DECISION TREE MODEL

Bùi Tiến Lên

2023

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

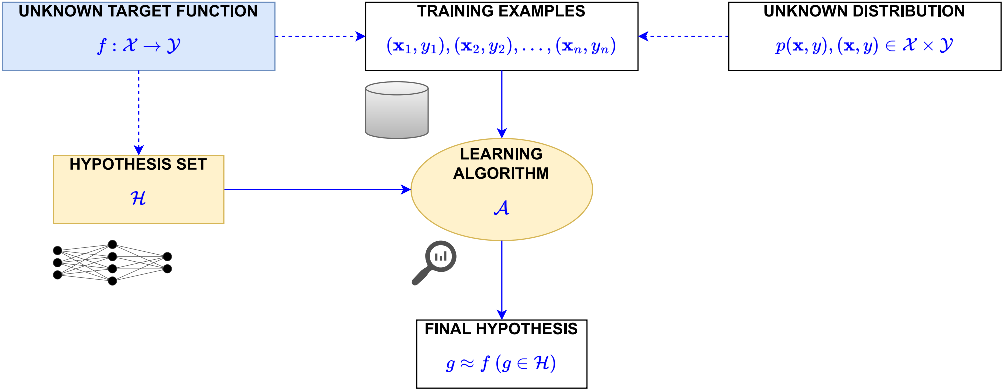
Contents

Notation

symbol	meaning
$a, b, c, N \ldots$	scalar number
$\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y} \ldots$	column vector
$\boldsymbol{X}, \boldsymbol{Y} \ldots$	matrix
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{N}$	set of natural numbers
$\mathbb{R}^D$	set of vectors
$\mathcal{X}, \mathcal{Y}, \ldots$	set
$\mathcal{A}$	algorithm

operator	meaning
$\boldsymbol{w}^\intercal$	transpose
$\boldsymbol{X}\boldsymbol{Y}$	matrix multiplication
$\boldsymbol{X}^{-1}$	inverse

Learning diagram

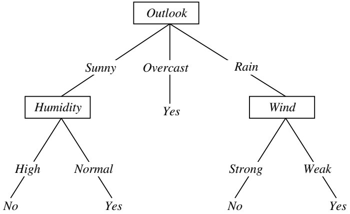


Decision Tree Representation

Decision tree representation

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No $\ominus$
D2	Sunny	Hot	High	Strong	No $\ominus$
D3	Overcast	Hot	High	Weak	Yes $\oplus$
D4	Rain	Mild	High	Weak	Yes $\oplus$
D5	Rain	Cool	Normal	Weak	Yes $\oplus$
...	...	...	...	...	...

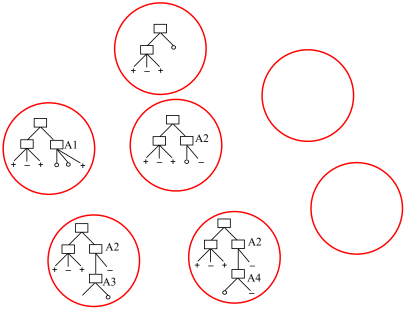


When to Consider Decision Trees

- Classification problems
- Instances describable by attribute–value pairs
- Attributes are discrete valued
- Target function is discrete valued

Problem Statement

- Hypothesis set $\mathcal{H}$ (**finite set**, there are 2^{2^n} trees for n binary attributes and binary target)
  - With 6 binary attributes, there are 18,446,744,073,709,551,616 trees



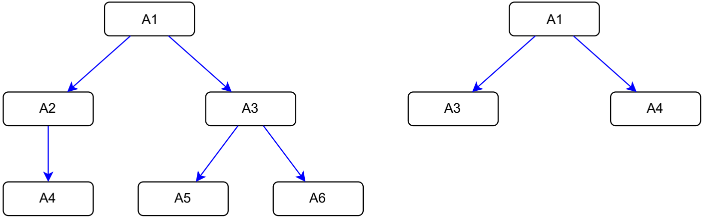
- **Task T**: to predict y from $\boldsymbol{x}$ by outputting $\hat{y} = h_T(\mathbf{x}) = T(\boldsymbol{x})$
- **Performance measure P**: classification error

Learning Algorithm

- Entropy
- Gini
- Misclassification

Which tree is best?

