**Learning from decision trees**

Decision tree is a classifier in the form of a tree structure.

* Decision tree learning uses a decision tree as a predictive model which maps observation about an item to conclusions about the item’s target value.
* A decision tree can be used to visually and explicitly represent decisions & decision making.
* Each node is a leaf node or as a decision node.
* Leaf node- Indicates the values of a target attribute(class)
* Decision node- Specifies som test to be carried out usually on a single attribute value with one branch for each outcome of the test

Decision tree: a tree in which each non-leaf node has associated it with an attrivute (a feature of class)

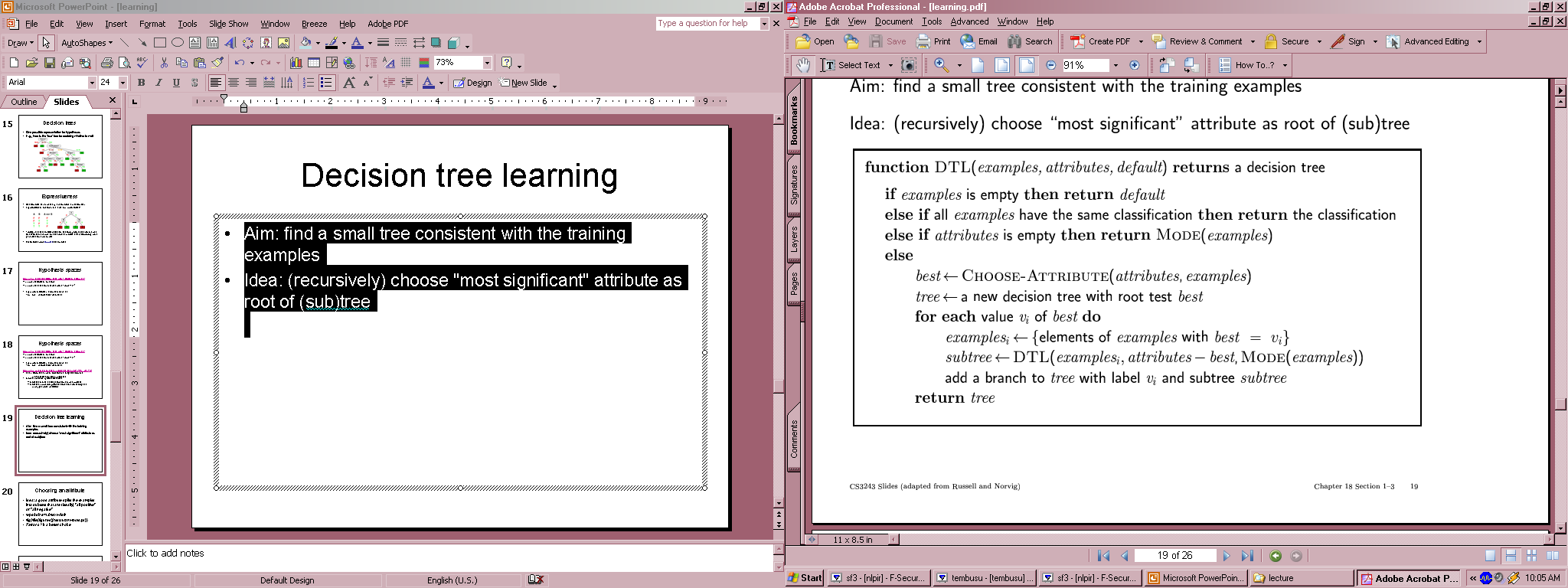
* Each leaf node has associated with it a classification .
* Each arc corresponds to one possible value of the attribute of its parent value.
* Decision trees are used for :

1. Classification
2. For classifying an example or set of examplesto do:
3. Start from root node
4. Follow appropriate decision branch
5. On reading a leaf node, the predicted class is obtained

* A decision tree can be represent – disjunction of conjunctions
* Learning problem for decision trees are also based on algorithm **ID3 Algorithm**

Decision tree learning

* Aim: find a small tree consistent with the training examples
* Idea: (recursively) choose "most significant" attribute as root of (sub)tree



Choosing Attribute:

Idea: A good attribute splits the examples into subsets that are “all positive” or “all negative”

* We need to pick attribute that goes as far as possible towards providing an exact classification of examples.
* We choose an attribute with highest entropy value as ROOT NODE.

