$ and one nominal target attribute $c$ with two possible values: $\oplus$ and $\ominus$. The training examples are shown in Figure 1.

a. Find a decision tree that classifies all training examples correctly.

b. Draw the decision surface of this tree in Figure 1.

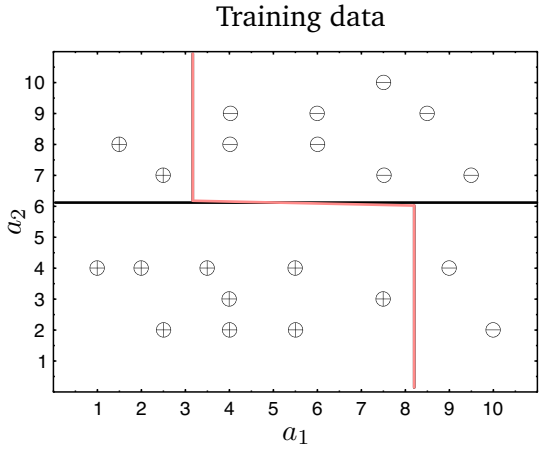



Figure 1: Training data for question 1.2.

### 1.3 Measuring Test Quality: Entropy, Information Gain, and Variance Reduction

Consider the following table of training examples:

| Instance | $a_1$ | $a_2$ | $y_1$ | $y_2$ |
|---|---|---|---|---|
| 1 | T | T | $\oplus$ | 1.0 |
| 2 | T | T | $\oplus$ | 1.0 |
| 3 | T | F | $\ominus$ | 1.0 |
| 4 | F | F | $\oplus$ | 5.0 |
| 5 | F | T | $\ominus$ | 6.0 |
| 6 | F | T | $\ominus$ | 5.5 |

a. What is the entropy with respect to the $y_1$ attribute?

b. What is the information gain of splitting on attribute $a_2$ (easy) and $a_1$ (use calculator) when $y_1$ is the target attribute? Which attribute is the best choice and why?

c. What is the variance reduction of splitting on $a_1$ and $a_2$ when $y_2$ is the target attribute?

d. When would you use information gain and when would you use variance reduction?

### 1.4 Constructing a Tree: ID3

The ID3 algorithm is a name for the *top-down induction of decision tree (TDIDT)* algorithm that uses *information gain* as a heuristic.

| Instance | **Sky** | **AirTemp** | **Humidity** | **Wind** | **Water** | **Forecast** | **EnjoySport** |
|---|---|---|---|---|---|---|---|
| 1 | sunny | warm | normal | strong | warm | same | yes |
| 2 | sunny | warm | high | strong | warm | same | yes |
| 3 | rainy | cold | high | strong | warm | change | no |
| 4 | sunny | warm | high | strong | cool | change | yes |
| 5 | sunny | warm | normal | weak | warm | same | no |

a. Construct a decision tree that correctly classifies the **EnjoySport** attribute using only instances 1-4 from the table above. Specifically, think of ways in which the positive instances can be separated from the negative instances by formulating split conditions on the other attributes.

b. We will now simulate the ID3 algorithm using all 5 instances.

Compute the class entropy for the entire dataset $S$:

| | $+$ | $-$ | Entropy |
|---|---|---|---|
| $S$ | 3 | 2 | |

Compute the split heuristic for each attribute and select the attribute for the root node:

| Attribute | Values | $\oplus$ | $\ominus$ | Entropy | IG |
|---|---|---|---|---|---|
| **Sky** | sunny | 3 | 1 | 0.811 | 0.32 |
| | rainy | 0 | 1 | 0.0 | |
| **AirTemp** | warm | | | | |
| | cold | | | | |
| **Humidity** | normal | 1 | 1 | 1.0 | 0.02 |
| | high | 2 | 1 | 0.918 | |
| **Wind** | strong | | | | |
| | weak | | | | |
| **Water** | warm | | | | |
| | cool | | | | |
| **Forecast** | same | | | | |
| | change | | | | |

Manually execute the ID3 algorithm until you obtain a complete decision tree.

