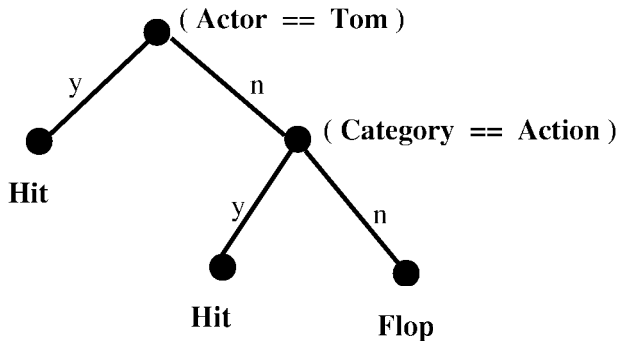
An Example Decision-Tree

Actor	Category	Class
Tom	Action	Hit
Arnold	Comedy	Flop
Tom	Tragedy	Hit
Bruce	Action	Hit
Bruce	Tragedy	Flop
Arnold	Action	Hit
Tom	Comedy	Hit

DECISION TREE



Decision-Tree Classification

Two phases:

1. Build an initial tree from the training data such that each leaf node is pure.
2. Prune this tree to increase its accuracy on test (unseen) data.

Tree Building

```
MakeTree(Training Data T)
{
        Partition(T);
}

Partition(Data S)
{
        if (all points in S are in the same class)) then
                return;
        for each attribute A do
                evaluate splits on attribute A;
        Use best split found to partition S into S_1 and S_2;
        Partition(S_1);
        Partition(S_2);
}
```

Splitting Index

A splitting index evaluates the "goodness" of the alternative splits for an attribute.

For a data set T containing examples from n classes, $gini(T)$ is defined as:

$$gini(T) = 1 - \sum p_j^2$$

where p_j is the relative frequency of class j in T.

If a split divides T into two subsets, T_1 and T_2, with n_1 and n_2 examples respectively, the new index of the divided data $gini_{split}(T)$ is given by

$$gini_{split}(T) = \frac{n_1}{n} gini(T_1) + \frac{n_2}{n} gini(T_2)$$

Split for Numeric Attributes

- Binary splits of the form $A \leq v$, where v is a real number.
- Sort the training examples in the partition based on the values of the attribute being considered for splitting.
- Let $v_1, v_2, \cdots, v_n$ be the sorted values of a numeric attribute A. Investigate the midpoint of each interval $v_i - v_{i+1}$ as a possible split point.

Split for Categorical Attributes

- If $S(A)$ is the set of possible values of a categorical attribute A, then the split test is of the form $A \in S^{'}$, where $S^{'} \subset S$.
- Use a greedy algorithm to avoid examining all possible subsets for an attribute with a large number of values.
- Start with an empty subset $S^{'}$ and add that one element of S to $S^{'}$ that gives the best split. Repeat the process until there is no improvement in the splits.

Tree Pruning

Choose the subtree with the least estimated error rate.

- First family (e.g. Cross-validation):
  - Take multiple pseudo-independent samples taken from the training data and grow a separate tree for each sample.
  - Use these multiple trees to estimate the error rates of the subtrees of the original tree.
- Second family:
  - Divide the training data into two parts.
  - Use one to build the tree and the other to prune.

Approach based on MDL (Minimum Description Length) principle