Classification and Prediction

Classification and Prediction

- What is classification? What is regression?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Scalable decision tree induction

Classification vs. Prediction

- Classification:
  - predicts categorical class labels
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Regression:
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical Applications
  - credit approval
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis

Why Classification? A motivating application

- Credit approval
  - A bank wants to classify its customers based on whether they are expected to pay back their approved loans
  - The history of past customers is used to train the classifier
  - The classifier provides rules, which identify potentially reliable future customers
  - Classification rule:
    - If age = "31...40" and income = high then credit_rating = excellent
  - Future customers
    - Paul: age = 35, income = high $\Rightarrow$ excellent credit rating
    - John: age = 20, income = medium $\Rightarrow$ fair credit rating

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction: training set
  - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test samples is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set, otherwise over-fitting will occur

Classification Process (1): Model Construction

Training Data

Classification Algorithms

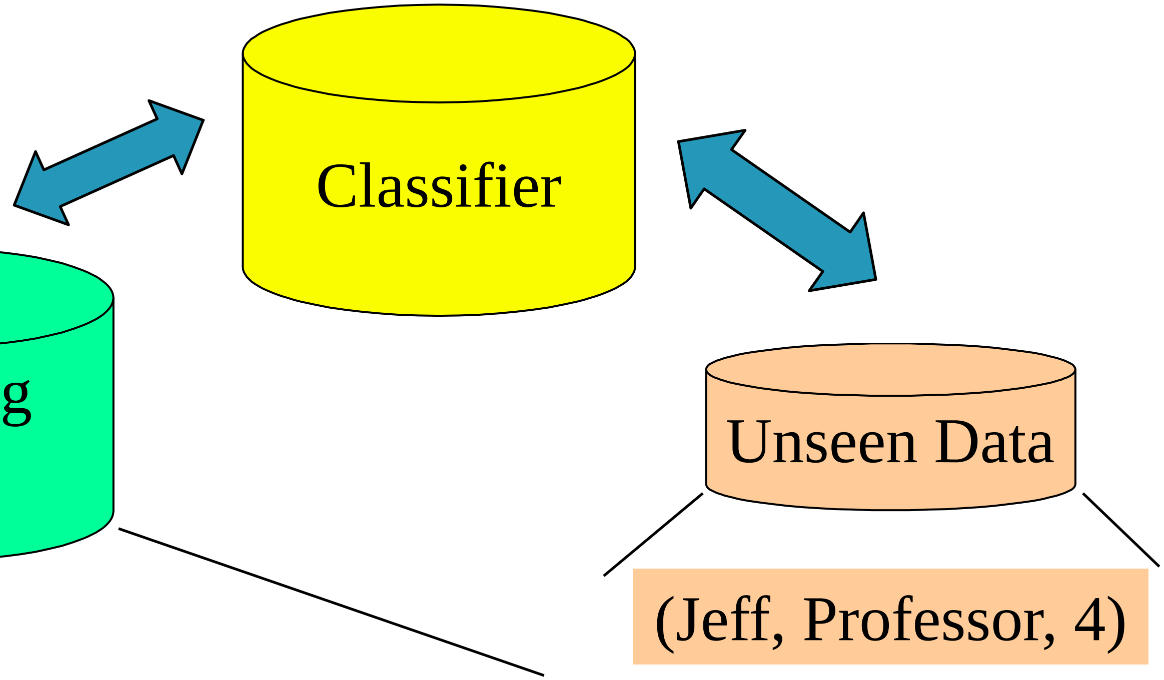
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classifier (Model)

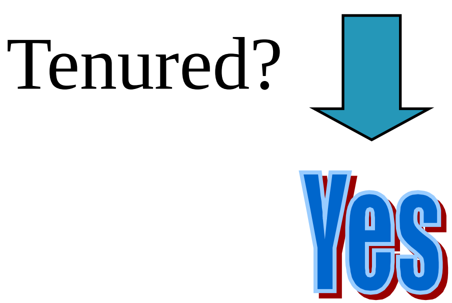
IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Classification Process (2): Use the Model in Prediction

Accuracy=?