### 1.5 Predicting Probabilities with Decision Trees

Formulate a strategy that uses a decision tree to define a conditional probability distribution of the discrete target attribute. For example, for the dataset in the previous exercise, the algorithm should not just predict a class (i.e., *yes* or *no*), but rather it should predict a probability value (e.g. 80% *yes*, and 100%-80%=20% *no*).

What would be the advantages and disadvantages of this approach?

### 1.6 Instance-Based Learning: $k$-NN

Instance based learning consists of storing examples and comparing new instances with the examples, taking the target values of the examples nearest to the new instance to compute a prediction for the new instance.

Instance based classification requires a distance measure. When data can be represented as n-dimensional vectors with numeric or symbolic components, a generally usable distance function is

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_j w_j \cdot \delta(x[j], y[j])^2},$$

where $x[j]$ is the $j$th attribute value of instance $x$. For symbolic values, $\delta$ is defined by $\delta(x, y) = 0$ if $x = y, 1$ otherwise; and for numerical values, $\delta(x, y) = |x - y|$.

Note that for purely numerical vectors, with $w_i = 1$ we obtain the Euclidean distance, whereas for purely symbolic vectors the same weights give us the square root of the Hamming distance (square root is monotonically increasing, so relative ordering is unaffected).

In this exercise the instance space is $\{\text{true}, \text{false}\} \times \mathbb{R}$. Unless specified otherwise, use the function mentioned above as the distance function.

Consider the following points.

| Instance | A | B | Class |
|---|---|---|---|
| 1 | true | 1 | $\oplus$ |
| 2 | false | 3 | $\oplus$ |
| 3 | true | 5 | $\oplus$ |
| 4 | false | 2 | $\ominus$ |

Use $w_j = 1$ for the first three tasks:

a. Classify (false, 4) with nearest-neighbors or 1-NN.

b. Classify (false, 4) with 3-NN.

c. Classify (false, 4) using the prototypes $\boldsymbol{p}_\oplus$ = (true, 3, $\oplus$) and $\boldsymbol{p}_\ominus$ = (false, 2, $\ominus$).

d. Repeat c. but *min-max*-normalize attribute **B** and use the following weights for each attribute:

$$w_j = 1 - \frac{1}{n} \sum_{\boldsymbol{p} \in \{\oplus, \ominus\}} \sum_{x \in \boldsymbol{p}} \delta(\boldsymbol{p}[j], x[j]),$$

with $\boldsymbol{p}[j]$ the $j$th component value of prototype $\boldsymbol{p}$. What is the interpretation of these weights?

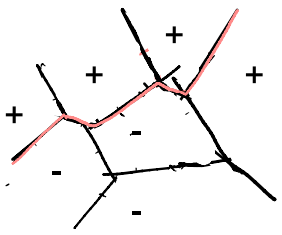
## 1.7 Visualizing 1-NN: Voronoi maps

Sketch a Voronoi map for the 1-NN method for the following examples, and indicate the decision surface that results.

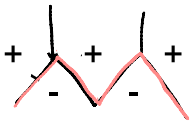
What would the decision surface look like if you used the prototype based approach (replace the set of positive examples by a single example lying in the middle of them, same for the negatives)?

Think about some other differences between the prototype based approach and storing each individual example.

(a)




(b)




## 1.8 Curse of Dimensionality

a. The $k$-NN algorithm's performance deteriorates as the number of attributes increases. Why is this true? What are possible solutions to this problem? Are decision trees affected in a similar way?

b. With a lot of attributes, spurious correlations can become a problem. How do $k$-NN and decision trees (mis)-use spurious correlations? What information could you use to determine whether a decision tree is affected by spurious correlations?