**Entropy** is a measure of the uncertainty of a random variable; acquisition of information corresponds to a reduction in entropy.

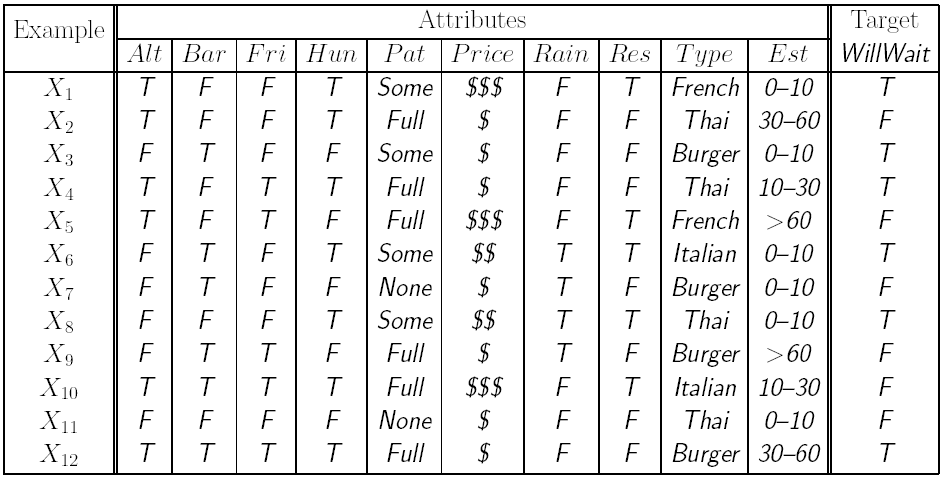
* A random variable with only one value—a coin that always comes up heads—has no uncertainty and thus its entropy is defined as zero; thus, we gain no information by observing its value.
* A flip of a fair coin is equally likely to come up heads or tails, 0 or 1, and we will soon show that this counts as "I bit" of entropy. The roll of a fair *four-sided* die has 2 bits of entropy, because it takes two bits to describe one of four equally probable choices.



Decision tree learning example

10 attributes:

1. **Alternate:** Is there a suitable alternative restaurant nearby? {yes,no}
2. **Bar:** Is there a bar to wait in? {yes,no}
3. **Fri/Sat:** Is it Friday or Saturday? {yes,no}
4. **Hungry:** Are you hungry? {yes,no}
5. **Patrons:** How many are seated in the restaurant? {none, some, full}
6. **Price:** Price level {$,$$,$$$}
7. **Raining:** Is it raining? {yes,no}
8. **Reservation:** Did you make a reservation? {yes,no}
9. **Type:** Type of food {French,Italian,Thai,Burger}
10. **Wait:** {0-10 min, 10-30 min, 30-60 min, >60 min}

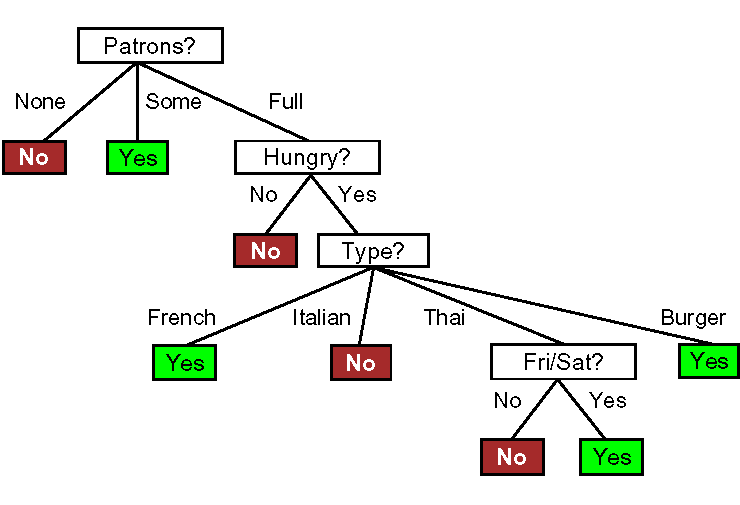


* We now calculate the number of positive chances and negative chances for each case:
* For ‘Alternative?’



* For ‘Raining?’
* For Patrons:
* We see that ,Patrons has highest entropy change from all.
* So we choose Patrons as root node and construct the decision tree.

The decision tree induced from the 12-example training set:



True tree:

