Decision Trees

Adopted from Machine Learning, Mitchell

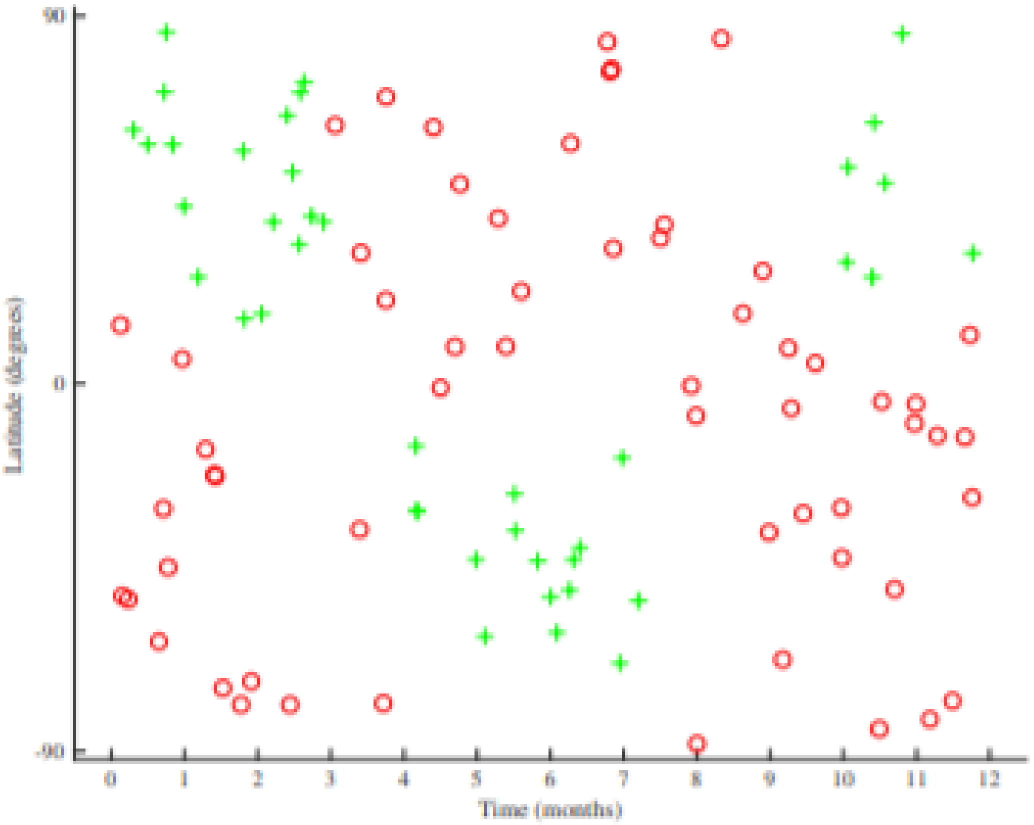
Decision Trees

- It approximates discrete-valued target function.

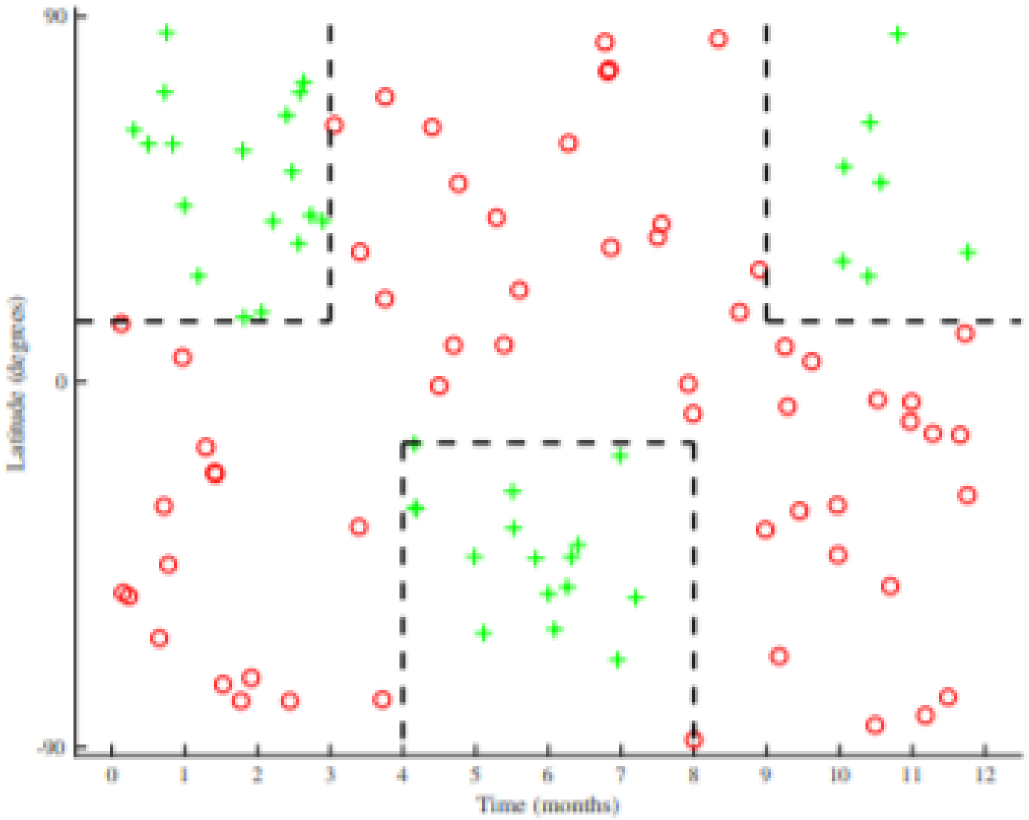
- Learning function is a set of *if-then* rules to improve human readability.

