Classification and Prediction

— Slides for Data Mining: Concepts and Techniques —
— Chapter 7 —

©Jiawei Han and Micheline Kamber
Intelligent Database Systems Research Lab
School of Computing Science
Simon Fraser University, Canada
http://www.cs.sfu.ca
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 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
Classification Process (1): Model Construction

IF rank = ‘professor’ OR years > 6
THEN tenured = ‘yes’
Classification Process (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)

Tenured? Yes
Supervised vs. Unsupervised Learning

- Supervised learning (e.g. 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
Evaluating Classification Methods

- Predictive accuracy
- Speed and scalability
  - time to construct the model
  - time to use the model
- Robustness
  - handling noise and missing values
- Scalability
  - efficiency in disk-resident databases
- Interpretability:
  - understanding and insight provided by the model
- Goodness of rules
  - 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
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
        - no
      - fair
        - yes
Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a \textit{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 (e.g., \textit{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 – \textit{majority voting} is employed for classifying the leaf
  - There are no samples left
Attribute Selection Measure

- **Information gain (ID3/C4.5)**
  - All attributes are assumed to be categorical
  - Can be modified for continuous-valued attributes

- **Gini index (IBM IntelligentMiner)**
  - 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**). 

Han: KDD --- Classification
Avoid Overfitting in Classification

- 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: Halt 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—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the “best pruned tree”
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)