Artificial Intelligence

LECTURE 3

Decision Trees

- Decision tree learning is an extraordinarily important algorithm for AI because it is very powerful, but also simple and efficient for extracting knowledge from data.
- A decision tree is cretaed by a process known as splitting on the value of attributes, i.e., testing the value of an attribute and then creating a branch for each of its possible values.
- In case of continuous attributes the test is normally whether the value is "less than or equal" or "greater than" a given value known the split value.

Decision Trees

- The splitting process continues until each branch can be labelled with just one decision class.
- The root node is a node corresponding to the original training set. All other nodes correspond to subsets of the training set.
- A decision node specifies some test to be carried out on a single attribute value, whith one branch and subtree for each possible outcome of the test.

Decision Trees

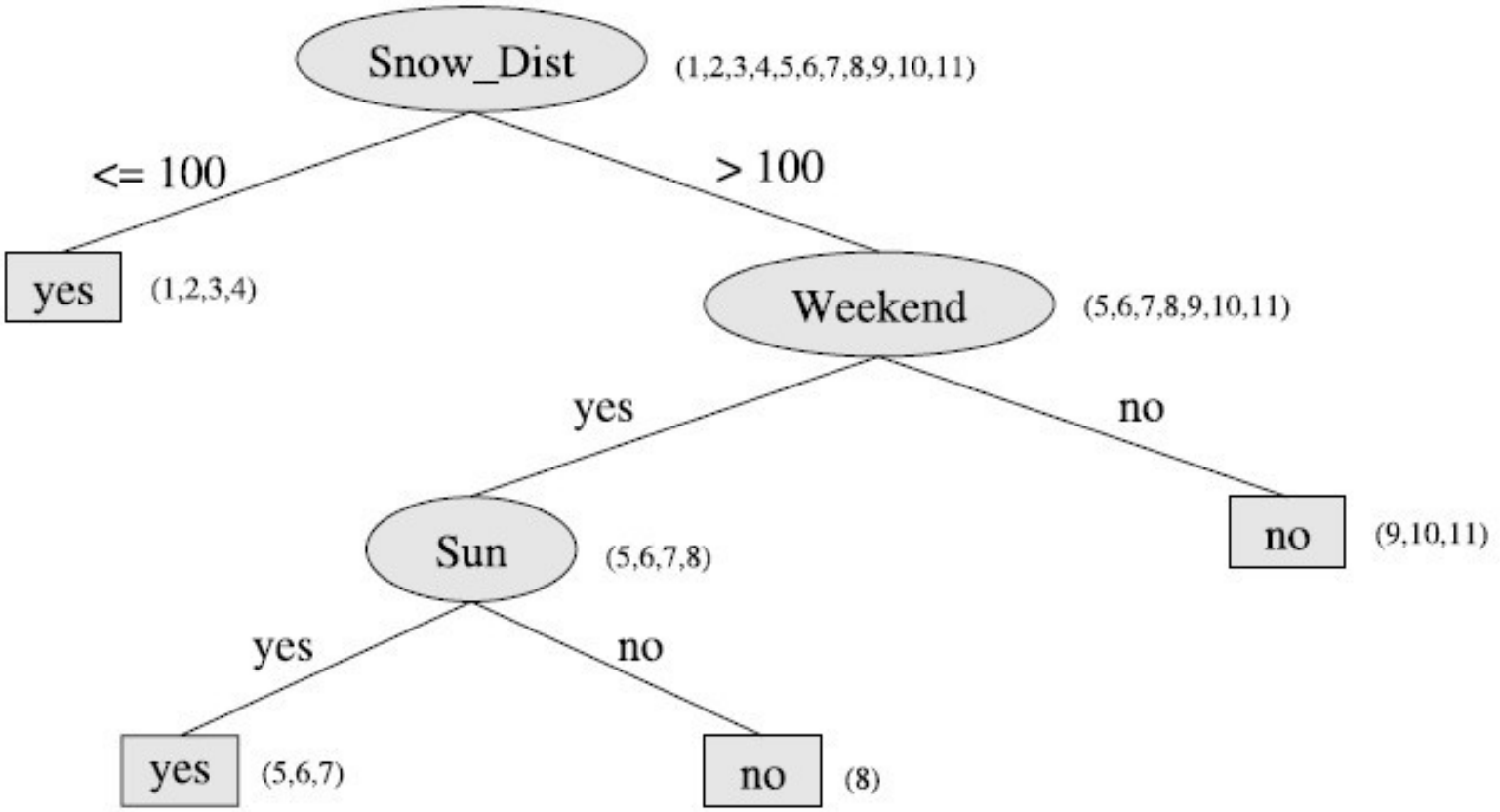
- A leaf indicates a decision class.
- Each branch from the root node to the leaf node corresponds to a classification rule.