- Highly interpretable

I want to ski

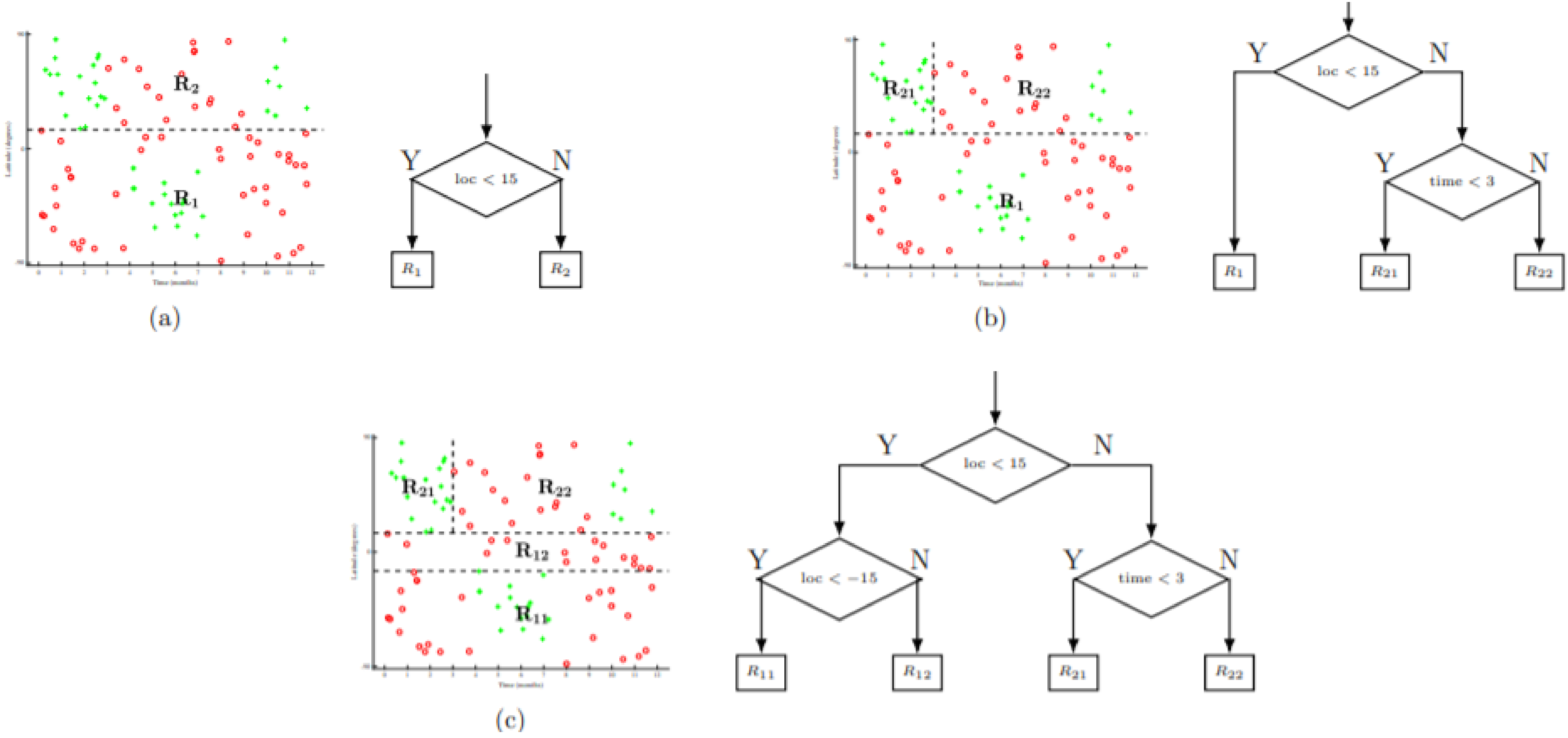


Latitude (degrees)
Time (months)
90
0
-90
0
1
2
3
4