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



Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Issues regarding classification and prediction (1): Data Preparation

- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
- Data transformation
  - Generalize and/or normalize data
    - numerical attribute income $\Rightarrow$ categorical {low,medium,high}
    - normalize all numerical attributes to [0,1)

Issues regarding classification and prediction (2): Evaluating Classification Methods

- Predictive accuracy
- Speed
  - time to construct the model
  - time to use the model
- Robustness
  - Resistant to handle noise, extreme and missing values
- Scalability
  - efficiency in disk-resident databases
- Interpretability:
  - understanding and insight provided by the model
- Goodness of rules (quality)
  - decision tree size
  - compactness of classification rules

Classification by Decision Tree Induction

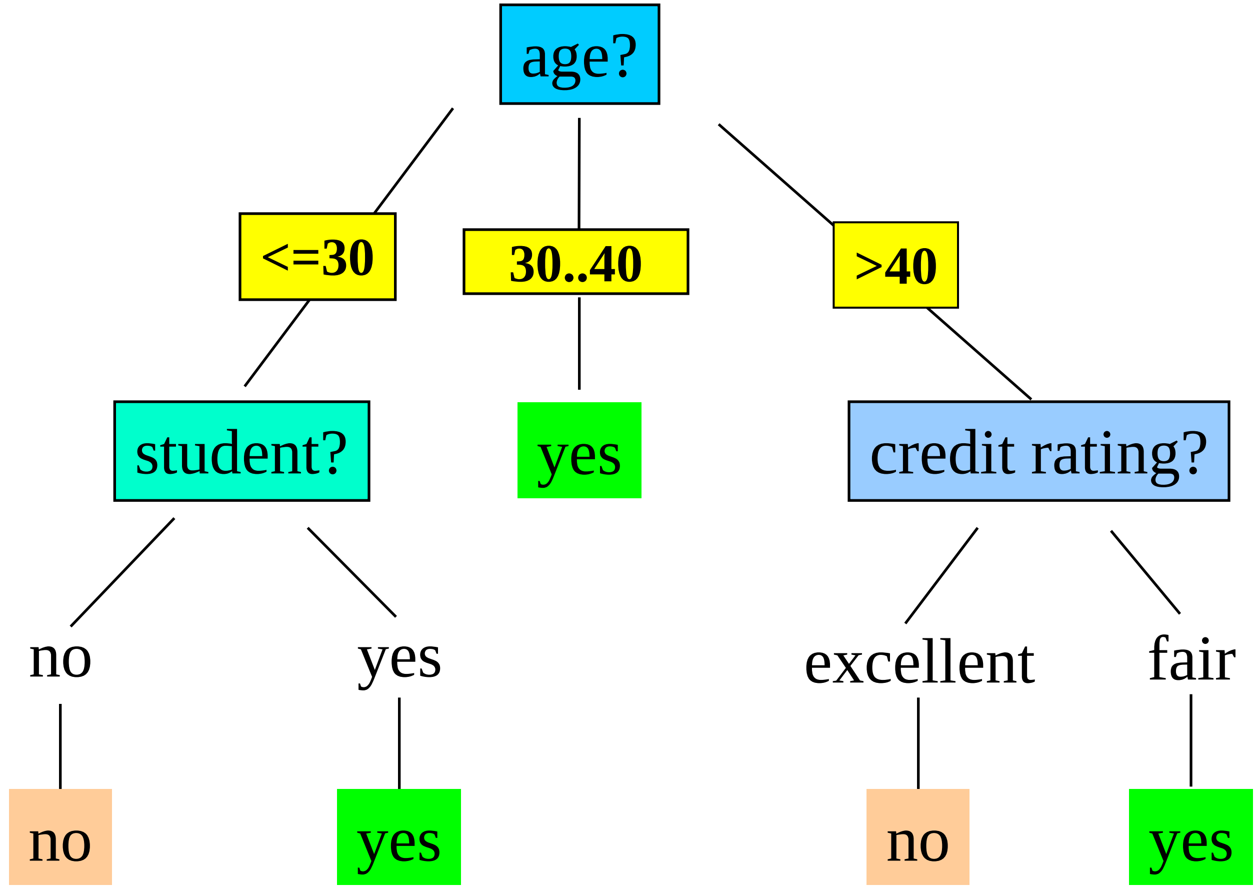
- Decision tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

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
31…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

Output: A Decision Tree for *“buys_computer”*



Algorithm for Decision Tree Induction

In early 1980s, J. Ross Quinlan, a researcher in machine learning, developed **ID3** (Iterative Dichotomiser). Quinlan later presented **C4.5** (a successor of ID3), which became a benchmark. In 1984, a group of statisticians (L. Breiman, J. Friedman, R. Olshen, and C. Stone) published the book *Classification and Regression Trees* (**CART**), which described the generation of binary decision trees

Algorithm for Decision Tree Induction

ID3 Iterative Dichotomiser. The algorithm is called with three parameters: D, attribute list, and Attribute selection method. We refer to D as a data partition. Second parameter is the attributes describing the tuples. Attribute selection method specifies a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class.