- Which tree would be chosen? if both trees are fitted to $\mathcal{D} = \{(\boldsymbol{x}_1, y_1)...(\boldsymbol{x}_N, y_N)\}$



Occam's Razor

Principle of Occam's Razor

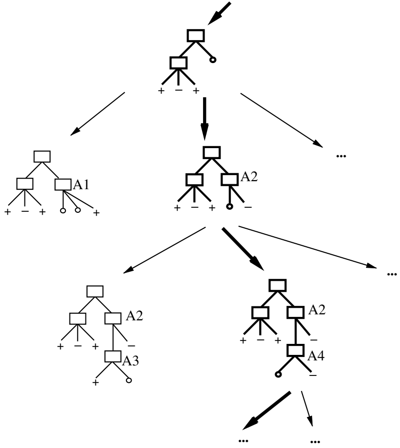
The **simplest** model that fits the data is also the most plausible (prefer the shortest hypothesis that fits the data)

- **Inductive Bias**: Preference for short trees, and for those with high *information gain* attributes near the root

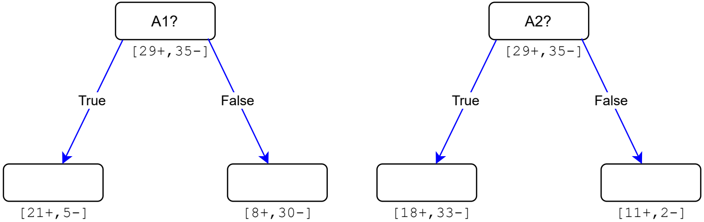
Top-Down Algorithm

function DECISION-TREE-LEARNING(*examples*, *attributes*)
  if all *examples* have *the same classification* **then** return *the classification*
  else if *attributes* is $\emptyset$ **then** return PLURALITY-VALUE(*examples*)
  else
    $A \leftarrow$ the "best" decision attribute for next *node*
    Assign A as decision attribute for *node*
    For each value of A, create new descendant of *node*
    Sort training examples to child nodes and **repeat** these steps

Top-Down Algorithm (cont.)



Which attribute is best?



Big Picture

Information Gain

- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S

Concept 1

- **Entropy** measures the impurity of S

$$Entropy(S) = -\left(p_\oplus \log_2 p_\oplus + p_\ominus \log_2 p_\ominus\right) \tag{1}$$

Information Gain (cont.)

- S is a set of items with C classes, and let $\boldsymbol{p} = \{p_i\}_{i=1}^{C}$ be the fraction of items labeled with class i in the set.

Concept 2

- **Entropy** measures the impurity of S

$$Entropy(S) = -\sum_{i=1}^{C} p_i \log_2 p_i \tag{2}$$

Information Gain (cont.)

Concept 3

- **Average entropy** on attribute A

$$AE(S, A) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{3}$$

- **Information gain** is expected reduction in entropy on A

$$Gain(S, A) = Entropy(S) - AE(S, A) \tag{4}$$

- The best attribute is an attribute that has the highest **information gain**

Gini index

Concept 4

- **Gini** impurity for a set of items S with C classes, and let $\boldsymbol{p} = \{p_i\}_{i=1}^{C}$ be the fraction of items labeled with class i in the set.

$$GiniImp(S) = 1 - \sum_{i=1}^{C} p_i^2 \tag{5}$$

- **Gini index** on attribute A

$$GiniIndex(S, A) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} GiniImp(S_v) \tag{6}$$

Misclassification index

Concept 5

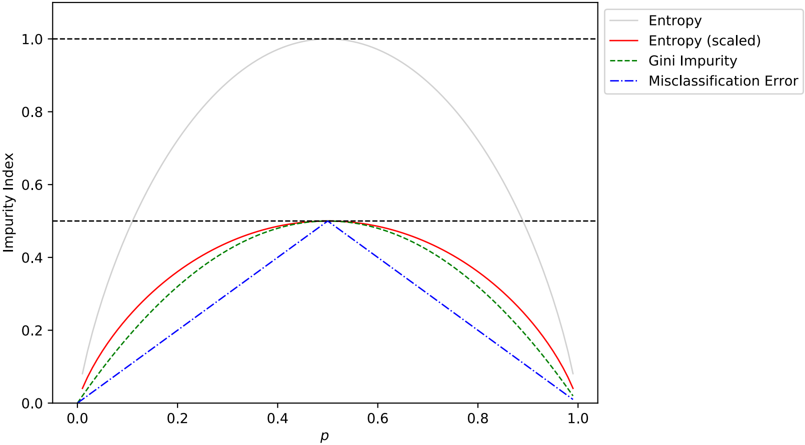
- **Misclassification impurity index** for a set of items S with C classes, and let $\boldsymbol{p} = \{p_i\}_{i=1}^{C}$ be the fraction of items labeled with class i in the set.