5
6
7
8
9
10
11
12



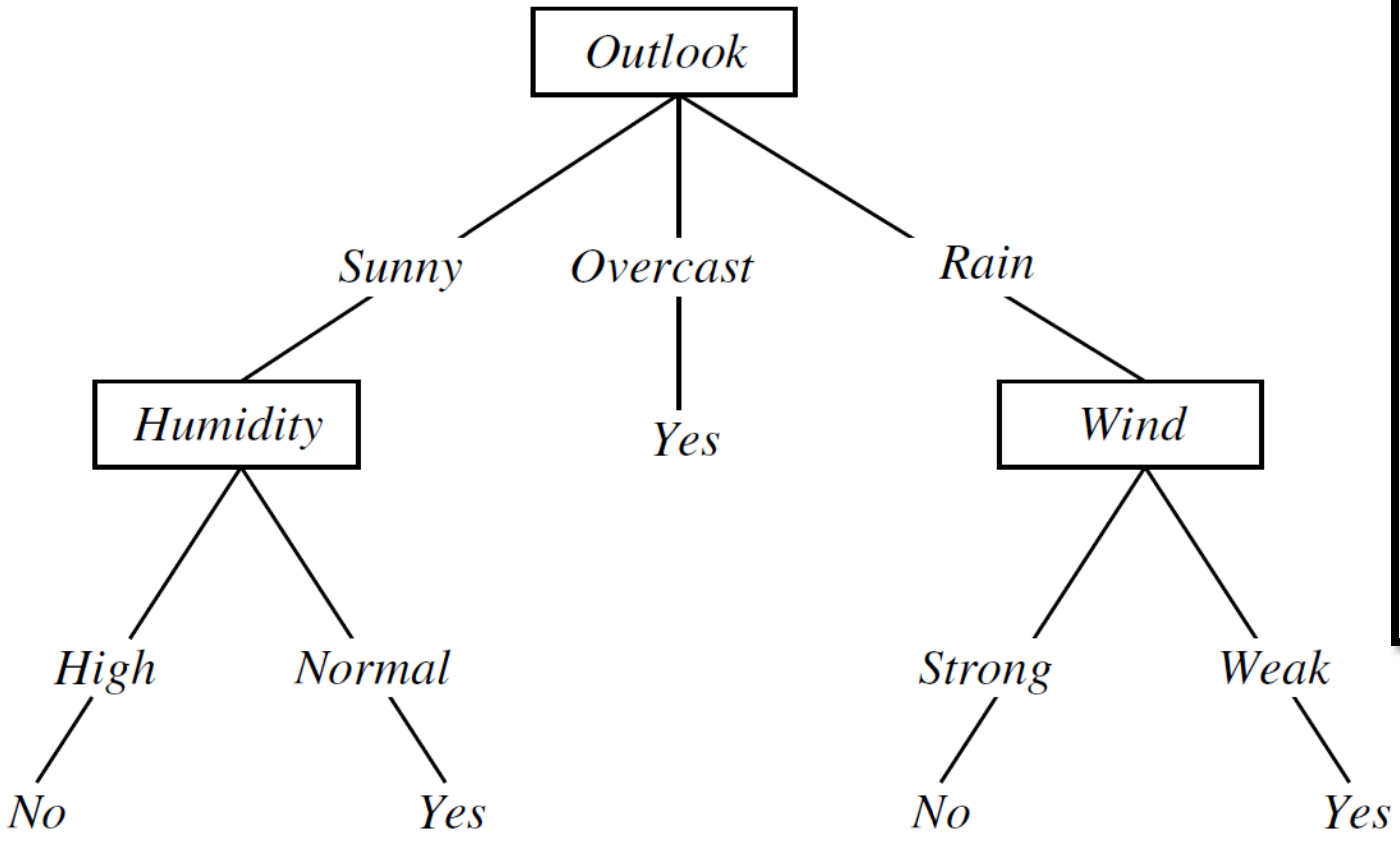
Latitude (degrees)
Time (months)
90
0
-90
0
1
2
3
4
5
6
7
8
9
10
11
12

