PE5:DATA MINING

Classification

Module - III

Dr. Bashirahamad F. Momin
CSE Dept., Walchand COE, Sangli.

Basic Concept

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

Prediction Problems: Classification vs. Numeric Prediction

- **Classification**
  - predicts categorical class labels (discrete or nominal)
  - 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
- **Numeric Prediction**
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical applications
  - Credit/loan approval:
  - Medical diagnosis: if a tumor is cancerous or benign
  - Fraud detection: if a transaction is fraudulent
  - Web page categorization: which category it is

Classification - Definition

- **Method of categorizing or assigning class labels to a pattern set under supervision of teacher : *Supervised Learning.***

- **The decision boundaries are generated to discriminate between patterns of different classes.**

- **Prediction of class from set of samples**

- **Different methods :**
  - ***Decision Trees***
  - ***Probabilistic or generative models***
  - ***Nearest Neighbor classifier***
  - ***Artificial Neural Network (ANN)***

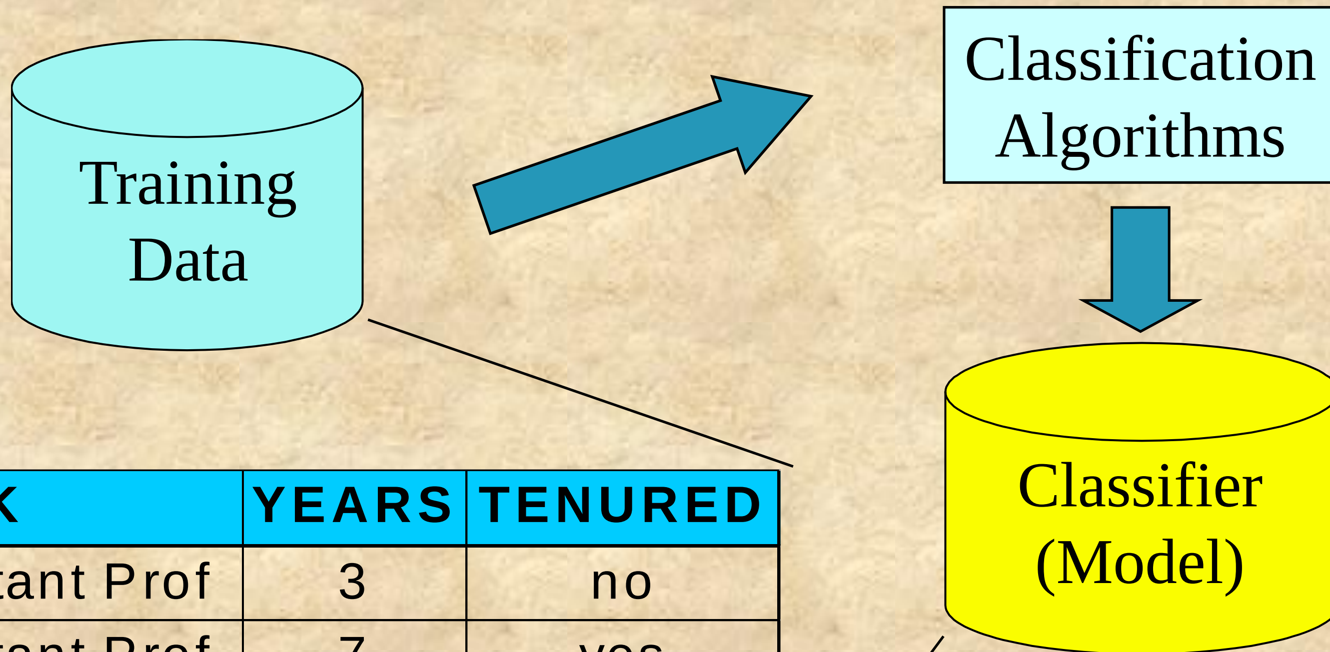
Classification Problem

- Given a database $D=\{t_1,t_2,\ldots,t_n\}$ and a set of classes $C=\{C_1,\ldots,C_m\}$, the ***Classification Problem*** is to define a mapping $f:D\rightarrow C$ where each t_i is assigned to one class.
- Actually divides D into ***equivalence*** *classes*.
- ***Prediction*** is similar, but may be viewed as having infinite number of classes.

