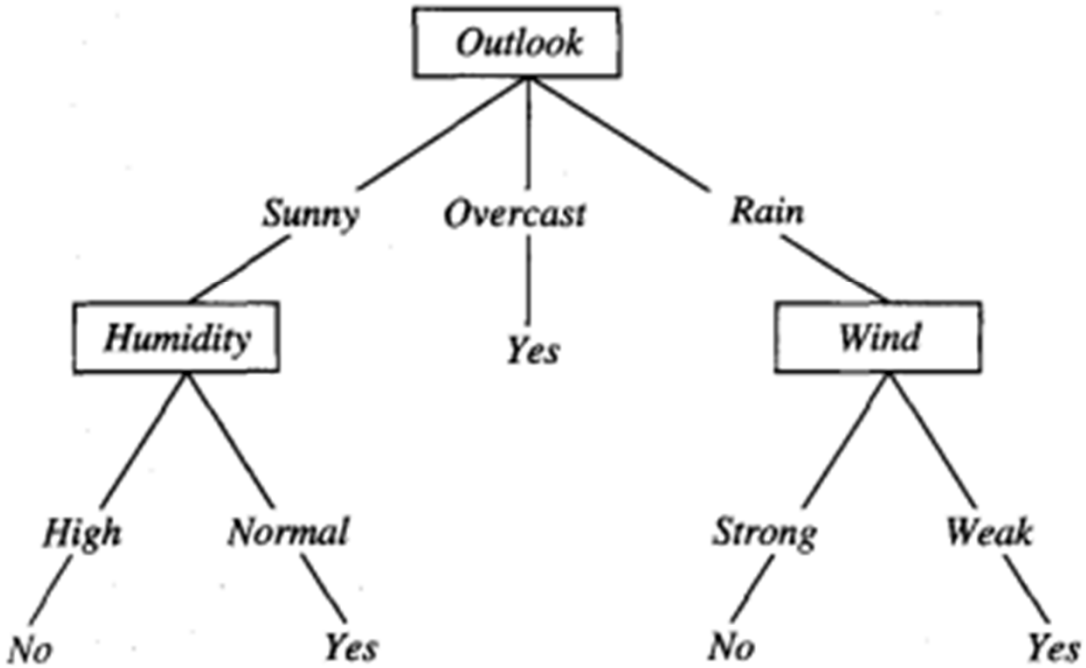
Decision Tree

Decision tree learning

- It is a method that uses inductive inference to approximate a target function, which will produce discrete values.It is widely used, robust to noisy data, and considered a practical method for learning disjunctive expressions.


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

Approriate problems

- *Instances are represented by attribute-value pairs.* Instances are described by a fixed set of attributes (e.g., *Temperature*) and their values (e.g., *Hot*). The easiest situation for decision tree learning is when each attribute takes on a small number of disjoint possible values (e.g., *Hot*, *Mild*, *Cold*). However, extensions to the basic algorithm (discussed in Section 3.7.2) allow handling real-valued attributes as well (e.g., representing *Temperature* numerically).
- *The target function has discrete output values.* The decision tree in Figure 3.1 assigns a boolean classification (e.g., *yes* or *no*) to each example. Decision tree methods easily extend to learning functions with more than two possible output values. A more substantial extension allows learning target functions with real-valued outputs, though the application of decision trees in this setting is less common.
- *Disjunctive descriptions may be required.* As noted above, decision trees naturally represent disjunctive expressions.
- *The training data may contain errors.* Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.
- *The training data may contain missing attribute values.* Decision tree methods can be used even when some training examples have unknown values (e.g., if the *Humidity* of the day is known for only some of the training examples). This issue is discussed in Section 3.7.4.