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Samples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Algorithm for Decision Tree Induction (pseudocode)

Algorithm GenDecTree(Sample S, Attlist A)

1. create a node N
2. If all samples are of the same class C then label N with C; terminate;
3. If A is empty then label N with the most common class C in S (majority voting); terminate;
4. Select a$\in$A, with the highest information gain; Label N with a;
5. For each value v of a:
    a. Grow a branch from N with condition a=v;
    b. Let S_v be the subset of samples in S with a=v;
    c. If S_v is empty then attach a leaf labeled with the most common class in S;
    d. Else attach the node generated by GenDecTree(S_v, A-a)

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i, D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Info. gain for Attribute=Income

$$\text{Info(D)} = -\frac{9}{14}\, log_2(\frac{9}{14}) - \frac{5}{14}\, log_2(\frac{5}{14}) = 0.94028$$

For Income=high

$$\text{Info(D)} = -\frac{2}{4}\, log_2(\frac{2}{4}) - \frac{2}{4}\, log_2(\frac{2}{4}) = 1$$

For Income=Medium

$$\text{Info(D)} = -\frac{4}{6}\, log_2(\frac{4}{6}) - \frac{2}{6}\, log_2(\frac{2}{6}) = 0.918296$$

For Income=Low

$$\text{Info(D)} = -\frac{3}{4}\, log_2(\frac{3}{4}) - \frac{1}{4}\, log_2(\frac{1}{4}) = 0.811278$$

Info. gain for Attribute=Income

$$\text{Info(D)}= -\frac{9}{14}\,log_2(\frac{9}{14}) - \frac{5}{14}\,log_2(\frac{5}{14})\text{=0.94028}$$

$$\text{Info}(D_{\text{Income}})=\frac{4}{14}\,X\,1 + \frac{6}{14}X0.918296+$$

$$\frac{4}{14}\text{ X 0.811278}$$

$$=0.911063$$

$$\text{Gain=Info(D)-Info}(D_{\text{Income}})$$

$$=0.94028-0.911063 = 0.029223$$

Splitting the samples using *age*

age?

<=30

income	student	credit_rating	buys_computer
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

30...40

income	student	credit_rating	buys_computer
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

labeled yes

>40

income	student	credit_rating	buys_computer
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

Info. gain for Attribute=Age

- Info(D)= $-\frac{9}{14}\,log_2(\frac{9}{14}) - \frac{5}{14}\,log_2(\frac{5}{14})$=0.94028

For Age is <=30

Info(D)=$-\frac{3}{5}\,log_2(\frac{3}{5}) - \frac{2}{5}\,log_2(\frac{2}{5})$=0.97095

For Age is 31-40

$$\text{Info}(D) = -\frac{4}{4}\,log_2(\frac{4}{4}) = 0$$

For Age is >40

$$\text{Info}(D) = -\frac{3}{5}\,log_2(\frac{3}{5}) - \frac{2}{5}\,log_2(\frac{2}{5}) = 0.97095$$

Info. gain for Attribute=Age

- Info(D)= $-\frac{9}{14}\, log_2(\frac{9}{14}) - \frac{5}{14}\, log_2(\frac{5}{14})$=0.94028

Info(D_{Age})=$\frac{5}{14}\ X\ 0.97095 + \frac{4}{14} X\ 0 +$

$\frac{5}{14}$ X 0.97095

=0.693536

Gain=Info(D)-Info(D_{Age})

=0.94028-0.693536 =0.24675

Info. gain for Attribute=Age

- Info(D)= $-\frac{9}{14}\ log_2(\frac{9}{14})\ -\frac{5}{14}\ log_2(\frac{5}{14})$=0.94028

For Age is <=30

Info(D)=$-\frac{3}{5}\ log_2(\frac{3}{5})\ -\frac{2}{5}\ log_2(\frac{2}{5})$=0.97095

For Age is 31-40

$\mathrm{Info(D)} = -\frac{4}{4}\ log_2(\frac{4}{4})$ = 0

For Age is >40

$\mathrm{Info(D)} = -\frac{3}{5}\ log_2(\frac{3}{5})\ -\frac{2}{5}\ log_2(\frac{2}{5})$ = 0.97095

Info. gain for Attribute=Age

- Info(D)= $-\frac{9}{14}\, log_2(\frac{9}{14}) - \frac{5}{14}\, log_2(\frac{5}{14})$=0.94028

Info(D_{Age})=$\frac{5}{14}\ X\ 0.97095 + \frac{4}{14} X\ 0+$

$\frac{5}{14}$ X 0.97095

=0.693536

Gain=Info(D)-Info(D_{Age})

=0.94028-0.693536 =0.24675

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Computing Information-Gain for Continuous-Value Attributes

- Let attribute A be a continuous-valued attribute
- Must determine the *best split point* for A
  - Sort the value A in increasing order
  - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
  - The point with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
  - D1 is the set of tuples in D satisfying A ≤ split-point, and D2 is the set of tuples in D satisfying A > split-point

Questions and Answers