Data Mining
(EECS 4412)

Decision Tree Learning

Aijun An
Department of Electrical Engineering and Computer Science
York University

Outline

- Overview of classification
- Basic concepts in decision tree learning
  - Data representation in decision tree learning
  - What is a decision tree?
    - Decision tree representation
- How to learn a decision tree from data
  - Basic decision tree learning algorithm
  - How to select best attribute
  - Pruning decision tree
- Other issues involved in decision tree learning

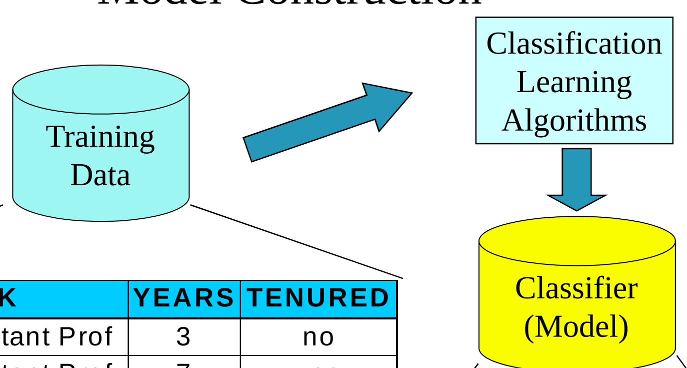
2

Classification—A Two-Step Process

- Model construction (i.e., learning):
  - Learn a model from a <u>*training set*</u> (*a set of pre-classified training examples*) -- supervised learning
  - The model can be represented as *classification rules*, *decision trees*, *neural networks*, *mathematical formulae*, *etc.*
- Model usage (i.e., prediction or classification):
  - Classify future or unknown objects -- main purpose
  - Test the learned model on a <u>*test set*</u> (*another set of pre-classified examples*) to estimate accuracy of the model
    - The known class label of a test example is compared with the classification result from the model
    - Accuracy rate is the percentage of test examples that are correctly classified by the model
    - Test set is independent of training set, otherwise the testing result is not reliable

3

Classification Process (1): Model Construction

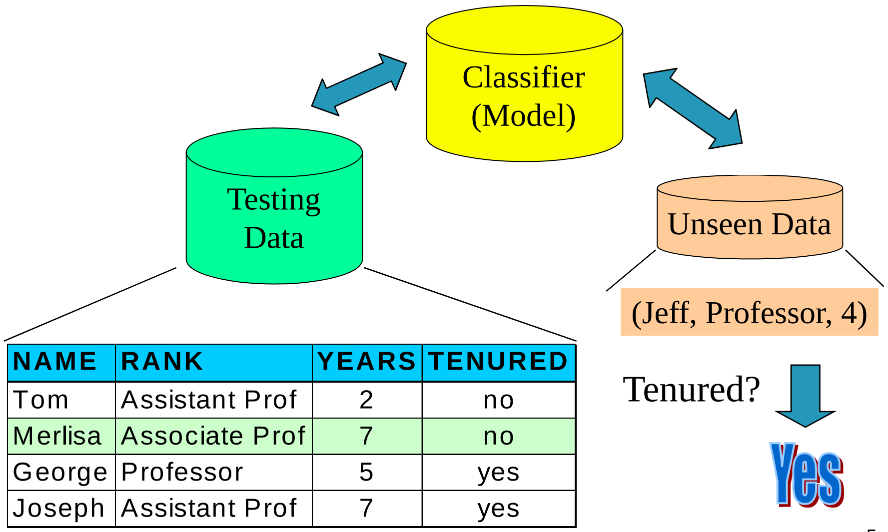


NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

4

Classification Process (2): Use the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

5

Classification Learning Techniques

- Decision tree learning
- Decision rule learning
- Bayesian classification
- Neural networks
- K-nearest neighbor method
- Support vector machines (SVM)
- Genetic algorithms
- etc.

6

Decision Tree Learning

- Objective of decision tree learning
  - Learn a decision tree from a set of training data
  - The decision tree can be used to classify new examples
- Decision tree learning algorithms
  - ID3 (Quinlan, 1986)
  - C4.5 (Quinlan, 1993)
  - CART (Breiman, Friedman, *et. al.* 1983)
  - CHAID (Kass, 1980)
  - QUEST(Loh and Shih, 1997)
  - etc.