I want to ski



R_2
R_1
loc < 15
Y
N
R_1
R_2
(a)
R_{21}
R_{22}
R_1
loc < 15
time < 3
R_1
R_{21}
R_{22}
(b)
R_{21}
R_{22}
R_{12}
R_{11}
loc < 15
loc < −15
time < 3
R_{11}
R_{12}
R_{21}
R_{22}
(c)

Decision tree for *PlayTennis*



Decision tree representation:

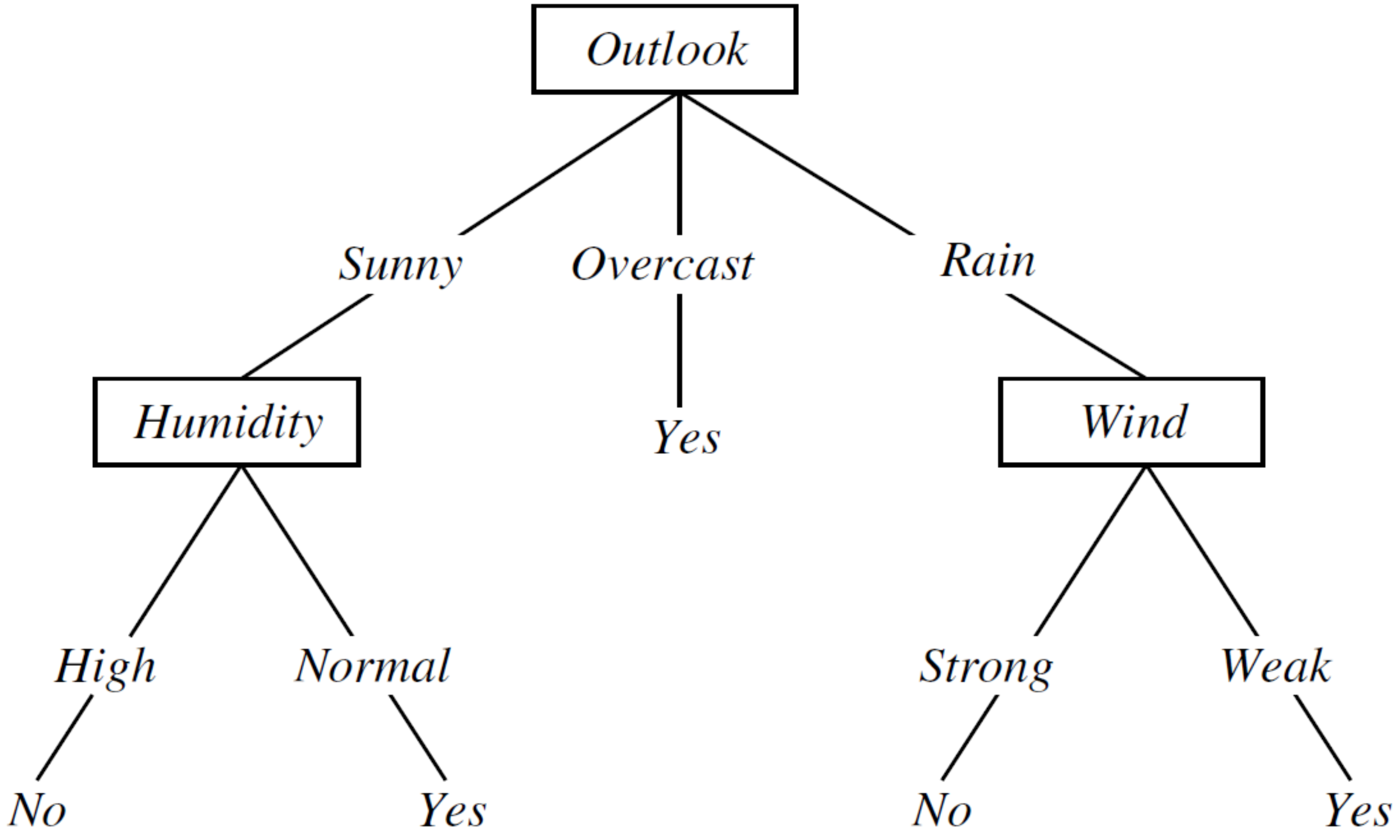
- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

How would we represent:

- $\wedge, \vee$, XOR
- $(A \wedge B) \vee (C \wedge \neg D \wedge E)$
- M of N

$\langle Outlook = Sunny,\ Temperature = Hot,\ Humidity = High,\ Wind = Strong \rangle$

Decision Tree



A disjunction of conjunctions of constraints on attribute values of instances.

$$
\begin{aligned}
&(Outlook = Sunny \wedge Humidity = Normal)\\
\vee\ &(Outlook = Overcast)\\
\vee\ &(Outlook = Rain \wedge Wind = Weak)
\end{aligned}
$$

When to use Decision Tress

1. Instances are represented by attribute-value pairs.
2. The target function has discrete output values.
3. Disjunctive descriptions may be required.
4. The training data may contain error.
5. The training data may contain missing attribute values.