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

to a collection of examples S, is defined as

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

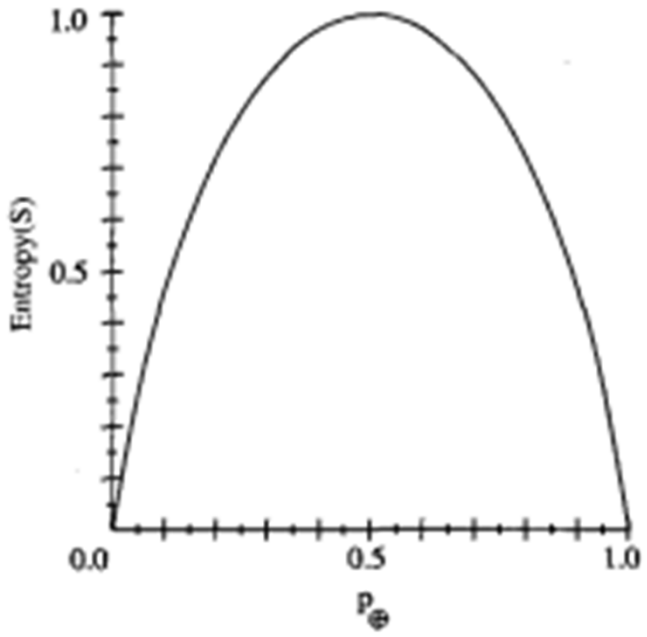


FIG
The
as
bet

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

TABLE 3.2
Training examples for the target concept *PlayTennis*

$$\begin{aligned} \textit{Entropy}([9+, 5-]) &= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) \\ &= 0.940 \end{aligned}$$

$$
\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) \\
&\quad - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}
$$

attributes are

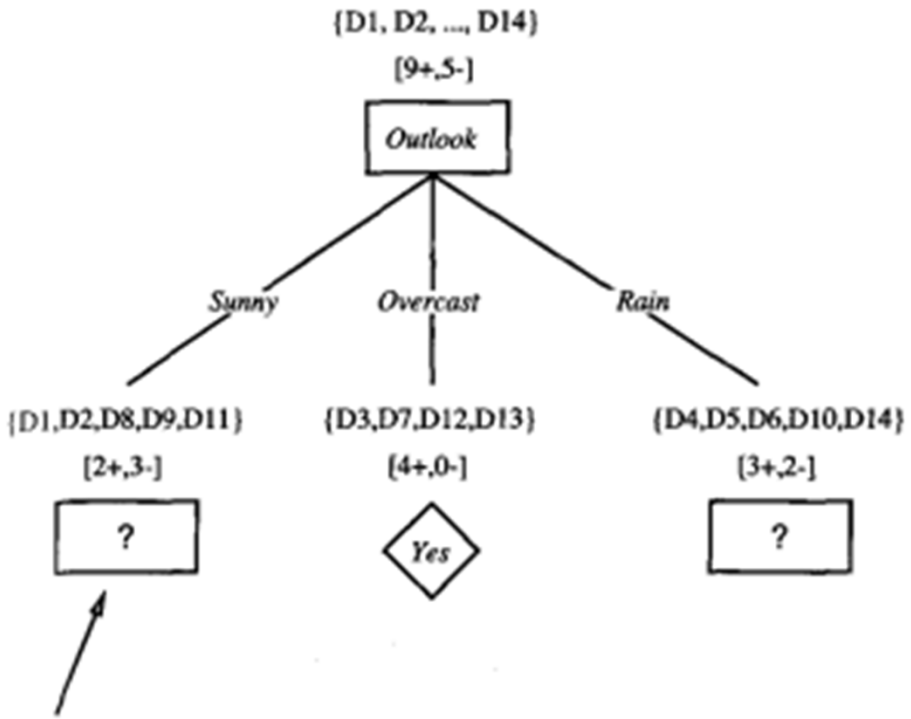
$$Gain(S, Outlook) = 0.246$$

$$Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048$$

$$Gain(S, Temperature) = 0.029$$

a collection of training examples from Table 2



Which attribute should be tested here?