$$MisImp(S) = 1 - \max \{p_i\}_{i=1}^{C} \tag{7}$$

- **Misclassification index** on attribute A

$$MisIndex(S, A) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} MisImp(S_v) \tag{8}$$

Entropy, Gini and Misclassification



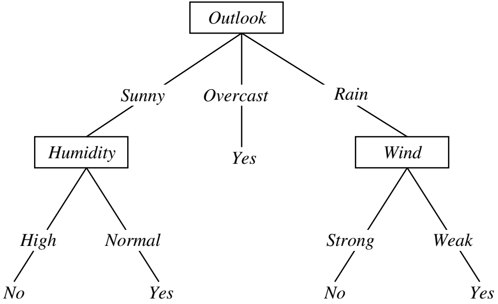
Example 1

- Find decision tree T given the following training data

$\mathcal{D} =$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example 1 - Finding Decision Tree and Converting to Rules



IF	$(Outlook = Sunny) \wedge (Humidity = High)$	**THEN**	$PlayTennis = No$
ELIF	$(Outlook = Sunny) \wedge (Humidity = Normal)$	**THEN**	$PlayTennis = Yes$
ELIF	$Outlook = Overcast$	**THEN**	$PlayTennis = Yes$
ELIF	$(Outlook = Rain) \wedge (Wind = Strong)$	**THEN**	$PlayTennis = No$
ELIF	$(Outlook = Rain) \wedge (Wind = Weak)$	**THEN**	$PlayTennis = Yes$
ELIF		**THEN**	**failure**

Evaluating Association Rules

Concept 6

An **association rule** is an implication of the form $X \rightarrow Y$ or **IF** X **THEN** Y

- Support of the association rule

$$\text{support}(X, Y) = P(X, Y) = \frac{\#\text{count}(X, Y)}{\text{total samples}} \tag{9}$$

- Confidence of the association rule

$$\text{confidence}(X \rightarrow Y) = P(Y \mid X) = \frac{\#\text{count}(X, Y)}{\#\text{count}(X)} \tag{10}$$

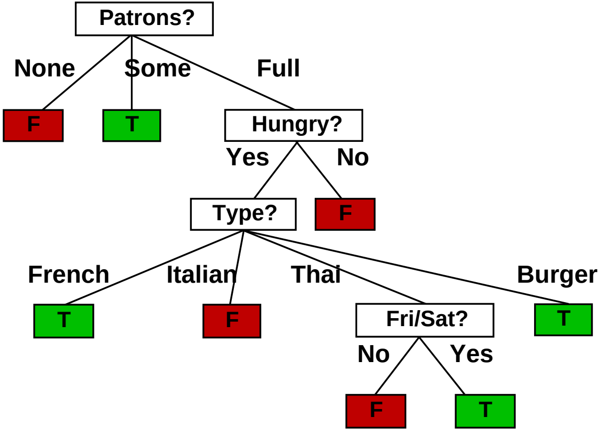
Example 2

- Find decision tree T given the following training data

$\mathcal{D} =$

#	Input attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Will Wait
1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	T
2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	F
3	No	Yes	No	No	Some	$	No	No	Burger	0–10	T
4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10–30	T
5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	F
6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	T
7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	F
8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	T
9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	F
10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	F
11	No	No	No	No	None	$	No	No	Thai	0–10	F
12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	T