Decision Table - Example

Day	*Snow_Dist*	*Weekend*	*Sun*	*Skiing*
1	≤100	Yes	Yes	Yes
2	≤100	Yes	Yes	Yes
3	≤100	Yes	No	Yes
4	≤100	No	Yes	Yes
5	>100	Yes	Yes	Yes
6	>100	Yes	Yes	Yes
7	>100	Yes	Yes	No
8	>100	Yes	No	No
9	>100	No	Yes	No
10	>100	No	Yes	No
11	>100	No	No	No

Ertel W.: Introduction to Artificial Intelligence. Springer-Verlag, London, 2011

Decision Tree - Example



Ertel W.: Introduction to Artificial Intelligence. Springer-Verlag, London, 2011

Entropy

$$E(S) = -\sum_{i=1}^{k} p_i \log p_i$$

k – the number of decision classes
n – the total number of instances
n_i – the number of instances with classification i (i.e., belonging to the i-th decision class)

$$p_i = \frac{n_i}{n}$$

$$i = 1, 2 \ldots, k$$

Information Gain

$$IG(S,a) = E(S) - \sum_{v \in V_a} \frac{n_v}{n} E(S_v)$$

a – attribute (feature)
V – the value set of a

n_V – the number of instances for which attribute a has value v

S_V – a subset of S for which attribute a has value v

Split Information

$$SI(S,a)=\sum_{v\in V_a}\frac{n_v}{n}\log_2\frac{n_v}{n}$$

a – attribute (feature)
V – the value set of a

n_V – the number of instances for which attribute a has value v

S_V – a subset of S for which attribute a has value v

Gain Ratio

$$GR(S,a)=\frac{IG(S,a)}{SI(S,a)}$$

Decision Tree Algorithms

- Decision trees are widely used as means of generating classification rules because of the existence a simple and powerful algorithm called (TDIDT – *Top Down Induction of Decision Trees*).
- TDIDT has formed the basis for many decision tree algorithms, e.g.:
  - CART
  - ID3
  - C4.5
- Decision tree algorithms use different attribute selection methods.

CART

- CART was proposed by Breiman et al. in 1984.
- A CART tree is a binary decision tree.

CART

Twoing criterion:

$$\Phi(S,t)=2\,P_L\,P_R\sum_{i=1}^{k}|P_L^i-P_R^i|$$

$$P_L=\frac{n_L}{n},\quad P_R=\frac{n_R}{n},\quad P_L^i=\frac{n_L^i}{n_t},\quad P_R^i=\frac{n_R^i}{n_t}$$

t - a node
k – the number of decision classes
n – the total number of instances

CART

n_L – the number of instances for the left branch

n_R – the number of instances for the right branch

n^i_L – the number of instances belonging to the *i*-th decision class for the left branch

n^i_R – the number of instances belonging to the *i*-th decision class for the right branch

n_t – the number of instances in a given node

ID3

- ID3 was invented by Quinlan.
- ID3 chooses an attribute for which the information gain is maximum.

C4.5

- C4.5 was proposed by Quinlan.
- C4.5 is an extension of Quinlan's earlier ID3 algorithm
- C4.5 builds decision trees from a set of training data in the same way as ID3, using the information gain.
- C4.5 made a number of improvements to ID3, e.g.:
  - handling both continuous and discrete attributes,
  - handling training data with missing attribute values.

C4.5

- Discrete attribute:
  - the standard test with one outcome and branch for each possible value.
- Continuous attribute:
  - the binary test with outcomes A<=V and A>V based on comparing the value of *A* against a threshold value *V*.

Overfitting

- A decision tree is said to overfit to the training data if it depends too much on irrelevant attributes of the training instances, with the results that it performs well on the training data but relatively poorly on unseen instances.
- To avoid the problem of overfitting decision trees are pruned down.

Pre-pruning

- Pre-pruning a decision tree involves using a termination condition to decide when it is desirable to terminate some of the branches prematurely as the tree is generated.