Data Mining
Andreas Kusiak
Intelligent Systems Laboratory
2139 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 - 1527
Tel: 319 - 335 5934 Fax: 319 - 335 5669
andreas-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Partially based on the material provided by J Han and M Kamber

Classification and Prediction
- Learning
  - Classification and prediction
  - Classification by decision tree induction
  - Classification by backpropagation
  - Other Classification Methods
  - Prediction
  - Classification accuracy

http://www.kdnuggets.com/

Learning
What is learning?
- Extraction of knowledge
- Pattern creation

- Basis of learning
  - Training data set

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 for classifying cases with unknown outcomes

- Prediction:
  - models continuous-valued functions, i.e., predicts unknown or missing values

http://www.twocrows.com/glossary.htm

Classification Process: Model Construction

Learning Algorithm

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

**Example 1**

- **Classifier (Model)**
- **Testing Data**
- e.g., decision rules
- **Unseen Data**
- **NAME** | **RANK** | **YEARS** | **TENURED**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom</td>
<td>Assistant Prof</td>
<td>2</td>
<td>no</td>
</tr>
<tr>
<td>Melissa</td>
<td>Associate Prof</td>
<td>7</td>
<td>no</td>
</tr>
<tr>
<td>George</td>
<td>Professor</td>
<td>6</td>
<td>yes</td>
</tr>
<tr>
<td>Joseph</td>
<td>Assistant Prof</td>
<td>7</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Example 2**

- **Classification Process:** Use the Model in Prediction
- **Testing Data**
- **Unseen Data**
- **NAME** | **RANK** | **YEARS** | **TENURED**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom</td>
<td>Assistant Prof</td>
<td>2</td>
<td>no</td>
</tr>
<tr>
<td>Melissa</td>
<td>Associate Prof</td>
<td>7</td>
<td>no</td>
</tr>
<tr>
<td>George</td>
<td>Professor</td>
<td>6</td>
<td>yes</td>
</tr>
<tr>
<td>Joseph</td>
<td>Assistant Prof</td>
<td>7</td>
<td>yes</td>
</tr>
</tbody>
</table>

Training Data Set

- **No.** | **F1** | **F2** | **F3** | **F4** | **D**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.02 Red</td>
<td>2.98 High</td>
<td>Good</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2.03 Black</td>
<td>1.04 Low</td>
<td>Bad</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0.99 Blue</td>
<td>3.04 High</td>
<td>Good</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2.03 Blue</td>
<td>3.11 High</td>
<td>Good</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>0.03 Orange</td>
<td>0.96 Low</td>
<td>Bad</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>0.04 Blue</td>
<td>1.04 Medium</td>
<td>Bad</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>0.99 Orange</td>
<td>1.04 Medium</td>
<td>Good</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1.02 Red</td>
<td>0.94 Low</td>
<td>Bad</td>
<td>4</td>
</tr>
</tbody>
</table>

**Example 3**

- **Training Data Set**
- **No.** | **F1** | **F2** | **F3** | **F4** | **D**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.02 Red</td>
<td>2.98 High</td>
<td>Good</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2.03 Black</td>
<td>1.04 Low</td>
<td>Bad</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0.99 Blue</td>
<td>3.04 High</td>
<td>Good</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2.03 Blue</td>
<td>3.11 High</td>
<td>Good</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>0.03 Orange</td>
<td>0.96 Low</td>
<td>Bad</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>0.04 Blue</td>
<td>1.04 Medium</td>
<td>Bad</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>0.99 Orange</td>
<td>1.04 Medium</td>
<td>Good</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1.02 Red</td>
<td>0.94 Low</td>
<td>Bad</td>
<td>4</td>
</tr>
</tbody>
</table>

**Extracted Rules**

- **Rule 1.** IF (F4 = High) THEN (D = Good); [1, 3, 4]
- **Rule 2.** IF (F4 = Medium) AND (F2 = Blue) THEN (D = Bad); [6]
- **Rule 3.** IF (F4 = Medium) AND (F2 = Orange) THEN (D = Good); [7]
- **Rule 4.** IF (F4 = Low) THEN (D = Bad); [2, 5, 8]

**Patterns**

- **No.** | **F1** | **F2** | **F3** | **F4** | **D** | **Rule**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.02 Red</td>
<td>2.98 High</td>
<td>Good</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.03 Black</td>
<td>1.04 Low</td>
<td>Bad</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.99 Blue</td>
<td>3.04 High</td>
<td>Good</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.03 Blue</td>
<td>3.11 High</td>
<td>Good</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.03 Orange</td>
<td>0.96 Low</td>
<td>Bad</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.04 Blue</td>
<td>1.04 Medium</td>
<td>Bad</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.99 Orange</td>
<td>1.04 Medium</td>
<td>Good</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.02 Red</td>
<td>0.94 Low</td>
<td>Bad</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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
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

Decision Tree Algorithm

Quinlan’s data set

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit rating</th>
<th>buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>30...40</td>
<td>high</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></td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31...40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

Information Gain (C4.5)

Output: A Decision Tree for “buys_computer”

<table>
<thead>
<tr>
<th>age</th>
<th>student</th>
<th>credit rating</th>
<th>buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>30...40</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