7

Representation of Training Examples

Condition attributes (Outlook, Temperature, Humidity, Wind)

Class/Target/Decision attribute (PlayTennis)

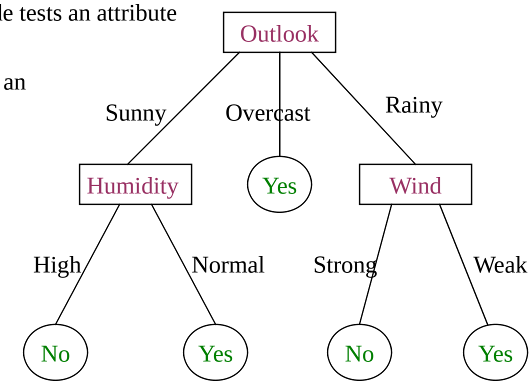
Training examples or cases

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	Rainy	Mild	High	Weak	Yes
D5	Rainy	Cool	Normal	Weak	Yes
D6	Rainy	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rainy	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rainy	Mild	High	Strong	No

8

Decision Tree Representation

- A decision tree: representation of classification knowledge
  - Each non-leaf (internal) node tests an attribute (Outlook, Humidity, Wind)
  - Each branch corresponds to an attribute value
  - Each leaf node assigns a classification



This tree classifies days according to whether or not they are suitable for playing tennis.

- Classification
  - A new case is classified by testing the case against the nodes from the root to a leaf node. The classification associated with the leaf is returned. For example,

⟨Outlook = Sunny, Temperature = Mild, Humidity = high, Wind = Strong⟩ → No

9

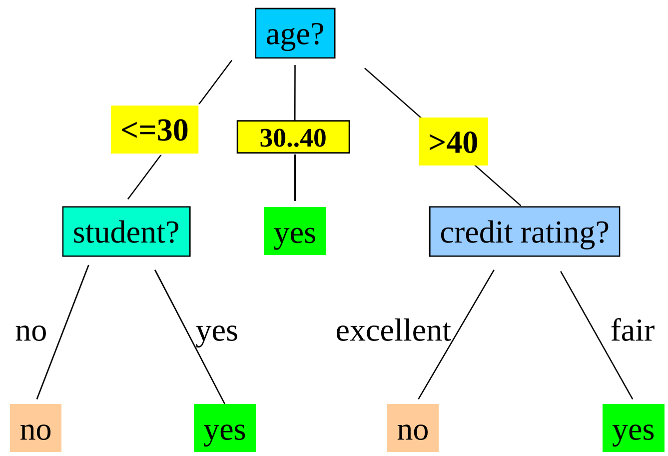
Another Example of Training Dataset

This follows an example from Quinlan's ID3

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

10

Output: A Decision Tree for *“buys_computer”*



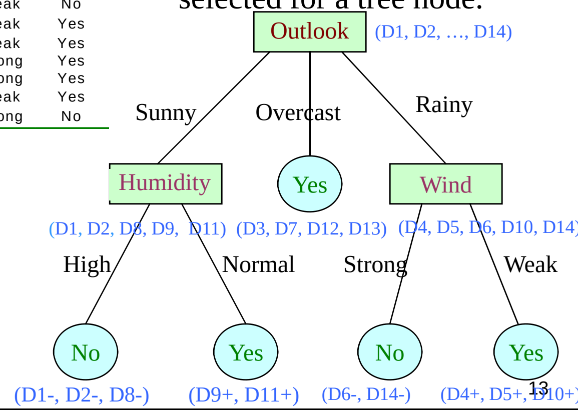
Outline

- Overview of classification
- Basic concepts in decision tree learning
  - Data representation in decision tree learning
  - What is a decision tree?
    - Decision tree representation
- <u>How to learn a decision tree from data</u>
  - Basic decision tree learning algorithm
  - How to select best attribute
  - Pruning decision tree
- Other issues involved in decision tree learning

Top-Down Induction of a Decision Tree

Day	Outlook	Tempe-rature	Humi-dity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rainy	Mild	High	Weak	Yes
D5	Rainy	Cool	Normal	Weak	Yes
D6	Rainy	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rainy	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rainy	Mild	High	Strong	No

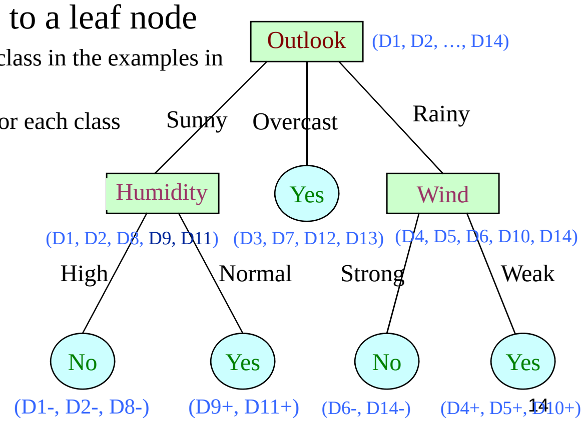
- Idea: divide and conquer
- Method: recursive partitioning of training data according to the attribute selected for a tree node.



