Topic 2: Classification
PartI
Classification

- What is classification? Classification vs clustering
- Issues regarding classification
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary
Supervised Learning vs. Unsupervised Learning

- **Supervised learning**
  - The training data are accompanied by labels indicating the class of the observations
  - Training: classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute is usually a model development
  - Goal: uses the model in classifying new data
  - Example:
    - Binary Images $\rightarrow$ \{'A','B','C'…'Z'\}
    - Disease Symptoms $\rightarrow$ \{Diseases\},
    - Documenter Vectors $\rightarrow$ category,
    - Web pages $\rightarrow$ \{relevant, non relevant\}

- **Unsupervised learning**
  - The class labels of training data is unknown
  - Given a set of observations, establish the existence or cluster in the data
Two step process of classification

- **Step 1: 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
- **Step 2: 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
Step 1: Model Construction

### Training Data

<table>
<thead>
<tr>
<th>NAME</th>
<th>RANK</th>
<th>YEARS</th>
<th>TENURED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>Assistant Prof</td>
<td>3</td>
<td>no</td>
</tr>
<tr>
<td>Mary</td>
<td>Assistant Prof</td>
<td>7</td>
<td>yes</td>
</tr>
<tr>
<td>Bill</td>
<td>Professor</td>
<td>2</td>
<td>yes</td>
</tr>
<tr>
<td>Jim</td>
<td>Associate Prof</td>
<td>7</td>
<td>yes</td>
</tr>
<tr>
<td>Dave</td>
<td>Assistant Prof</td>
<td>6</td>
<td>no</td>
</tr>
<tr>
<td>Anne</td>
<td>Associate Prof</td>
<td>3</td>
<td>no</td>
</tr>
</tbody>
</table>

### Classification Algorithms

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’
Step 2: Use the Model in Prediction

<table>
<thead>
<tr>
<th>NAME</th>
<th>RANK</th>
<th>YEARS</th>
<th>TENURED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom</td>
<td>Assistant Prof</td>
<td>2</td>
<td>no</td>
</tr>
<tr>
<td>Merlisa</td>
<td>Associate Prof</td>
<td>7</td>
<td>no</td>
</tr>
<tr>
<td>George</td>
<td>Professor</td>
<td>5</td>
<td>yes</td>
</tr>
<tr>
<td>Joseph</td>
<td>Assistant Prof</td>
<td>7</td>
<td>yes</td>
</tr>
</tbody>
</table>

Unseen Data

(Jeff, Professor, 4)

Testing Data

Classifier

Tenured?
Classification

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary
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
Evaluating Classification Methods

- Predictive accuracy
- Speed and scalability
  - time to construct the model
  - time to use the model
- Robustness
  - the ability of the model to handle noise and missing values
- Scalability
  - the ability of the model to deal with huge volume of data
- Interpretability:
  - The ability of understanding and insight provided by the model
- Goodness of rules
  - decision tree size
  - compactness of classification rules
Classification

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by back propagation
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary
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 instances are at the root
    - Partition instances 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 instances against the decision tree
### Training Dataset

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit_rating</th>
<th>( \text{buys_computer} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \leq 30 )</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>( \leq 30 )</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31\ldots40</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31\ldots40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>( \leq 30 )</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>( \leq 30 )</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>( \leq 30 )</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31\ldots40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31\ldots40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
</tbody>
</table>

This follows an example from Quinlan’s ID3.
Output: A Decision Tree for “buys_computer”

- **age?**
  - **<=30**
    - **student?**
      - **no**
        - **no**
      - **yes**
        - **yes**
  - **30..40**
    - **yes**
  - **>40**
    - **credit rating?**
      - **excellent**
      - **fair**
Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down divide-and-conquer manner
  - At start, all the training instances are at the root
  - Attributes are discrete (if continuous-valued, they are discretized in advance)
  - Instances are partitioned recursively based on selected attributes
    - A selected attribute is also called a split or a test
  - Select the next test attribute that has the highest information gain
  - Partitioning terminates if
    - 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 instances left
Information Gain

Entropy

For probability $p_1, ..., p_n$, where $\Sigma p_i = 1$

Entropy $E(p_1, ..., p_n) = - \Sigma p_i \log p_i$

Zero entropy: Only in one group, or only one of $p_i$ is 1

Large entropy: equally distributed

Assume there are two classes, class 1 and class 2

- Let the set of instances $S$ contain $S_1$ elements of class 1 and $S_2$ elements of class 2
- The amount of information, needed to decide if an arbitrary instance in $S$ belongs to class 1 or 2 is the entropy of $p_1$ and $p_2$, where $p_1 = S_1/S$ and $p_2 = S_2/S$

$$I(S_1, S_2) = - \frac{S_1}{S} \log_2 \frac{S_1}{S} - \frac{S_2}{S} \log_2 \frac{S_2}{S}$$
Information Gain in Decision Tree

- S: training set
- Goal: Classifying S into $C_1, C_2, \ldots, C_m$, $s_i = |\{X \in C_i\}|$
- The information of classifying into C
  \[
  I(s_1, \ldots, s_m) = - \sum_{i=1}^{m} \frac{C_i}{S} \log \frac{C_i}{S}
  \]
- $a_j$: values of attribute A, $(j=1,\ldots,v)$
- Let $S_j = \{X | \text{attributes of } X \text{ is } a_j\}$
- $S_{ij} = \{X \in C_i | \text{attributes of } X \text{ is } a_j\}$
- $E(A) = \sum_{j=1}^{v} \frac{s_{1j} + s_{2j} + \ldots + s_{mj}}{S} I(s_{1j}, \ldots, s_{mj})$
  where $I(s_{lj}, s_{lj}, \ldots, s_{mj})$ is the entropy of $S_j$
  
  The smaller the $E(A)$ is, the greater the purity of the subset partition is
- The encoding information that would be gained by branching on $A$
  \[
  \text{Gain (A)} = I(s_1, \ldots, s_m) - E(A)
  \]
- The bigger the information gain, the better the attribute A is
Attribute Selection by Information Gain Computation

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- \( I(p, n) = I(9, 5) = 0.940 \)
- Compute the entropy for \( age \):

\[
E(\text{age}) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) \\
+ \frac{5}{14} I(3,2) = 0.69
\]

Hence

\[
\text{Gain}(\text{age}) = I(p, n) - E(\text{age})
\]

Similarly

\[
\text{Gain}(\text{income}) = 0.029 \\
\text{Gain}(\text{student}) = 0.151 \\
\text{Gain}(\text{credit_rating}) = 0.048
\]
Attribute Selection Measure

- **Information gain (ID3/C4.5)**
  Shortcoming: prefer attributes with lots of values.

- **Gain ratio**
  - Gain ratio(A) = Gain(A)/InfoVal(A), where
    
    \[
    \text{InfoVal} (A) = -\sum_{i=1}^{v} \frac{S'_i}{S} \log \frac{S'_i}{S}
    \]
    
    and attribute A partition S into \{S_1,...,S'_v\}
  
  - Select attribute A which maximize the gain ratio

- **Gini index** (IBM IntelligentMiner, CART (Classification and Regression Tree))
  - All attributes are assumed continuous-valued
  
  - Assume there exist several possible split values for each attribute
  
  - May need other tools, such as clustering, to get the possible split values
  
  - Can be modified for categorical attributes
**Gini Index (IBM IntelligentMiner)**

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

  
  \[
  gini(T) = 1 - \sum_{j=1}^{n} p_j^2
  \]

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

- If a data set $T$ is split into two subsets $T_1$ and $T_2$ with sizes $N_1$ and $N_2$ respectively, the gini index of the split data contains examples from $n$ classes, the gini index $gini(T)$ is defined as

  \[
  gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)
  \]

- The attribute provides the smallest $gini_{split}(T)$ is chosen to split the node *(need to enumerate all possible splitting points for each attribute)*.
Extracting Classification Rules from Trees

- Represent the knowledge in the form of **IF-THEN** rules
- One rule is created for each path from the root to a leaf
- Each attribute-value pair along a path forms a conjunction
- The leaf node holds the class prediction
- Rules are easier for humans to understand
- Example

  IF \( \text{age} = "<=30" \) AND \( \text{student} = "no" \) THEN \( \text{buys\_computer} = "no" \)
  IF \( \text{age} = "<=30" \) AND \( \text{student} = "yes" \) THEN \( \text{buys\_computer} = "yes" \)
  IF \( \text{age} = "31...40" \) THEN \( \text{buys\_computer} = "yes" \)
  IF \( \text{age} = ">40" \) AND \( \text{credit\_rating} = "excellent" \) THEN \( \text{buys\_computer} = "yes" \)
  IF \( \text{age} = ">40" \) AND \( \text{credit\_rating} = "fair" \) THEN \( \text{buys\_computer} = "no" \)
Why Pruning?

- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result is in poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning:
    - Terminate tree construction early
      - do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning:
    - Remove branches from a “fully grown” tree
    - Use a set of data different from the training data to decide which is the “best pruned tree”
    - Usually, postpruning is more accurate than prepruning, yet requires more computation cost
Approaches to Determine the Final Tree Size

- Separate training (2/3) and testing (1/3) sets
- Use all the data for training
  - but apply a statistical test (e.g., chi-square) to estimate whether expanding or pruning a node may improve the entire distribution
- *Use minimum description length (MDL) principle:*
  - terminate growth of the tree when the encoding is minimized
Enhancements to basic decision tree induction

- Allow for continuous-valued attributes
  - Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals
- Handle missing attribute values
  - Assign the most common value of the attribute
  - Assign probability to each of the possible values
- Incremental adjustment
  - Build not from the scratch, but dynamically adjust from the previous trees
- Attribute construction
  - Create new attributes based on existing ones that are sparsely represented
  - This reduces fragmentation, repetition, and replication
Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why decision tree induction in data mining?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods
Scalable Decision Tree Induction Methods in Data Mining Studies

- **SLIQ** (EDBT’96 — Mehta et al.)
  - builds an index for each attribute and only class list and the current attribute list reside in memory

- **SPRINT** (VLDB’96 — J. Shafer et al.)
  - constructs an attribute list data structure

- **PUBLIC** (VLDB’98 — Rastogi & Shim)
  - integrates tree splitting and tree pruning: stop growing the tree earlier

- **RainForest** (VLDB’98 — Gehrke, Ramakrishnan & Ganti)
  - separates the scalability aspects from the criteria that determine the quality of the tree
  - builds an AVC-list (attribute, value, class label)
Data Cube-Based Decision-Tree Induction

- Integration of generalization with decision-tree induction (Kamber et al’97).
- Classification at primitive concept levels
  - E.g., precise temperature, humidity, outlook, etc.
  - Low-level concepts, scattered classes, bushy classification-trees
  - Semantic interpretation problems.
- Cube-based multi-level classification
  - Relevance analysis at multi-levels.
  - Information-gain analysis with dimension + level.
Classification

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by back propagation
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary
Bayesian Classification: Why?

- **Probabilistic learning**: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems.

- **Incremental**: Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.

- **Probabilistic prediction**: Predict multiple hypotheses, weighted by their probabilities.

- **Standard**: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured.
Bayesian Theorem

- Given training data $D$, posteriori probability of a hypothesis $h$, $P(h|D)$ follows the Bayes theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- MAP (maximum posteriori) hypothesis

$$h_{MAP} = \arg\max_{h\in H} P(h|D) = \arg\max_{h\in H} P(D|h)P(h).$$

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost
Bayesian classification

- The classification problem may be formalized using a-posteriori probabilities:
  \[ P(C|X) = \text{prob. that the sample tuple } X = \langle x_1, \ldots, x_k \rangle \text{ is of class } C. \]

- E.g. \( P(\text{class}=N \mid \text{outlook}=\text{sunny}, \text{windy}=\text{true}, \ldots) \)

- Idea: assign to sample \( X \) the class label \( C \) such that \( P(C|X) \) is maximal
Estimating a-posteriori probabilities

- Bayes theorem:
  \[ P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \]
- \( P(X) \) is constant for all classes
- \( P(C) = \) relative freq of class C samples
- C such that \( P(C|X) \) is maximum = C such that \( P(X|C) \cdot P(C) \) is maximum
- Problem: computing \( P(X|C) \) is unfeasible!
Naïve Bayesian Classification