## 1.9 Programming Exercise: Experimenting with the Effects of Normalization and Irrelevant Features on $k$-NN

You can find a Python script `mnist_knn.py` and a dataset `mnist.h5` on Toledo. Download both files and store them in the same folder. Make sure you have the following Python packages installed:

- `numpy`
- `pandas`
- `tables` (support for the HDF5 data format)
- `scikit-learn`

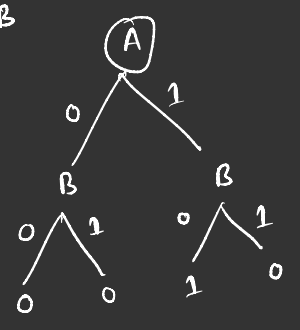
- `matplotlib`

You can use `pip` or `Conda` for installing Python packages. On Linux, you likely can install these packages using your package manager.

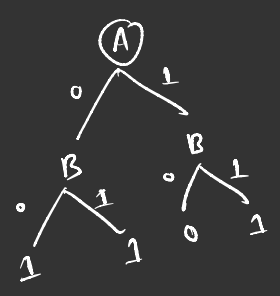
Answer the questions in the Python file.

1.1

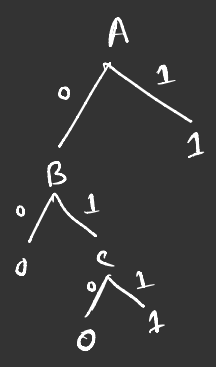
a) $A \wedge \neg B$



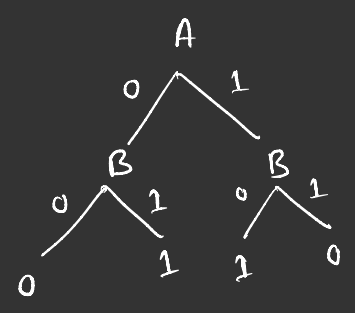

b) $\neg A \vee B$



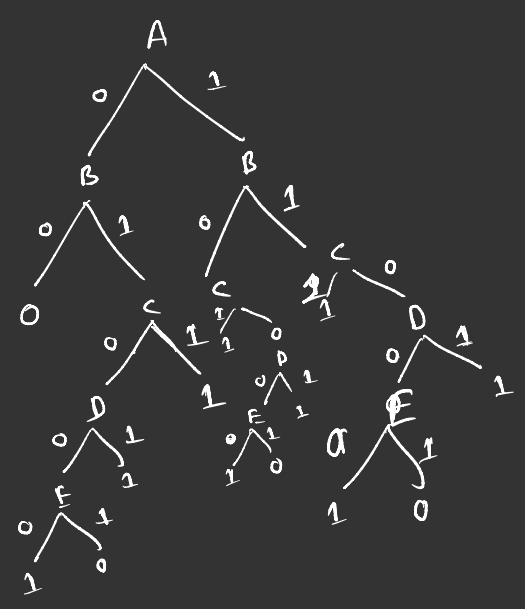

c) $A \vee (B \wedge C)$



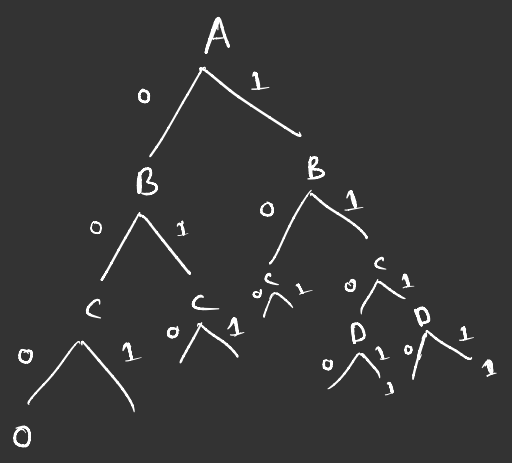

d) $(\neg A \wedge B) \vee (A \wedge \neg B)$



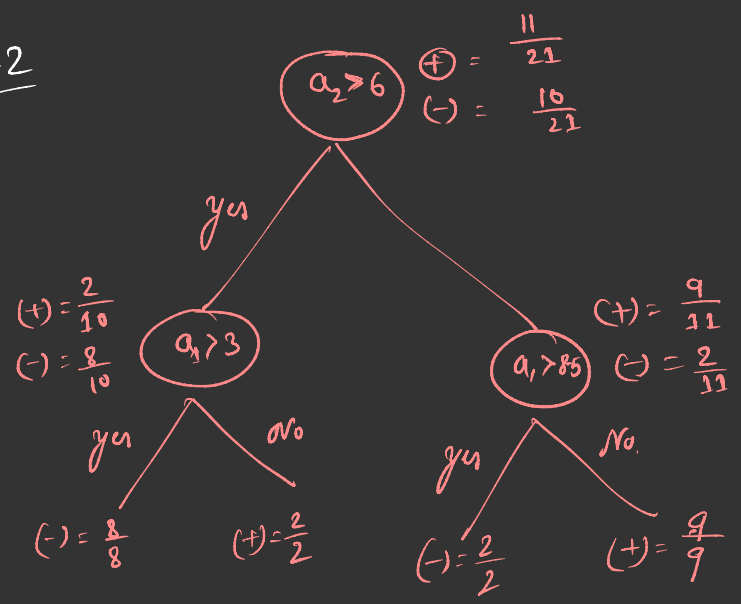

e) $(A \vee B) \wedge (C \vee D \vee \neg E)$




f) $(A \vee B \vee C) \wedge (D \vee E \vee F)$




1.2



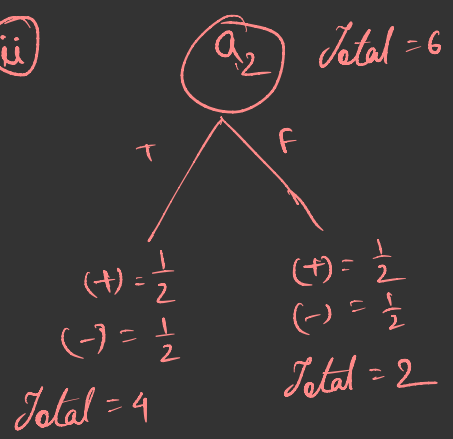

1.3

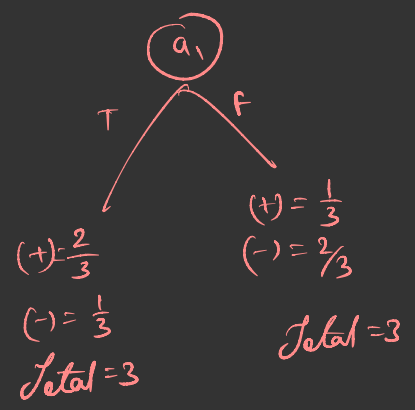
i) $(+) = \frac{1}{2}$, $(-) = \frac{1}{2}$

$$Ent = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}$$

$$= -\log\frac{1}{2} = 1$$

ii)




$$ENT_{a_2} = \frac{4}{6}E(1,1) + \frac{2}{6}E(1,1)$$

$$= \frac{4}{6} + \frac{2}{6} = 1$$

$$ENT_{a_1} = E(2,1) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.918$$

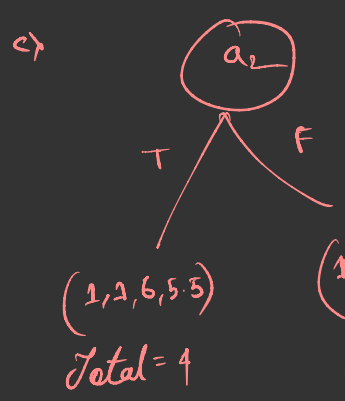
$$IG = ENT - ENT(a_2)$$

$$= 0$$

$$I'_G = ENT - ENT(a_1)$$

$$= 1 - 0.918 = 0.082$$

c)




$$mean(S) = \frac{19.5}{6}$$

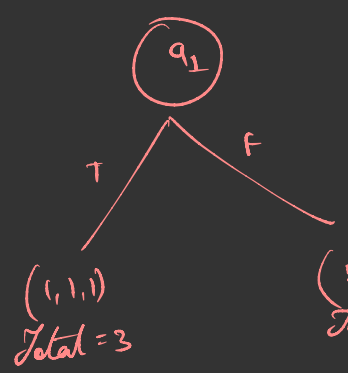
$$Var(S) = \frac{3\left(1 - \frac{19.5}{6}\right)^2 + \left(5 - \frac{19.5}{6}\right)^2 + \left(6 - \frac{19}{6}\right)^2 + \left(5.5 - \frac{19.5}{6}\right)^2}{5}$$