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 is **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 sample 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
  - If the accuracy is acceptable, use the model to **classify data** tuples whose class labels are not known

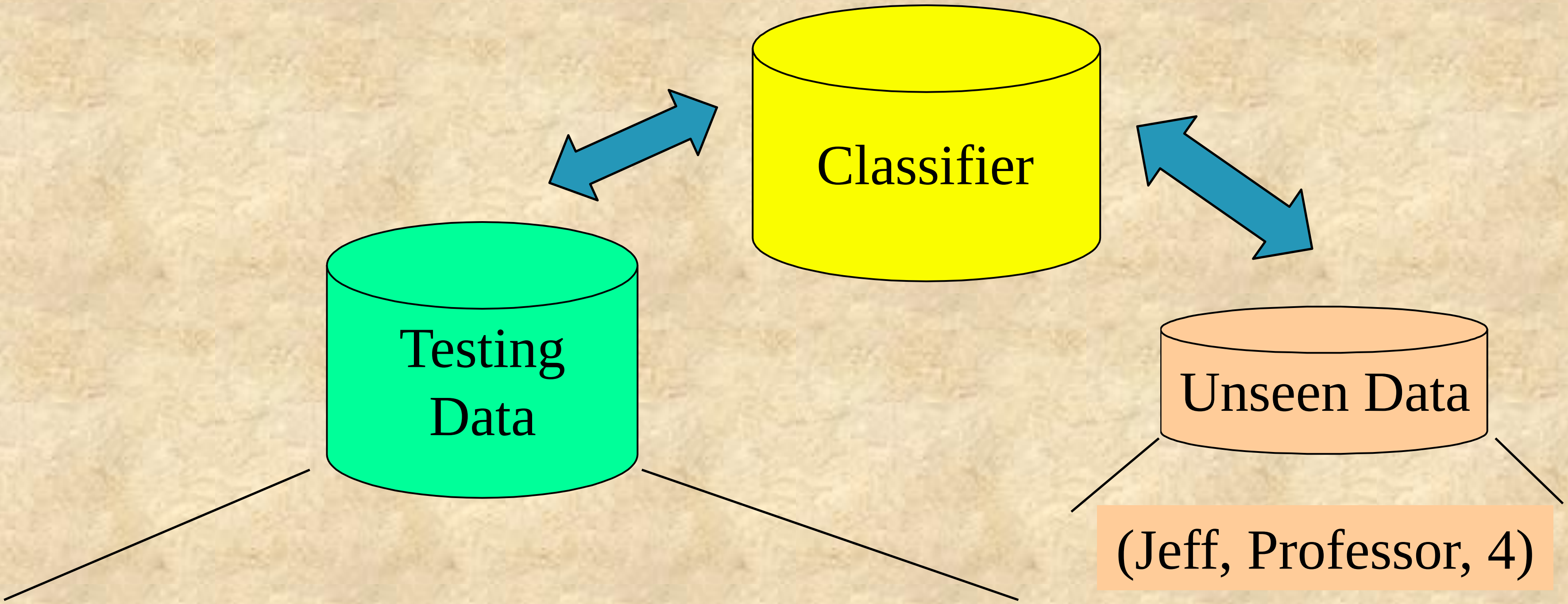
Process (1): Model Construction



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’

Process (2): Using 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

Tenured?

Yes

Classification Examples

- Teachers classify students' grades as A, B, C, D, or F.
- Identify mushrooms as poisonous or edible.
- Predict when a river will flood.
- Identify individuals with credit risks.
- Speech recognition
- Pattern recognition

Classification Ex: Letter Recognition

View letters as constructed from 5 components:

Featues : Strokes , Tees , joint , end points etc

 Letter A

 Letter B

 Letter C

 Letter D

 Letter E

 Letter F

Decision Tree Induction

Dr. Bashirahamad F. Momin
CSE Dept., Walchand COE, Sangli.

Introduction

- learning of decision trees from class-labeled training tuples.
- A **decision tree** is a flowchart-like tree structure, where each **internal node** (nonleaf node) denotes a test on an attribute, each **branch** represents an outcome of the test, and each **leaf node** (or *terminal node*) holds a class label.
- The topmost node in a tree is the **root** node.