- Naïve assumption: attribute independence
  \[ P(x_1, \ldots, x_k | C) = P(x_1 | C) \cdots P(x_k | C) \]

- If i-th attribute is categorical:
  \( P(x_i | C) \) is estimated as the relative freq of samples having value \( x_i \) as i-th attribute in class \( C \)

- If i-th attribute is continuous:
  \( P(x_i | C) \) is estimated thru a Gaussian density function

- Computationally easy in both cases
Play-tennis example: estimating $\Pr(x_i | C)$

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>N</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>N</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>N</td>
</tr>
<tr>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>mild</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>P</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>false</td>
<td>P</td>
</tr>
<tr>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>N</td>
</tr>
</tbody>
</table>

$\Pr(p) = 9/14$

$\Pr(n) = 5/14$

| Outlook | $\Pr(\text{sunny} | p) = 2/9$ | $\Pr(\text{sunny} | n) = 3/5$ |
|---------|-------------|-------------|
| temperature | $\Pr(\text{overcast} | p) = 4/9$ | $\Pr(\text{overcast} | n) = 0$ |
| $\Pr(\text{rain} | p) = 3/9$ | $\Pr(\text{rain} | n) = 2/5$ |

| humidity | $\Pr(\text{hot} | p) = 2/9$ | $\Pr(\text{hot} | n) = 2/5$ |
|----------|-------------|-------------|
| $\Pr(\text{mild} | p) = 4/9$ | $\Pr(\text{mild} | n) = 2/5$ |
| $\Pr(\text{cool} | p) = 3/9$ | $\Pr(\text{cool} | n) = 1/5$ |

| windy | $\Pr(\text{true} | p) = 3/9$ | $\Pr(\text{true} | n) = 3/5$ |
|--------|-------------|-------------|
| $\Pr(\text{false} | p) = 6/9$ | $\Pr(\text{false} | n) = 2/5$ |
Play-tennis example: classifying X

An unseen sample X = <rain, hot, high, false>

\[
P(X|p) \cdot P(p) = P(\text{rain}|p) \cdot P(\text{hot}|p) \cdot P(\text{high}|p) \cdot P(\text{false}|p) \cdot P(p) = \frac{3}{9} \cdot \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{6}{9} \cdot \frac{9}{14} = 0.010582
\]

\[
P(X|n) \cdot P(n) = P(\text{rain}|n) \cdot P(\text{hot}|n) \cdot P(\text{high}|n) \cdot P(\text{false}|n) \cdot P(n) = \frac{2}{5} \cdot \frac{2}{5} \cdot \frac{4}{5} \cdot \frac{2}{5} \cdot \frac{5}{14} = 0.018286
\]

Sample X is classified in class n (don’t play)
The independence hypothesis...

- makes computation possible
- yields optimal classifiers when attribute independence satisfied
- ... but is seldom satisfied in practice, as attributes (variables) are often correlated.
- Attempts to overcome this limitation:
  - Bayesian networks, that combine Bayesian reasoning with causal relationships between attributes
  - Decision trees, that reason on one attribute at the time, considering most important attributes first
Bayesian Belief Networks (I)

Directed acyclic graph:
node: random variable
Arc: probabilistic dependencies
Conditional probability table for each variable

Bayesian Belief Networks

The conditional probability table for the variable LungCancer

<table>
<thead>
<tr>
<th></th>
<th>(FH, S)</th>
<th>(FH, ~S)</th>
<th>(~FH, S)</th>
<th>(~FH, ~S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC</td>
<td>0.8</td>
<td>0.5</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>~LC</td>
<td>0.2</td>
<td>0.5</td>
<td>0.3</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Bayesian Belief Networks (II)

- Bayesian belief network allows a *subset* of the variables conditionally independent
- A graphical model of causal relationships
- Several cases of learning Bayesian belief networks
  - Known both network structure and all the variables: easy
  - Known network structure but only some variables
  - When the network structure is not known in advance
References (I)


References (II)