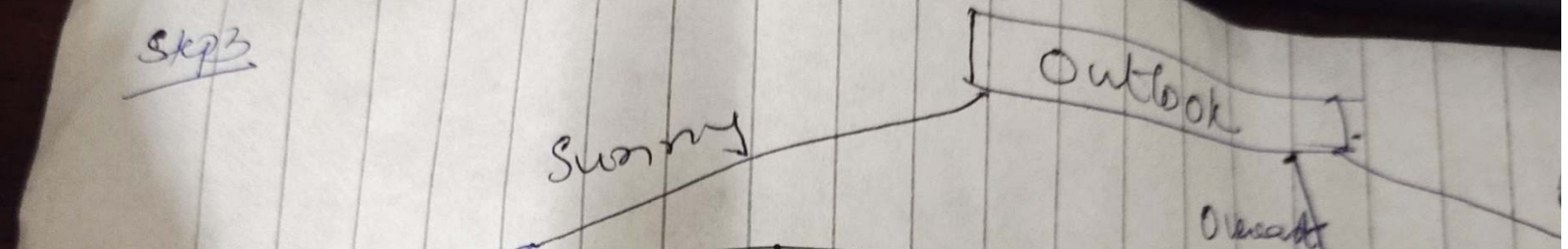
S_{sunny} = {D1,D2,D8,D9,D11}

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

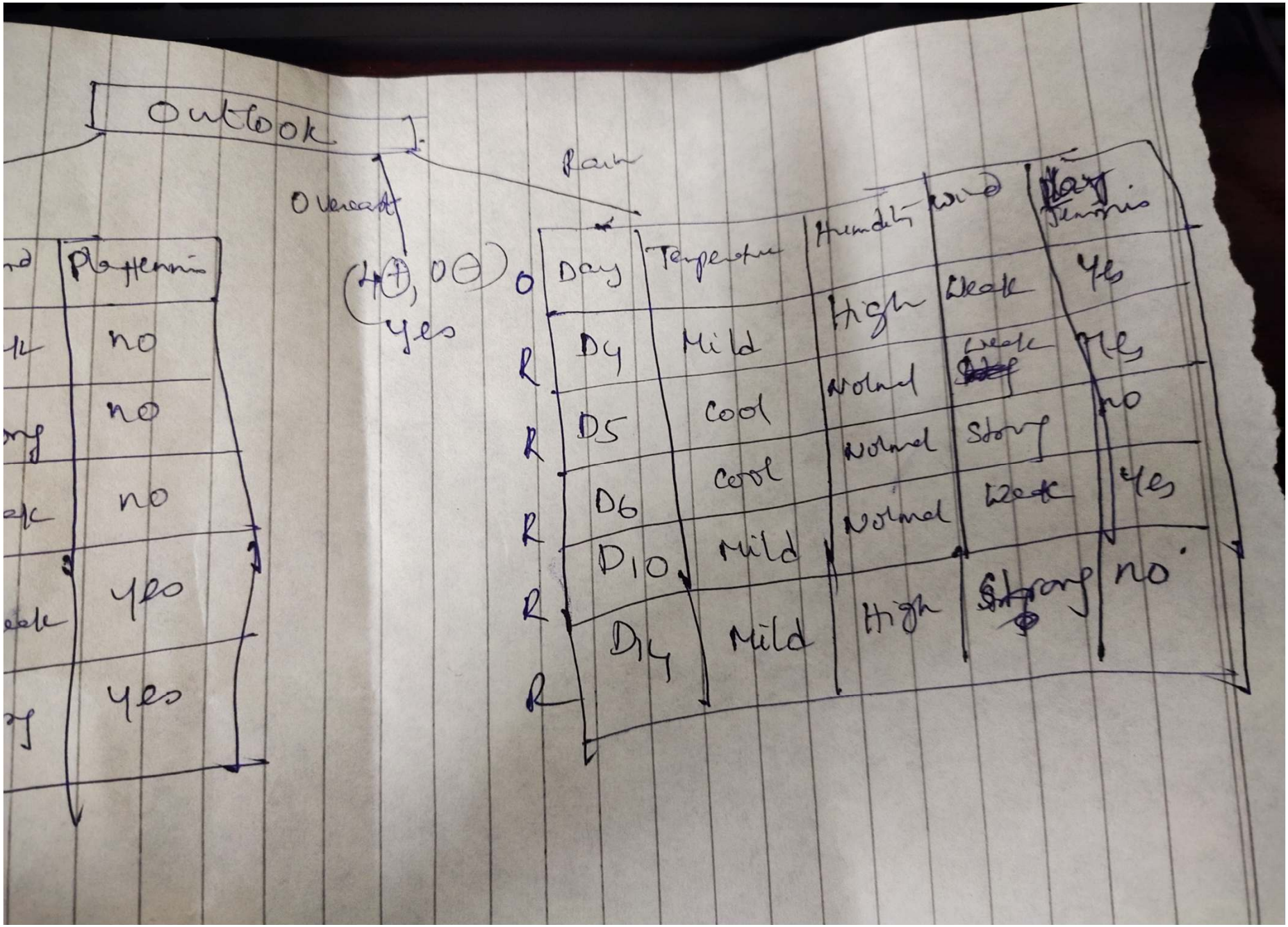
Gain (S_{sunny}, *Wind*) = .970 - (2/5) 1.0 - (3/5) .918 = .019