$$= 6.17$$

$$Var_{a_2} = \frac{4}{6}Var(S_1) + \frac{2}{6}Var(S_2)$$

$$= 7.70$$

Variance reduction = −1.53



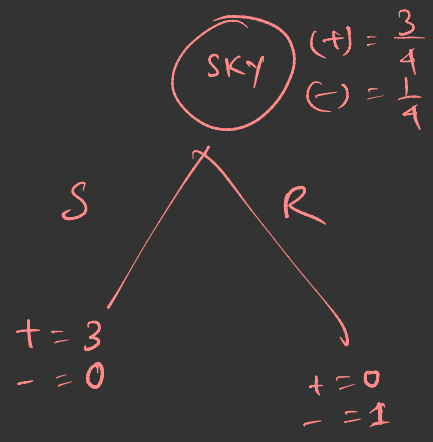

$$Var(a_1) = \frac{1}{2}Var(S_1) + \frac{1}{2}Var(S_2)$$

$$= 0.125$$

Information gain: Categorical variable

Variance gain: Numerical variable

1.4



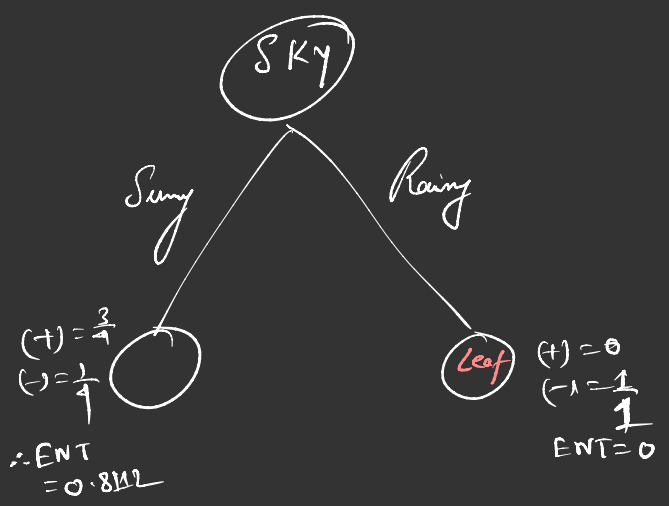

$ENT = -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4}$

$= 0.81$

$ENT_{Sky} = 0$

$IG(Sky) = 1$

## Choose Sky as root node



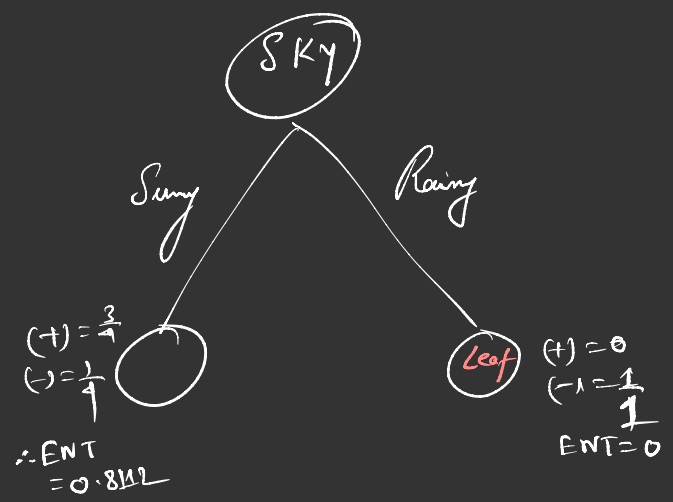

AirTemp → Warm → yes

→ Cold → no

ENT = 0.97

b)