13

Three Issues in Decision Tree Induction

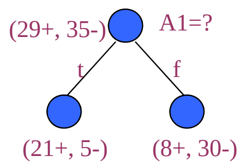
- How to select an attribute for a node
- When to declare a node terminal
  - a naïve, but not robust method: when node is pure, stop growing.
- How to assign a class to a leaf node
  - Assign the most common class in the examples in the node to the node
  - Or output the probability for each class

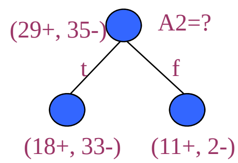


14

How to Select Attribute

- Which attribute is the best attribute given a set of attributes and a set of examples?





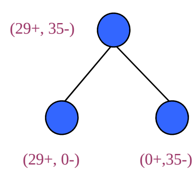
- Many selection criteria, including:
  - Information gain (Quinlan, 1983; used in ID3)
  - Gain ratio (Quinlan, 1986; used in C4.5)
  - Gini index (Breiman, 1984; used in CART)
  - Chi-square statistic (Kass,1980; used in CHAID. Mingers, 1989)
  - Binarization (Bratko & Kononenko, 86)
  - Normalized information gain (Lopez de Mantaras, 91)

15

Information Gain

- Objective:
  - Select an attribute so that the data in each of the descendant subsets are the "purest".

An ideal split:



However, there may not be an attribute in the data set leading to such a split.

- Based on the concept of *entropy*
  - *Entropy* is a measure, commonly used in information theory, that characterizes the impurity (uncertainty, chaos) of an arbitrary collection of examples.

16

Entropy

- Given a set S of examples and k classes ($C_1, \ldots, C_k$), the *entropy* of S with respect to the k classes is defined as:

$$Entropy(S) = -\sum_{i=1}^{k} P(C_i)\log_2(P(C_i))$$

where $P(C_i)$ is the probability of examples in S that belong to C_i.

- The bigger *Entropy(S)* is, the more impure S is.
- Examples:
  - If all examples in S belong to the same class (i.e., S is pure), *Entropy(S)*=0.
  - If half of the examples in S belong to class 1 and the other half belong to class 2, *Entropy(S)*=1.
  - Suppose 9 examples are in class 1 and 5 examples in class 2, *Entropy*(*S*)= -(9/14)$\log_2$(9/14) - (5/14)$\log_2$(5/14) = 0.940
  - If the examples are uniformly distributed in 3 classes, *Entropy*(S)= - ((1/3)$\log_2$(1/3))×3=$\log_2 3$ =1.59

17

Information Gain (*Cont'd*)

- An attribute-selection criterion:
  - Used to choose an attribute to split a data set
- Assume that attribute A has m values.
  - Using A, data set S is split into $S_1, S_2, \ldots, S_m$.
- Information Gain

  Gain(S, A) = expected reduction in entropy due to partitioning S on attribute A

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{m} \frac{|S_i|}{|S|} Entropy(S_i)$$

where $|S|$ is the number of examples in set S, and $|S_i|$ is the number of examples in S_i.