Information Gain in Decision Tree Induction

- Assume that using attribute A a set S will be partitioned into sets \( S_1, S_2, \ldots, S_v \), i.e., v attribute A values
  - If \( S_i \) contains \( p_i \) examples of \( P \) and \( n_i \) examples of \( N \), the entropy, or the expected information needed to classify objects in all subsets \( S_i \) is
    \[
    E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)
    \]
- The encoding information that would be gained by branching on \( A \)
  \[
  \text{Gain}(A) = I(p, n) - E(A)
  \]

Example

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit rating</th>
<th>buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>30...40</td>
<td>high</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></td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31...40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

Continuous attribute values \( \rightarrow \) a split value
Attribute Selection by Information Gain Computation

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

\[
\text{Gain (age)} = I - E = 0.246
\]

Gain(income) = 0.029
Gain(student) = 0.151
Gain(credit_rating) = 0.048

Decision Tree Revisited

<table>
<thead>
<tr>
<th>age</th>
<th>student</th>
<th>credit rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>no</td>
<td>fair</td>
</tr>
<tr>
<td>30...40</td>
<td>yes</td>
<td>excellent</td>
</tr>
<tr>
<td>&gt;40</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

Neural Networks

- Advantages
  - prediction accuracy is generally high
  - robust, works when training examples contain errors
  - output may be discrete, real-valued, or a vector of several discrete or real-valued attributes
  - fast evaluation of the learned target function

- Disadvantages
  - long training time
  - difficult to understand the learned function (weights)
  - not easy to incorporate domain knowledge


Neural Network Training

- The ultimate objective of training
  - obtain a set of weights that makes almost all the tuples in the training data classified correctly

- Steps
  - Initialize weights with random values
  - Feed the input tuples into the network one by one
  - For each unit
    - Compute the net input to the unit as a linear combination of all the inputs to the unit
    - Compute the output value using the activation function
    - Compute the error
    - Update the weights and the bias

Multi-Layer Perceptron

\[
E_{\text{err}} = O (1 - O) \sum_i E_{\text{error}_i} w_i \\
E_{\text{err}_i} = \theta + (1) E_{\text{error}_i} O_i \\
w_i = w_i + (1) E_{\text{error}_i} O_i \\
O_i = \frac{1}{1 + e^{-1}} \\
O_i = \sum_i w_i O_i + \theta
\]
Other Classification Methods

- k-nearest neighbor classifier
- Case-based reasoning
- Genetic algorithm
- Rough set approach
- Fuzzy set approach

Genetic Algorithms

- GA: based on an analogy to biological evolution
- Each rule is represented by a string of bits
- An initial population is created consisting of randomly generated rules
  - e.g., IF A₁ and Not A₂ then C₂ can be encoded as 100
- Based on the notion of survival of the fittest, a new population is formed to consists of the fittest rules and their offspring
- The fitness of a rule is represented by its classification accuracy on a set of training examples
- Offspring are generated by crossover and mutation


Rough Set Approach

- Rough sets are used to approximately or “roughly” define equivalent classes
- A rough set for a given class C is approximated by two sets: a lower approximation (certain to be in C) and an upper approximation (cannot be described as not belonging to C)
- Finding the minimal subsets (reducts) of attributes (for feature reduction) is NP-hard but a discernibility matrix is used to reduce the computation intensity

Regression Analysis and Log-Linear Models in Prediction

- **Linear regression**: \( Y = \alpha + \beta X \)
  - Two parameters, \( \alpha \) and \( \beta \) specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of \( Y₁, Y₂, \ldots, X₁, X₂, \ldots \)
- **Multiple regression**: \( Y = b₀ + b₁ X₁ + b₂ X₂ \).
  - Many nonlinear functions can be transformed into the above.
- **Log-linear models**:
  - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
  - Probability: \( p(a, b, c, d) = \alpha \beta \chi \delta ab \chi bcd \)

Prediction: Numerical Data

Prediction: Categorical Data

http://www.research.att.com/areas/stat/xgobi/
Classification Accuracy 1

Classification accuracy (CA) of a rule set is the ratio of the number of correctly classified objects from the test set and all objects in the test set.

Classification Accuracy 2

\[
\text{Accuracy} = \frac{A + D}{A + B + C + D}
\]

Classification Accuracy 3

\[
\begin{array}{c|c|c|c}
\text{Predicted result} & + & - \\
\hline
A & + & + \\
B & - & - \\
C & + & - \\
D & - & + \\
\end{array}
\]

- Sensitivity (true positive rate) = \( \frac{A}{A+B} \)
- Specificity (true negative rate) = \( \frac{D}{C+D} \)
- False negative rate = \( \frac{B}{A+B} = 1 - \text{Sensitivity} \) (Type I error)
- False positive rate = \( \frac{C}{C+D} = 1 - \text{Specificity} \) (Type II error)
- Positive predicted value = \( \frac{A}{A+C} \)
- Negative predicted value = \( \frac{D}{B+D} \)

Classification Accuracy: Estimating Error Rates

- Partition: Training-and-testing
  - use two independent data sets, e.g., training set (2/3), test set (1/3)
  - used for data set with large number of samples
- Cross-validation
  - divide the data set into \( k \) subsamples
  - use \( k-1 \) subsamples as training data and one sub-sample as test data --- \( k \)-fold cross-validation
  - for data set with moderate size
- Bootstrapping (leave-one-out)
  - for small size data

Reference