Step 3



	Day	Temperature	Humidity	Wind	Playtennis
S	D_1	Hot	High	Weak	no
S	D_2	Hot	High	Strong	no
S	D_8	Mild	High	Weak	no
S	D_9	Cool	Normal	Weak	yes
S	D_{11}	Mild	Normal	Strg	yes

O
R
R
R
R
R


Outlook
Overcast
Rain
(4⊕, 0⊖)
yes
Play Tennis
no
no
no
yes
yes
Day
Temperature
Humidity
Wind
Play Tennis
D4
Mild
High
Weak
Yes
D5
Cool
Normal
Strong
no
D6
Cool
Normal
Weak
Yes
D10
Mild
Normal
Weak
Yes
D14
Mild
High
Strong
no

Basic Decision tree Learning Algorithm

- ID3 –Constructs the trees in top down approach
- Starts with the question- Which attribute should be tested at the root of the tree?

```
ID3 (Examples, Target_Attribute, Attributes)
    Create a root node for the tree
    If all examples are positive, Return the single-node tree Root,
with label = +.
    If all examples are negative, Return the single-node tree Root,
with label = -.
    If Atributes list is empty, then Return the single node tree
Root,
    with label = most common value of the target attribute in the
examples.
    Otherwise Begin
        A ← The Attribute that best classifies examples.
        Decision Tree attribute for Root = A.
        For each possible value, vi, of A,
            Add a new tree branch below Root, corresponding to the
test A = vi.
            Let Examples(vi) be the subset of examples that have the
value vi for A
            If Examples(vi) is empty
                Then below this new branch add a leaf node with label
= most common target value in the examples
            Else below this new branch add the subtree ID3
(Examples(vi), Target_Attribute, Attributes - {A})
    End
    Return Root
```

Issues in Decision Tree Learning

Practical issues in learning decision trees

- Determining how deeply to grow the decision tree
- handling continuous attributes
- choosing an appropriate attribute selection measure,
- handling training data with missing attribute values,
- handling attributes with differing costs,
- improving computational efficiency.