Examples:

- Equipment or medical diagnosis
- Credit risk analysis
- Modeling calendar scheduling preferences

Types of Decision Trees

- **ID3:** Categorical feature that will yield the **largest information gain** for categorical targets

- **C4.5:** Successor to ID3 and removes the restriction that features must be categorical by dynamically defining a discrete attribute (based on numerical variables) that partitions the continuous attribute value into a discrete set of intervals.

- **CART (Classification and Regression Trees):** Similar to C4.5, but it differs in that it supports numerical target variables (regression) and does not compute rule sets.

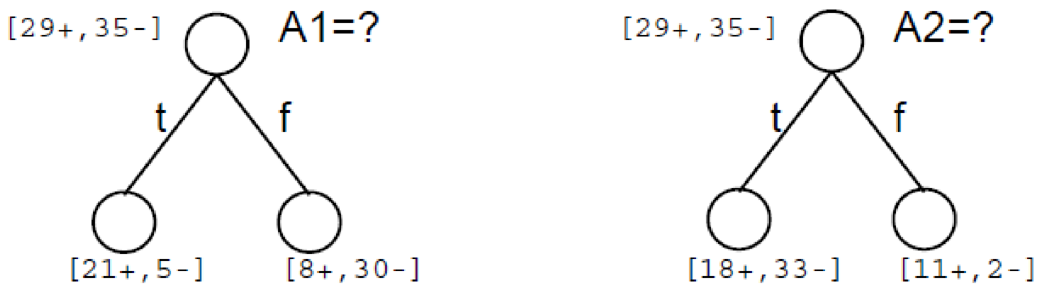
  - https://www.quora.com/What-are-the-differences-between-ID3-C4-5-and-CART (ID: Iterative Dichotomiser)
  - https://medium.com/datadriveninvestor/tree-algorithms-id3-c4-5-c5-0-and-cart-413387342164

Inductive Learning of Decision Tree

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?



- Greedy
- Top-down
- Recursive partitioning

Loss Function

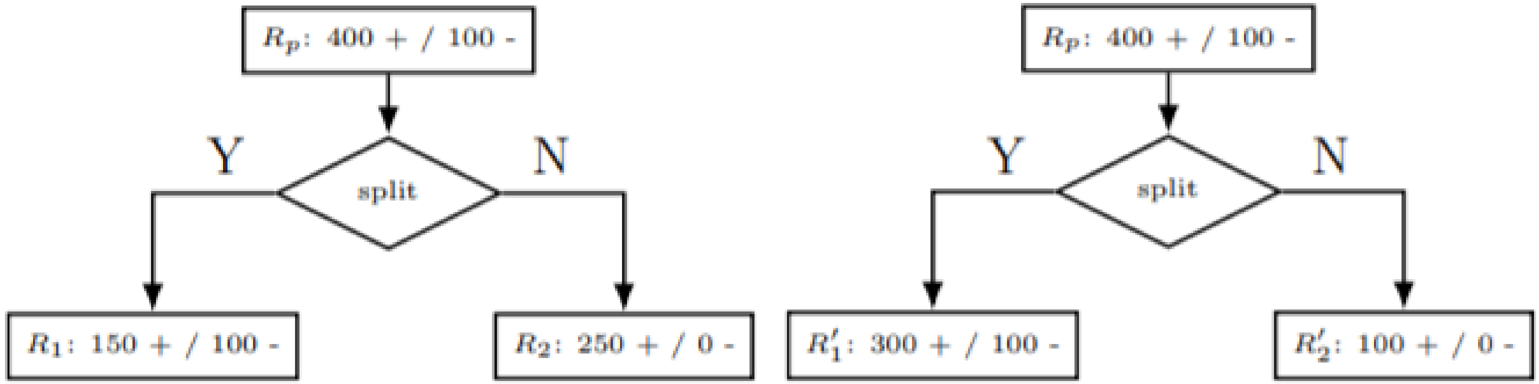
- We calculate the loss of the parent $L(R_p)$ as well as the cardinality-weighted loss of the children