18

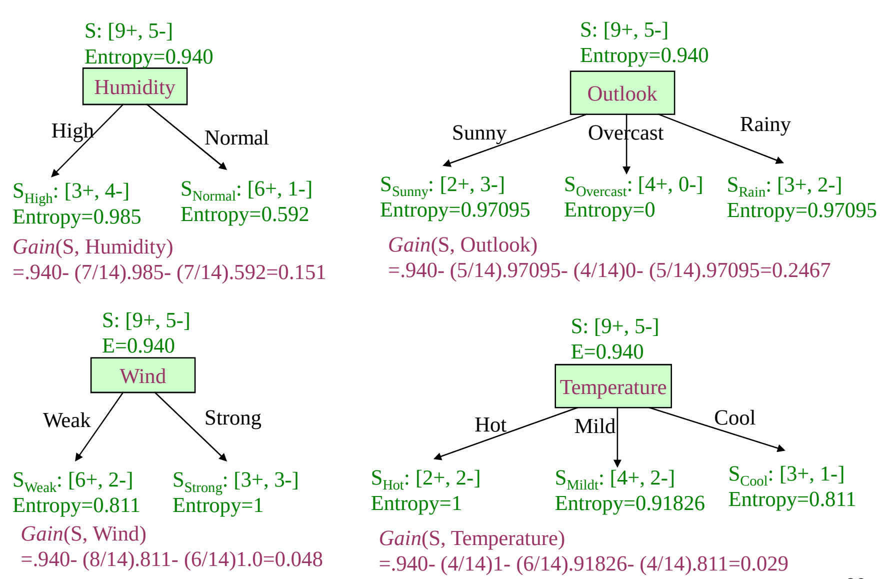
An illustrative example

- 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	Rainy	Mild	High	Weak	Yes
D5	Rainy	Cool	Normal	Weak	Yes
D6	Rainy	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rainy	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rainy	Mild	High	Strong	No

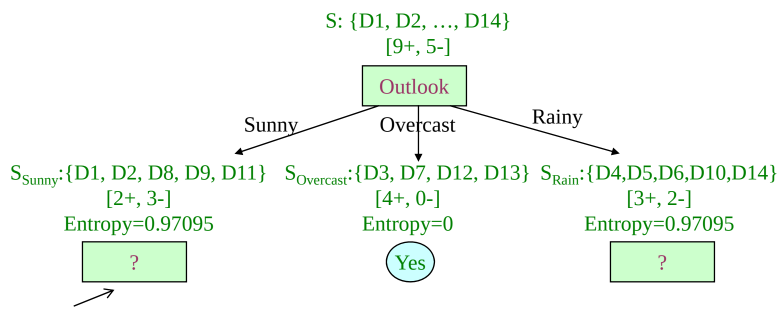
19

Which attribute is the best for the root?



20

An illustrative example (Cont’d.)



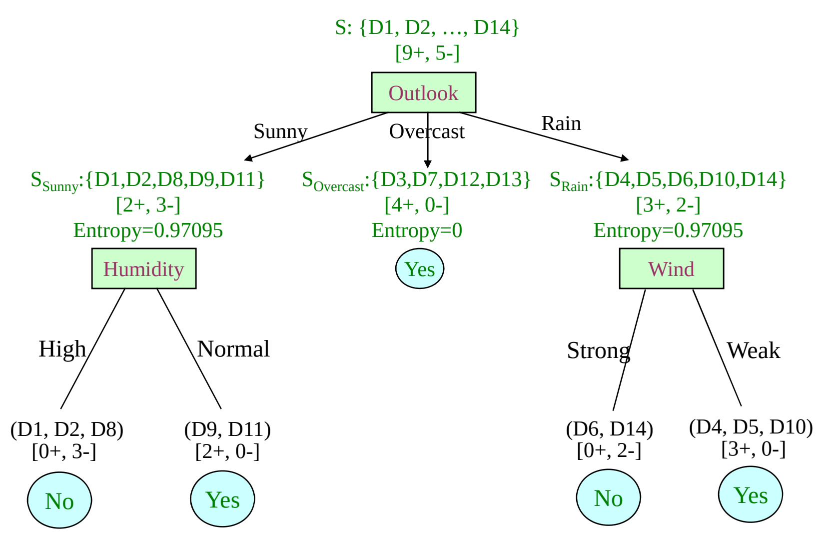
Which attribute should be tested here, Humidity, Temperature, or Wind?

$Gain(S_{sunny}$, Humidity) = .97095- (3/5)0.0- (2/5)0.0 = 0.97095
$Gain(S_{sunny}$, Temperature) = .97095- (2/5)0.0- (2/5)1.0- (1/5)0.0 = 0.57095
$Gain(S_{sunny}$, Wind) = .97095- (2/5)1.0- (3/5).918 = 0.02015

Therefore, **Humidity** is chosen as the next test attribute for the left branch.

21

An illustrative example (Cont’d.)



22

Basic Decision Tree Learning Algorithm

1. Select the "best" attribute *A* for the root node

2. Create new descendents of the node according to the values of *A*:

3. Sort training examples to the descendent nodes.

4. For each descendent node,

when to terminate the recursive process

- if the training examples associated with the node belong to the same class, the node is marked as a leaf node and labeled with the class
- else if there are no remaining attributes on which the examples can be further partitioned, the node is marked as a leaf node and labeled with the most common class among the training cases for classification;
- else if there is no example for the node, the node is marked as a leaf node and labeled with the majority class in its parent node.
- otherwise, recursively apply the process on the new node.

23