| attribute | | (+) | (-) | Total | Entropy | Total entropy | IG |
|---|---|---|---|---|---|---|---|
| Sky | Sunny | 3 | 1 | 4 | 0.8112 | $\frac{4}{5}$ ENT(3,1) | 0.97 - 0.65 |
| | Rainy | 0 | 1 | 1 | 0 | $\frac{1}{5}$ ET(0,1) | = 0.32 |
| | | | | | | = 0.65 | |
| AT | Warm | 3 | 1 | 4 | 0.8112 | 0.65 | 0.32 |
| | Cold | 0 | 1 | 0 | 0 | | |
| Hum | normal | 1 | 1 | 2 | 1 | $\frac{2}{5}1 + \frac{3}{5}$ 0.9m | 0.02 |
| | high | 2 | 1 | 3 | 0.918 | = 0.951 | |
| Wind | Strong | 3 | 1 | 4 | 0.8112 | 0.65 | 0.32 |
| | weak | 0 | 1 | 1 | 0 | | |
| Mat | warm | 2 | 2 | 4 | 1 | 4/5 = 0.8 | 0.17 |
| | cool | 1 | 0 | 1 | 0 | | |
| Forecast | Same | 2 | 1 | 3 | 0.918 | 0.951 | 0.02 |
| | change | 1 | 1 | 2 | 1 | | |




| Att | | (+) | (-) | Total | ENT | TENT | IG |
|---|---|---|---|---|---|---|---|
| AT | W | 3 | 1 | 4 | 0.8112 | 0.8112 | 0 |
| | C | 0 | 0 | 0 | 0 | | |
| H | N | 1 | 1 | 2 | 1 | 0.5 | 0.3112 |
| | H | 2 | 0 | 2 | 0 | | |
| W | S | 3 | 0 | 3 | 0 | 0 | 0.8112 |
| | W | 0 | 1 | 1 | 0 | | |

1.5

① Use the same splitting criteria.
② Count instances in every leaf.

1.6

$x_0 = (0, 4)$

$d(x_0, x_1) = \sqrt{1+9} = \sqrt{10}$

$d(x_0, x_2) = 1$

$d(x_0, x_3) = 1+1 = \sqrt{2}$

$d(x_0, x_3) = \sqrt{0+4} = 2$

$(1, 1, +)$
$(0, 3, +)$
$(1, 5, +)$
$(0, 2, -)$

a) +

b) 2(+), 1(−), → (+)

c) $d(x_0, P\oplus) = \sqrt{1+1} = \sqrt{2}$

$d(x_0, P\ominus) = \sqrt{4} = 2$

d) $P_S(+) = (\text{true}, 0.5) = (t, 0.5) = (1, 0.5)$

$S = \left(f, \frac{4-1}{4}\right) = (f, 0.75) = (0, 0.75)$

$P_S(-) = \left(f, \frac{2-1}{4}\right) = (f, 0.25) = (0, 0.25)$

$$W_j = 1 - \frac{1}{n} \sum_{P \in \{+,-\}} \sum_{x \in P} \delta((P(j), x(j)))$$

| B |
|---|
| 0 |
| 0.5 |
| 1 |
| 0.25 |

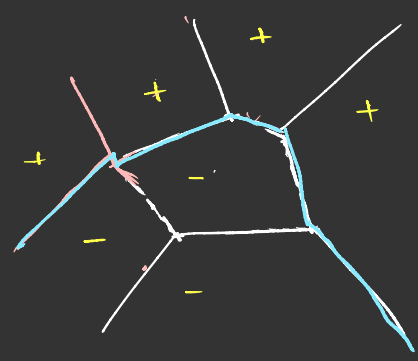
min=1
max=5
2
4
2

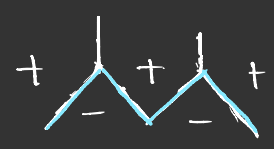
$W_A = 1 - \frac{1}{4}(0+1+0+0) = 0.75$

$W_B = 1 - \frac{1}{4}(0.5+0+0.5+0)$

$= 1 - \frac{1}{4} = \frac{3}{4} = 0.75$

## Voronoi maps





a) The number of attributes ↑.