- Select an attribute greedily that maximizes the decrease in loss

$$L(R_p) - \frac{|R_1|L(R_1) + |R_2|L(R_2)}{|R_1| + |R_2|}$$

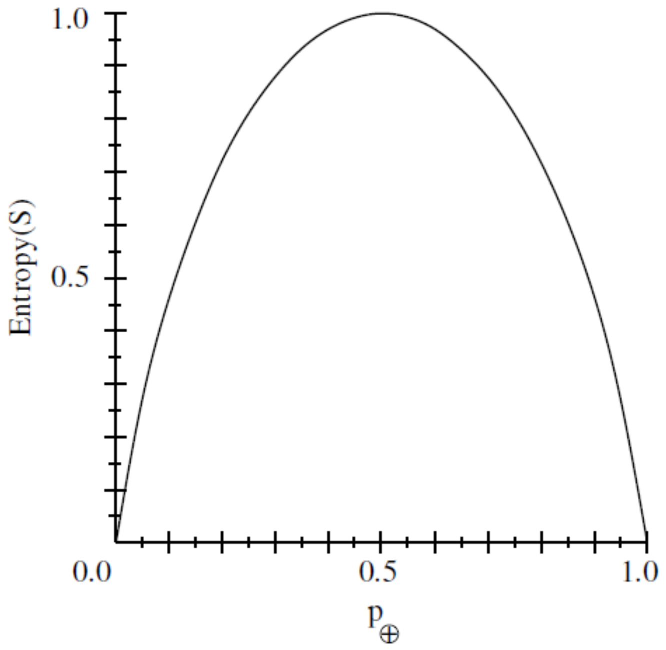
- Misclassification loss: $L_{misclass}(R) = 1 - \max_c(\hat{p}_c)$

Loss Function: Misclassification loss



$$L(R_p) = \frac{|R_1|L(R_1) + |R_2|L(R_2)}{|R_1| + |R_2|} = \frac{|R_1'|L(R_1') + |R_2'|L(R_2')}{|R_1' + |R_2'|} = 100$$

(Shannon's) Entropy



Entropy measures homogeneity of examples.

- 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
- Entropy measures the impurity of S

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

(Shannon's) Entropy

$Entropy(S)$ = expected number of bits needed to encode class ($\oplus$ or $\ominus$) of randomly drawn member of S (under the optimal, shortest-length code)

Why?

Information theory: optimal length code assigns $-\log_2 p$ bits to message having probability p.

So, expected number of bits to encode $\oplus$ or $\ominus$ of random member of S:

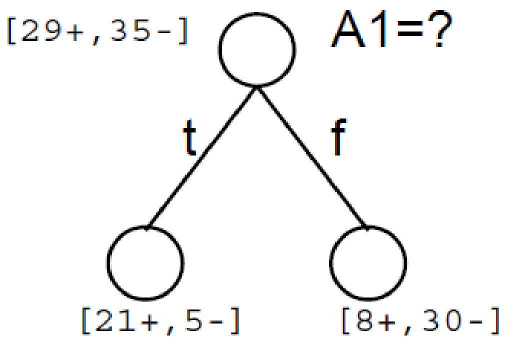
$$p_{\oplus}(-\log_2 p_{\oplus}) + p_{\ominus}(-\log_2 p_{\ominus})$$

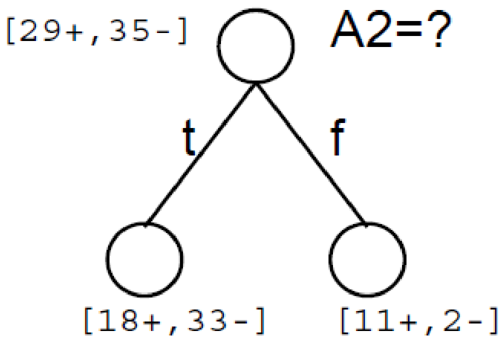
$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Information Gain