Avoiding Overfitting the Data

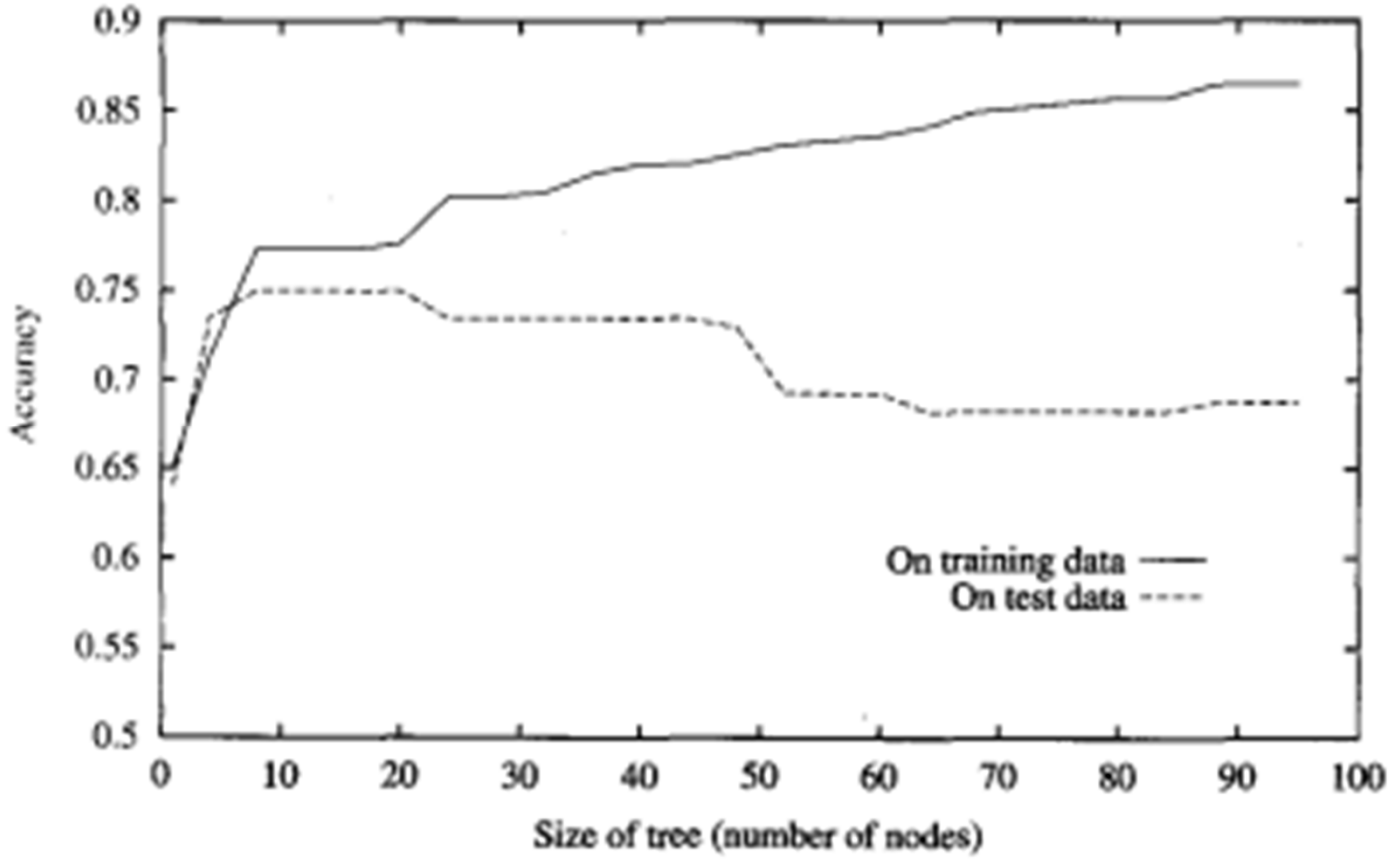
- In the example,tree is grown deeply enough to perfectly classify the training examples.
- it can lead to difficulties when there is noise in the data
- or when the number of training examples is too small to produce a representative sample of the true target function.
- In either of these cases, this simple algorithm can produce trees that overfit the training examples.

Overfitting

- A hypothesis overfits the training examples if some other hypothesis that fits the training examples less well actually performs better over the entire distribution of instances (i.e., including instances beyond the training set).
- Definition

  Given a hypothesis space H, a hypothesis h E H is said to overfit the training data if there exists some alternative hypothesis h' E H, such that h has smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instances.

Overfitting in decision tree learning. (ID3)



Approaches to avoiding overfitting in decision tree learning

- approaches that stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data
- approaches that allow the tree to overfit the data, and then post-prune the tree.