...Cont Introduction

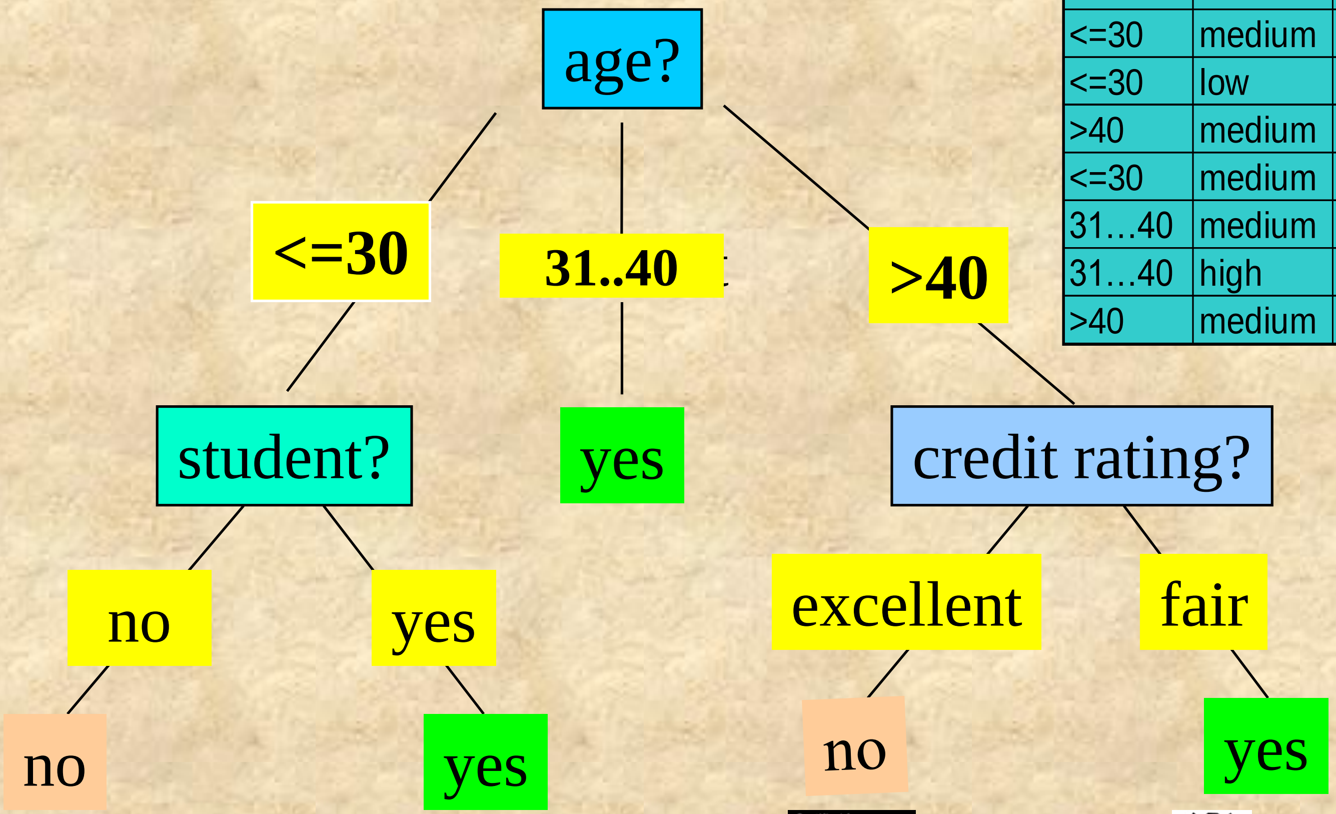
- Internal nodes are denoted by ***rectangles***, and leaf nodes are denoted by ***ovals***.
- Some decision tree algorithms produce only ***binary*** trees (where each internal node branches to exactly two other nodes), whereas others can produce ***nonbinary*** trees.

Dr. Bashirahamad F. Momin
Department of Computer Science & Engineering
Walchand College of Engg, Sangli, INDIA

Decision Tree Induction: An Example

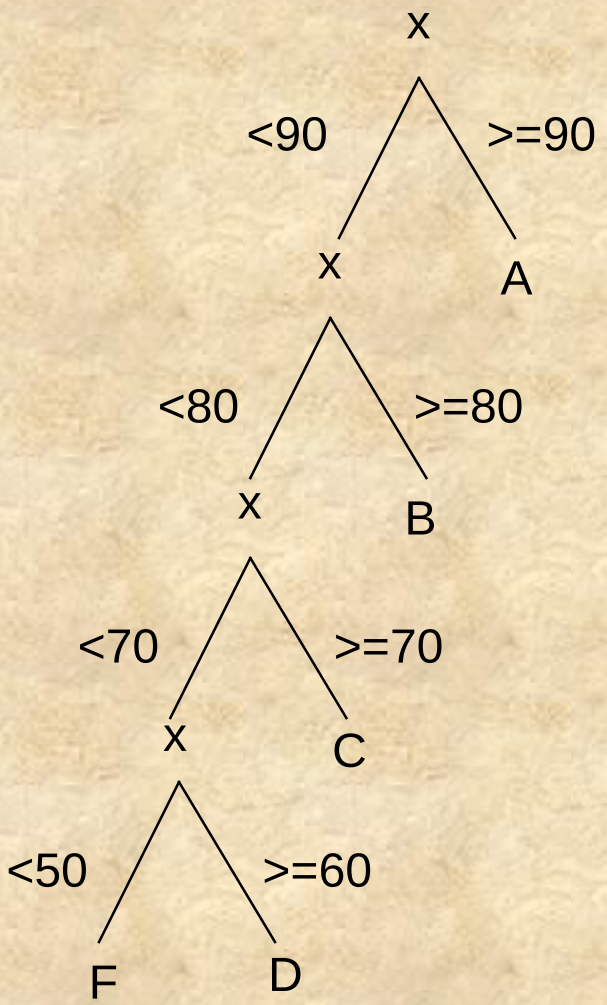
- Training data set: Buys_computer
- Resulting tree:

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



Decision Tree for Grading

- If x >= 90 then grade =A.
- If 80<=x<90 then grade =B.
- If 70<=x<80 then grade =C.
- If 60<=x<70 then grade =D.
- If x<50 then grade =F.