$Gain(S, A)$ = expected reduction in entropy due to sorting on A

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



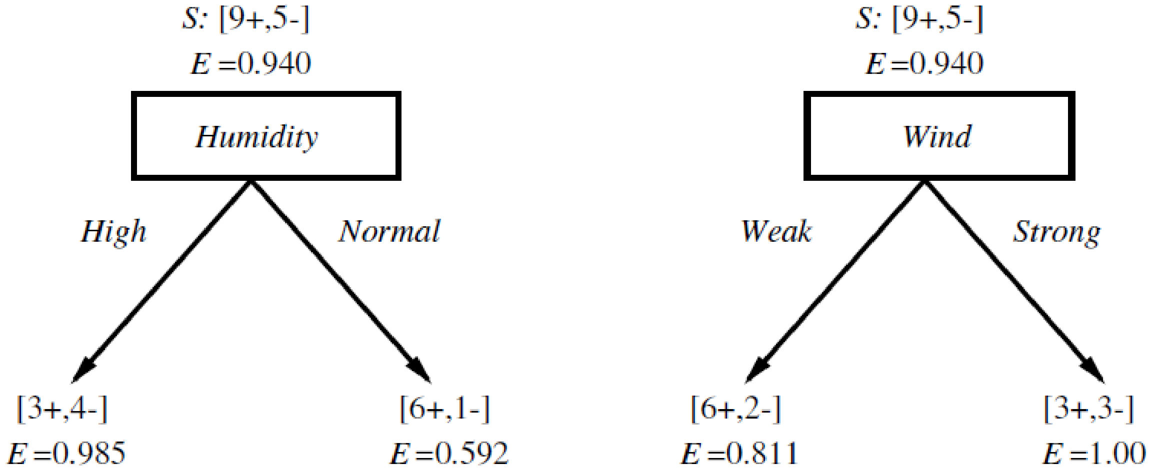


Gain(S,A): Number of bits saved when encoding the target value of an arbitrary member of S, by knowing the value of attribute A.

Training examples

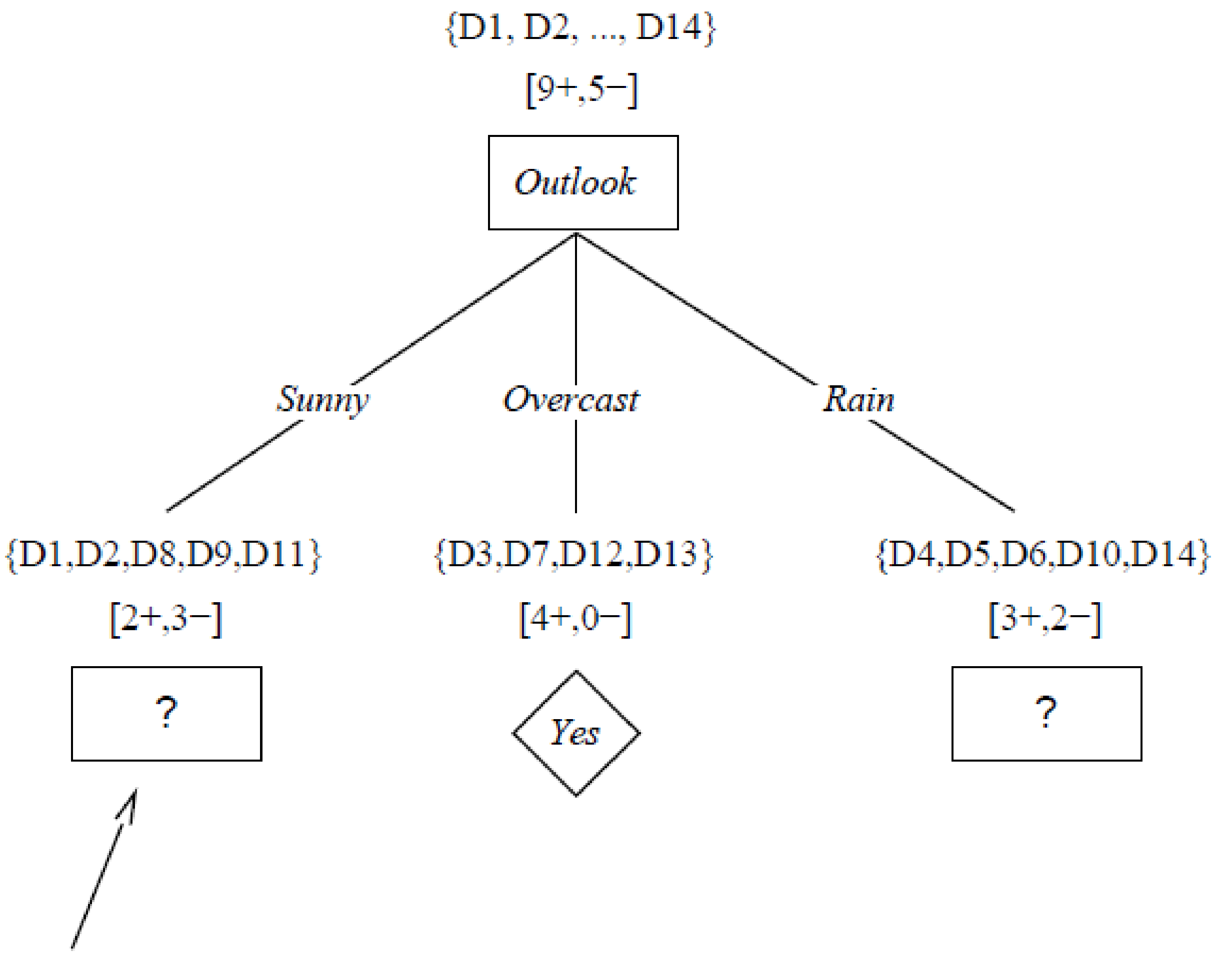
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal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

Which attribute is the best classifier?



Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048



Which attribute should be tested here?

Day	Outlook	Temperature	Humidity	Wind	PlayTenni
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

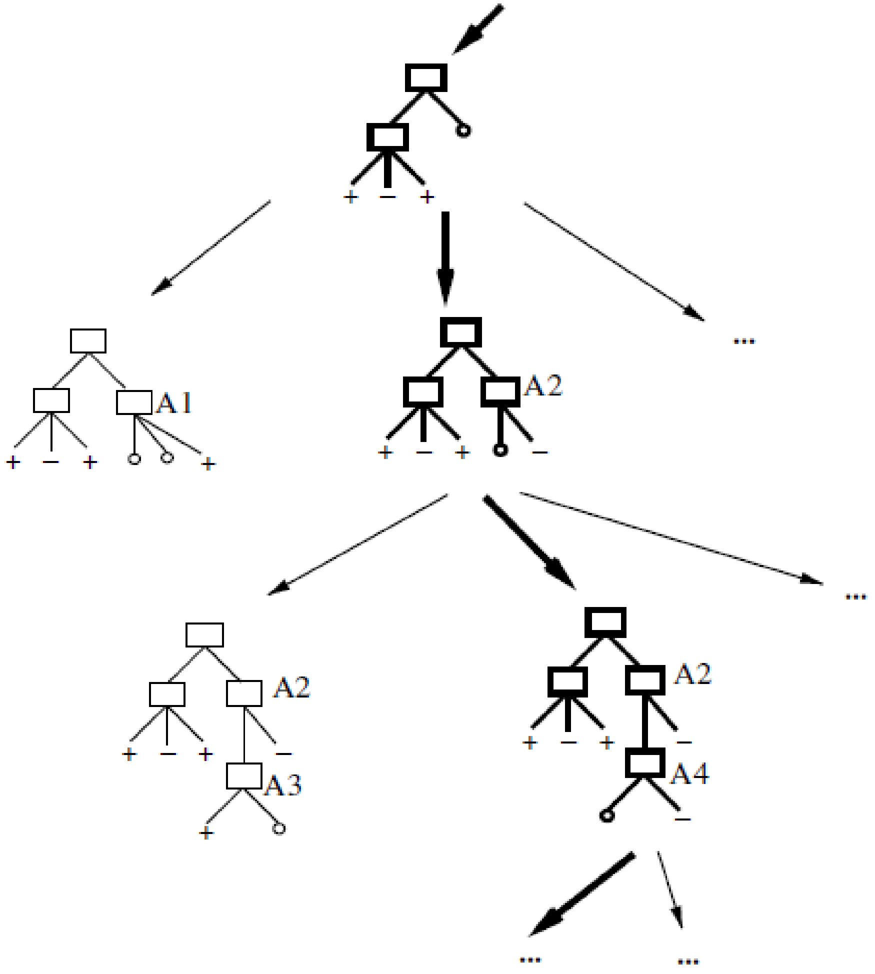
S_{sunny} = {D1,D2,D8,D9,D11}

$Gain(S_{sunny}, Humidity) = .970 - (3/5)\ 0.0 - (2/5)\ 0.0 = .970$

$Gain(S_{sunny}, Temperature) = .970 - (2/5)\ 0.0 - (2/5)\ 1.0 - (1/5)\ 0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\ 1.0 - (3/5)\ .918 = .019$

Hypothesis space search in ID3



- ID3: Iterative Dichotomiser 3 !!

Inductive Bias

Set of assumptions that, together with the training data, justify the classifications assigned by the learner to future instances.

Inductive Bias in Decision Trees

- Selects in favor of short trees over longer ones
- Select trees that place the attributes with highest information gain closest to the root

Approximate inductive bias of ID3: Shorter trees are preferred over longer trees

- Think about a BFS-ID3 algorithm
- ID3 is just a greedy version of BFS-ID3

Shorter trees are preferred over longer trees. that place the attributes with highest information gain closest to the root are preferred over those that do not.