Example 2 - Finding Decision Tree



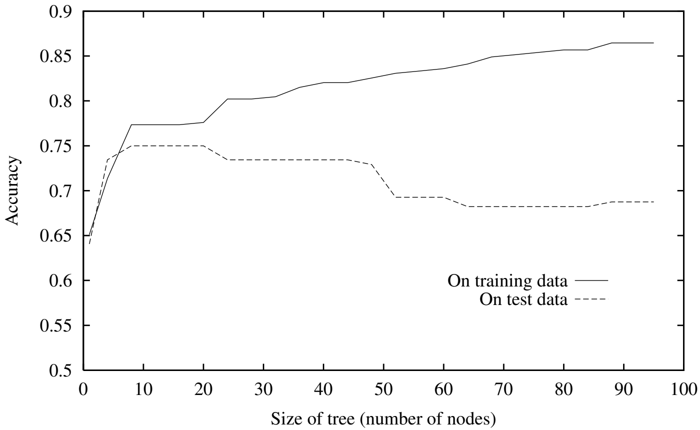
Word Example

1. Find decision tree T given the following training datasets
2. Find all **stumps** (decision tree with one node)

#	Vị	Màu	Vỏ	Độc tính
1	Ngọt	Đỏ	Nhẵn	Không
2	Cay	Đỏ	Nhẵn	Có
3	Chua	Vàng	Có gai	Không
4	Cay	Vàng	Có gai	Có
5	Ngọt	Tím	Có gai	Không
6	Chua	Vàng	Nhẵn	Không
7	Ngọt	Tím	Nhẵn	Không
8	Cay	Tím	Có gai	Có
9	Cay	Tím	Có gai	Không
10	Cay	Tím	Có gai	Có
11	Cay	Vàng	Có gai	Có

Generalization And Overfitting

Overfitting in Decision Tree Learning



Avoiding Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune

How to select “best” tree:

- Measure performance over training data
- Measure performance over separate validation data set
- Minimize

$$error(tree) + \lambda size(tree)$$

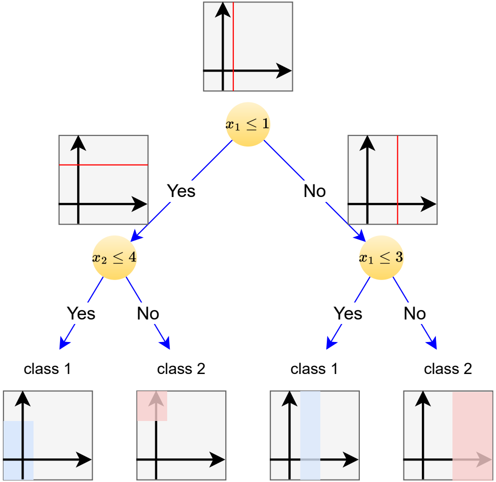
Continuous Valued Attributes

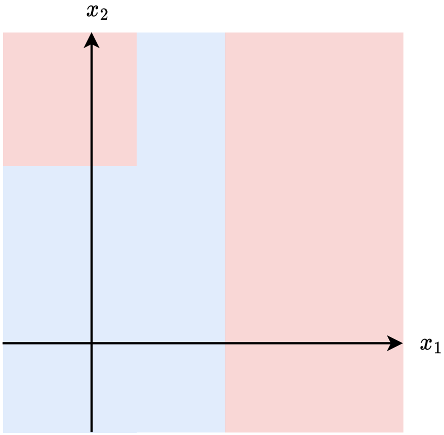
Continuous Valued Attributes

Create a discrete attribute for continuous variable

- Binary node
  $\mathit{Temperature} > 36$ or $\mathit{Temperature} \leq 36$
- General node
  $\mathit{Temperature} \in \{(-\infty, 0], (0, 10], (10, 20], (20, \infty)\}$

Decision Tree with Continuous Valued Attributes





Extended Top-Down Algorithm

procedure $\textsc{GenerateTree}(\mathcal{D})$
    if $\textsc{Entropy}(\mathcal{D}) < \epsilon$
        Create leaf labelled by majority class in $\mathcal{D}$
        return
    $i \leftarrow \textsc{SplitAttribute}(\mathcal{D})$
    for each branch of X_i
        Find $\mathcal{D}_i$ falling in branch
        $\textsc{GenerateTree}(\mathcal{D}_i)$

Extended Top-Down Algorithm (cont.)

function $\textsc{SplitAttribute}(\mathcal{D})$
  $entropy_{min} \leftarrow \infty$
  for all attributes X_i where $i = 1, ..., d$
    if X_i is discrete with n values
      Split $\mathcal{D}$ into $\mathcal{D}_1, ..., \mathcal{D}_n$ by X_i
      $e \leftarrow \textsc{AverageEntropy}(\mathcal{D}_1, ..., \mathcal{D}_n)$
      if $e < entropy_{min}$: $entropy_{min} \leftarrow e$, $i_{min} \leftarrow i$
    if X_i is numeric
      for all possible splits
        Split $\mathcal{D}$ into $\mathcal{D}_1, \mathcal{D}_2$ on X_i
        $e \leftarrow \textsc{AverageEntropy}(\mathcal{D}_1, \mathcal{D}_2)$
        if $e < entropy_{min}$: $entropy_{min} \leftarrow e$, $i_{min} \leftarrow i$
  return i_{min}