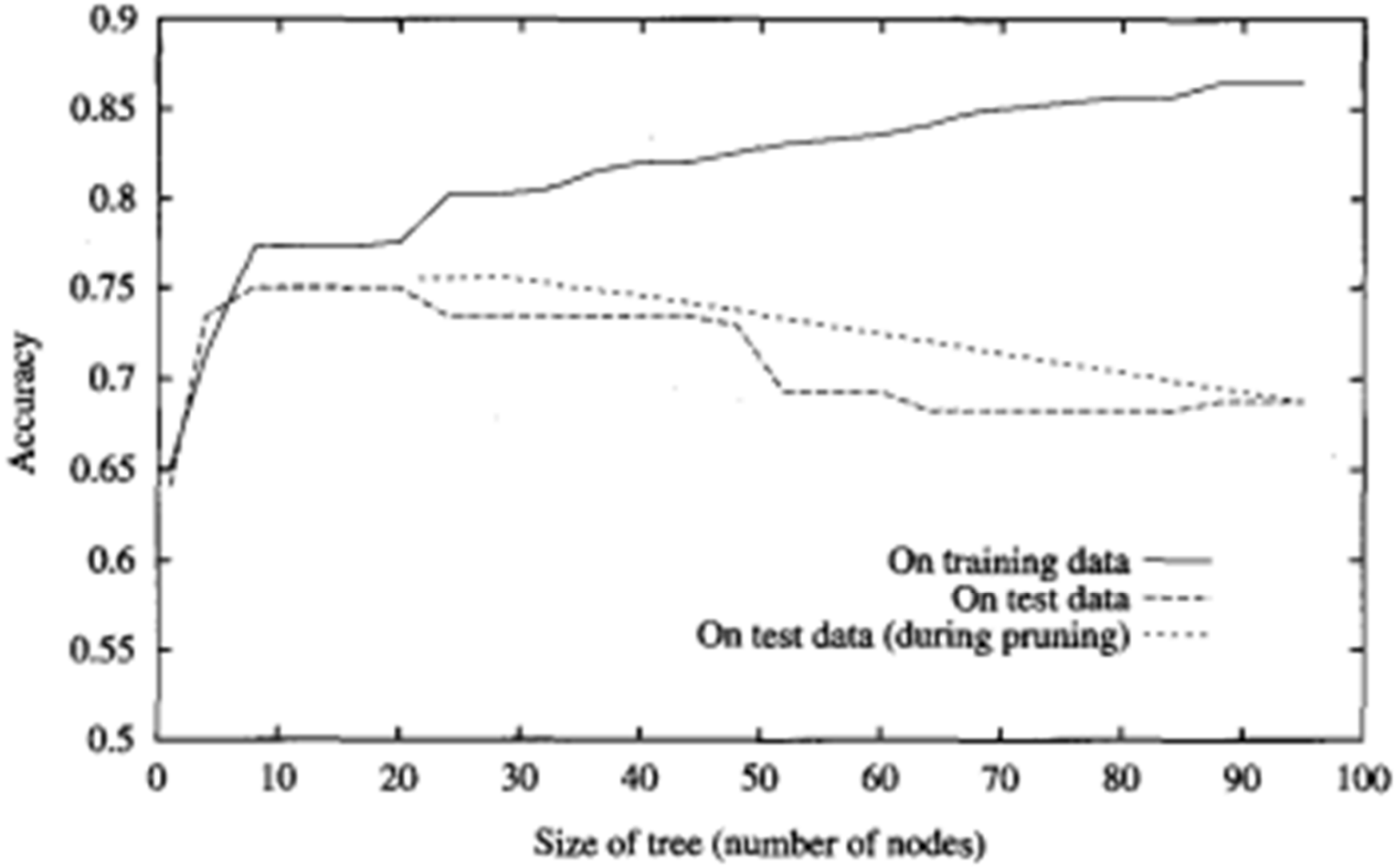
criterion is to be used to determine the correct final tree size

- Use a separate set of examples, distinct from the training examples, to evaluate the utility of post-pruning nodes from the tree.
- Use all the available data for training, but apply a statistical test to estimate whether expanding (or pruning) a particular node is likely to produce an improvement beyond the training set.
- For example, Quinlan (1986) uses a chi-square test to estimate whether further expanding a node is likely to improve performance over the entire instance distribution, or only on the current sample of training data.
- Use an explicit measure of the complexity for encoding the training examples and the decision tree, halting growth of the tree when this encoding size is minimized.
- This approach, based on a heuristic called the Minimum Description Length principle.

Validation Set

- The first of the above approaches is the most common and is often referred to as a training and validation set approach.
- In this approach, the available data are separated into two sets of examples: a training set, which is used to form the learned hypothesis, and a separate validation set, which is used to evaluate the accuracy of this hypothesis over subsequent data and,
- in particular, to evaluate the impact of pruning this hypothesis.
- The motivation is this: Even though the learner may be misled by random errors and coincidental regularities within the training set, the validation set is unlikely to exhibit the same random fluctuations.
- Therefore, the validation set can be expected to provide a safety check against overfitting the spurious characteristics of the training set.
- Of course, it is important that the validation set be large enough to itself provide a statistically significant sample of the instances.
- One common heuristic is to withhold one-third of the available examples for the validation set, using the other two-thirds for training.

REDUCED ERROR PRUNING



References

- https://www.youtube.com/watch?v=UzpwBb3qAbs
- https://heartbeat.comet.ml/introduction-to-decision-tree-learning-cd604f85e236 (python code)