How are the decision trees used for classification ?

- Given a tuple, ***X***, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree.
- A path is traced from the root to a leaf node, which holds the class prediction for that tuple – ***tree traversal***.
- Decision trees can easily be converted to **classification rules**
- #rules = #path from root to leaf node

Dr. Bashirahamad F. Momin
Department of Computer Science & Engineering
Walchand College of Engg. Sangli, INDIA

Why are decision tree classifiers so popular?

- The construction of decision tree classifiers does not require any domain knowledge or parameter setting.
- appropriate for exploratory knowledge discovery.
- can handle multidimensional data.
- representation of acquired knowledge in tree form is intuitive and easy to assimilate/understand by humans
- decision tree classifiers have good accuracy.

Dr. Bashirahamad F. Momin
Department of Computer Science & Engineering
Walchand College of Engg. Sangli, INDIA

How to build Decision Tree ?

Dr. Bashirahamad F. Momin
CSE Dept., Walchand COE, Sangli.

History

- During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as **ID3** (Iterative Dichotomiser).
- Quinlan later presented **C4.5** (a successor of ID3), which became a benchmark to which newer supervised learning algorithms are often compared.
- 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.
- ID3 and CART were invented independently of one another at
- around the same time, yet follow a similar approach for learning decision trees from
- training tuples.
- **ID3 , C4.5 & CART** algorithms becomes de-facto standards for decision tree induction

Dr. Bashirahamad F. Momin
Department of Computer Science & Engineering
Walchand College of Engg, Sangli, INDIA

Greedy Algorithm

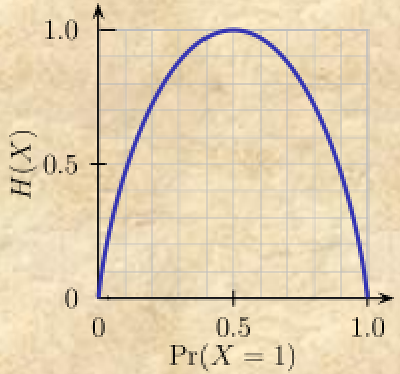
- **ID3**, **C4.5**, and **CART** adopt a greedy (i.e., nonbacktracking) approach.
- decision trees are constructed in a top-down recursive divide-and-conquer manner.
- It starts with a training set of tuples and their associated class labels.
- The training set is recursively partitioned into smaller subsets as the tree is being built.

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)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure – **Entropy , Gain Ratio & Gini Index**)
- 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

Entropy

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random variable
  - Calculation: For a discrete random variable *Y* taking *m* distinct values $\{y_1, \ldots, y_m\}$,
    - $H(Y) = -\sum_{i=1}^{m} p_i \log(p_i)$, where $p_i = P(Y = y_i)$
  - Interpretation:
    - Higher entropy => higher uncertainty
    - Lower entropy => lower uncertainty
- Conditional Entropy
  - $H(Y|X) = \sum_x p(x) H(Y|X = x)$



m = 2

Attribute Selection using 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:

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

- **Information gained** by branching on attribute A

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

Gain Ratio for Attribute Selection (C4.5)

- Information gain measure is biased towards attributes with a large number of values
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

  – GainRatio(A) = Gain(A)/SplitInfo(A)

- Ex.

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

  – gain_ratio(income) = 0.029/1.557 = 0.019

- The attribute with the maximum gain ratio is selected as the splitting attribute

Gini Index (CART, IBM IntelligentMiner)

- If a data set D contains examples from n classes, gini index, *gini*(D) is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class j in D

- If a data set D is split on A into two subsets D_1 and D_2, the *gini* index *gini*(D) is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Comparing Attribute Selection Measures

- The three measures, in general, return good results but
  - **Information gain**:
    - biased towards multivalued attributes
  - **Gain ratio**:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - **Gini index**:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions

Example : Building Decision Tree

Training Data Set

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Rule Based Classification

Dr. Bashirahamad F. Momin
CSE Dept., Walchand COE, Sangli.

Metrics for Evaluating Classifier Performance

Dr. Bashirahamad F. Momin
CSE Dept., Walchand COE, Sangli.