Example 3

- Find decision tree T given the following training data

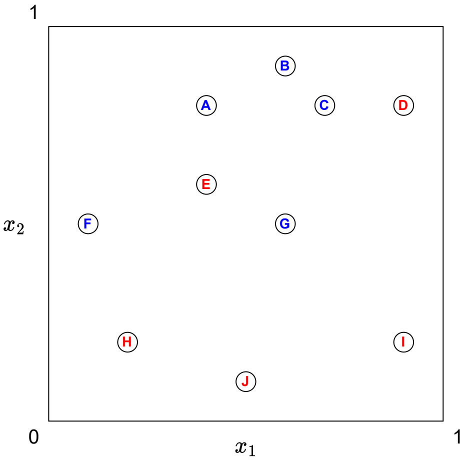
$\mathcal{D} =$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	37	High	Weak	No
D2	Sunny	37	High	Strong	No
D3	Overcast	38	High	Weak	Yes
D4	Rain	28	High	Weak	Yes
D5	Rain	20	Normal	Weak	Yes
D6	Rain	18	Normal	Strong	No
D7	Overcast	19	Normal	Strong	Yes
D8	Sunny	27	High	Weak	No
D9	Sunny	21	Normal	Weak	Yes
D10	Rain	26	Normal	Weak	Yes
D11	Sunny	26	Normal	Strong	Yes
D12	Overcast	27	High	Strong	Yes
D13	Overcast	36	Normal	Weak	Yes
D14	Rain	28	High	Strong	No

Example 4

- Find decision tree T given the following training data

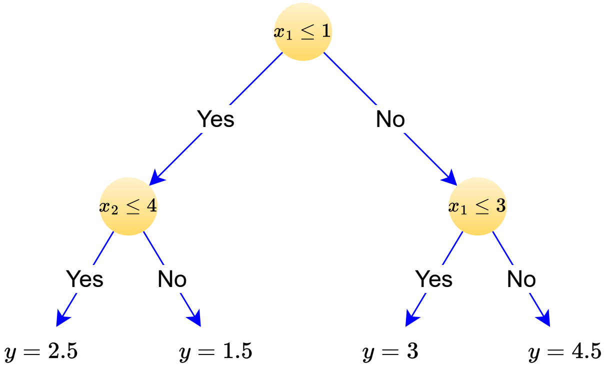
#	x_1	x_2	label
A	0.4	0.8	1
B	0.6	0.9	1
C	0.7	0.8	1
D	0.9	0.8	2
E	0.4	0.6	2
F	0.1	0.5	1
G	0.6	0.5	1
H	0.2	0.2	2
I	0.9	0.2	2
J	0.5	0.1	2



Regression Trees

Regression Trees

- A **regression tree** is constructed in almost the same manner as a classification tree, except that the impurity measure that is appropriate for classification is replaced by a measure appropriate for regression.



Loss Function

	Classification	Regression
Dataset	$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$	
Target	y_i categorical value	y_i real value
Loss function	Entropy	Error $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ $e = \frac{1}{n}\sum(y_i - \overline{y})^2$ $cv = \frac{\sqrt{e}}{\overline{y}}$

Top-Down Algorithm

procedure $\textsc{GenerateTree}(\mathcal{D})$
    if $\textsc{Error}(\mathcal{D}) < \epsilon$
        Create leaf valued by $\overline{y}$
        return
    $i \leftarrow \textsc{SplitAttribute}(\mathcal{D})$
    for each branch of X_i
        Find $\mathcal{D}_i$ falling in branch
        $\textsc{GenerateTree}(\mathcal{D}_i)$

Top-Down Algorithm (cont.)

function $\textsc{SplitAttribute}(\mathcal{D})$
  $e_{min} \leftarrow \infty$
  for all attributes X_i where $i = 1, ..., d$
    if X_i is discrete with n values
      Split $\mathcal{D}$ into $\mathcal{D}_1, ..., \mathcal{D}_n$ by X_i
      $e \leftarrow \textsc{AverageError}(\mathcal{D}_1, ..., \mathcal{D}_n)$
      if $e < e_{min}$: $e_{min} \leftarrow e$, $i_{min} \leftarrow i$
    if X_i is numeric
      for all possible splits
        Split $\mathcal{D}$ into $\mathcal{D}_1, \mathcal{D}_2$ on X_i
        $e \leftarrow \textsc{AverageError}(\mathcal{D}_1, \mathcal{D}_2)$
        if $e < e_{min}$: $e_{min} \leftarrow e$, $i_{min} \leftarrow i$
  return i_{min}

Example 5

- Find regression tree T given the following training data

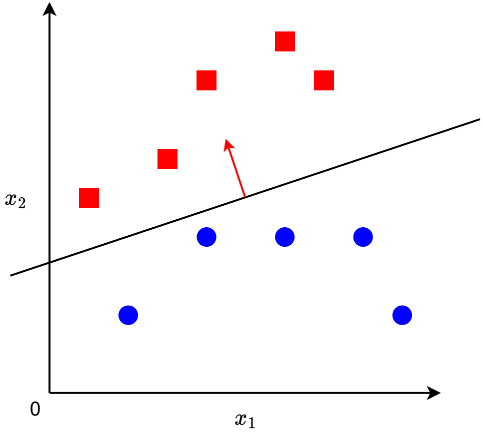
$\mathcal{D} =$

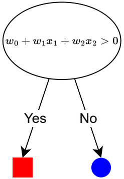
Day	Outlook	Temperature	Humidity	Wind	Play time (m)
D1	Rainy	Hot	High	Weak	26
D2	Rainy	Hot	High	Strong	30
D3	Overcast	Hot	High	Weak	46
D4	Sunny	Mild	High	Weak	46
D5	Sunny	Cool	Normal	Weak	62
D6	Sunny	Cool	Normal	Strong	23
D7	Overcast	Cool	Normal	Strong	43
D8	Rainy	Mild	High	Weak	36
D9	Rainy	Cool	Normal	Weak	38
D10	Sunny	Mild	Normal	Weak	46
D11	Rainy	Mild	Normal	Strong	48
D12	Overcast	Mild	High	Strong	62
D13	Overcast	Hot	Normal	Weak	44
D14	Sunny	Mild	High	Strong	30

Multivariate Trees

Multivariate Trees

- In the case of a univariate tree, only **one** input dimension **is used** at a split.
- In a multivariate tree, at a decision node, **all** input dimensions **can be used** and thus it is more general.



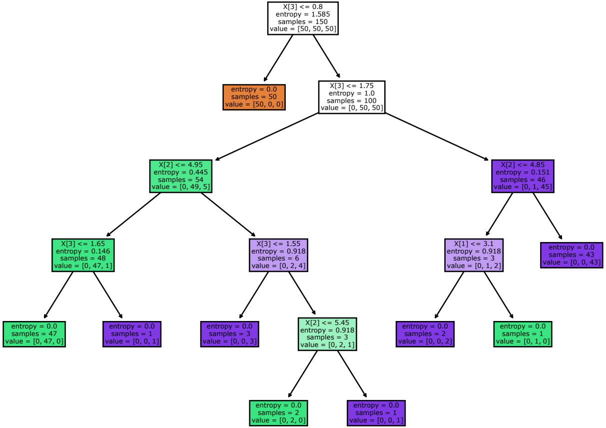


Programming Examples

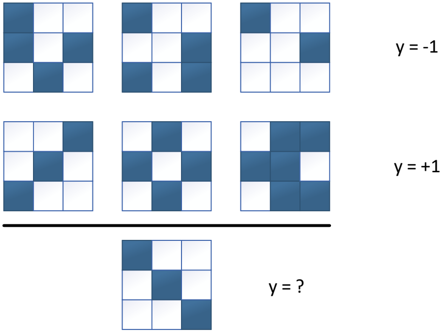
```python
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree

iris = load_iris()
clf = DecisionTreeClassifier(criterion="entropy")
clf.fit(iris.data, iris.target)
plot_tree(clf, filled=True)
plt.show()
```

Programming Examples (cont.)



A Learning Puzzle Revisited



References

Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep learning.
MIT press.

Lê, B. and Tô, V. (2014).
Cở sở trí tuệ nhân tạo.
Nhà xuất bản Khoa học và Kỹ thuật.

Russell, S. and Norvig, P. (2021).
Artificial intelligence: a modern approach.
Pearson Education Limited.