MSCIT 5210: Knowledge Discovery and Data Mining

Acknowledgement: Slides modified by Dr. Lei Chen based on the slides provided by Jiawei Han, Micheline Kamber, Jian Pei and Raymond Wong

©2012 Han, Kamber & Pei &Raymond. All rights reserved.
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- K Nearest Neighbor Classification Methods
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary
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—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 overfitting)
    - If the accuracy is acceptable, use the model to **classify new data**
  - Note: If *the test set* is used to select models, it is called **validation (test) set**
Process (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’
Process (2): Using 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
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- K Nearest Neighbor Classification Methods
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary
The K-Nearest Neighbor Method

- Used for prediction/classification
- Given input $x$, (e.g., <sunny, normal, ..?>)
- #neighbors = $K$ (e.g., $k=3$)
  - Often a parameter to be determined
    - The form of the distance function
  - $K$ neighbors in training data to the input data $x$:
    - Break ties arbitrarily
- All $k$ neighbors will vote: majority wins
**How to decide the distance?**

Try 3-NN on this data: assume distance function = ' # of different attributes.'

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>FALSE</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>TRUE</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>high</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>TRUE</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>TRUE</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>FALSE</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>TRUE</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>TRUE</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>FALSE</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>high</td>
<td>TRUE</td>
<td>?</td>
</tr>
</tbody>
</table>
Nearest Neighbor Classifier

<table>
<thead>
<tr>
<th>Computer</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>40</td>
</tr>
<tr>
<td>90</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>95</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Diagram showing a scatter plot with axes labeled 'Computer' and 'History'.
# Nearest Neighbor Classifier

<table>
<thead>
<tr>
<th>Computer</th>
<th>History</th>
<th>Buy Book?</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>40</td>
<td>No (-)</td>
</tr>
<tr>
<td>90</td>
<td>45</td>
<td>Yes (+)</td>
</tr>
<tr>
<td>20</td>
<td>95</td>
<td>Yes (+)</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

![Scatter plot with data points marked as + and -](image-url)
Nearest Neighbor Classifier:

Step 1: Find the nearest neighbor

Step 2: Use the “label” of this neighbor

<table>
<thead>
<tr>
<th>Computer</th>
<th>History</th>
<th>Buy Book?</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>40</td>
<td>No (-)</td>
</tr>
<tr>
<td>90</td>
<td>45</td>
<td>Yes (+)</td>
</tr>
<tr>
<td>20</td>
<td>95</td>
<td>Yes (+)</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Suppose there is a new person

<table>
<thead>
<tr>
<th>Computer</th>
<th>History</th>
<th>Buy Book?</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>35</td>
<td>?</td>
</tr>
</tbody>
</table>
Nearest Neighbor Classifier:

**Step 1:** Find $k$ nearest neighbors  
**Step 2:** Use the majority of the labels of the neighbors

<table>
<thead>
<tr>
<th>Computer</th>
<th>History</th>
<th>Buy Book?</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>40</td>
<td>No (-)</td>
</tr>
<tr>
<td>90</td>
<td>45</td>
<td>Yes (+)</td>
</tr>
<tr>
<td>20</td>
<td>95</td>
<td>Yes (+)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Suppose there is a new person

<table>
<thead>
<tr>
<th>Computer</th>
<th>History</th>
<th>Buy Book?</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>35</td>
<td>?</td>
</tr>
</tbody>
</table>
Why important?

- Often a baseline
  - Must beat this one to claim innovation

- Forms of KNN
  - Weighted KNN
  - "K" is a variable:
    - Often we experiment with different values of K=1, 3, 5, to find out the optimal one

- Document similarity
  - Cosine

- Case based reasoning
  - Edited data base
  - Sometimes, the accuracy (CBR)/accuracy (KNN) can be better than 100%: why?

- Image understanding
  - Manifold learning
  - Distance metric
**K-NN can be misleading**

- Consider applying K-NN on the training data
  - What is the accuracy? 100%
  - Why? Distance to self is zero
  - What should we do in testing?
    - Use new data for testing, rather than training data.
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- K Nearest Neighbor Classification Methods
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary
Decision Tree Induction: An Example

- Training data set: Buys_computer
- The data set follows an example of Quinlan’s ID3 (Playing Tennis)
- Resulting tree:

```
<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>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31...40</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...40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=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>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31...40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31...40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>
```

```
Training data set: Buys_computer
The data set follows an example of Quinlan’s ID3 (Playing Tennis)
Resulting tree:
```
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 (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
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 $D$:
  \[
  Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)
  \]
- **Information gained** by branching on attribute $A$
  \[
  Gain(A) = Info(D) - Info_A(D)
  \]
Computing Information-Gain for Continuous-Valued 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 \(A \leq\) split-point, and D2 is the set of tuples in D satisfying \(A >\) split-point
Entropy

- **Entropy** is used to measure how informative is a node.
- If we are given a probability distribution
  \( P = (p_1, p_2, \ldots, p_n) \) then the **Information** conveyed by this distribution, also called the **Entropy** of \( P \), is:
  \[
  I(P) = - (p_1 \times \log p_1 + p_2 \times \log p_2 + \ldots + p_n \times \log p_n)
  \]
- All logarithms here are in base 2.
Entropy

- For example,
  - If $P$ is $(0.5, 0.5)$, then $I(P)$ is 1.
  - If $P$ is $(0.67, 0.33)$, then $I(P)$ is 0.92,
  - If $P$ is $(1, 0)$, then $I(P)$ is 0.

- The entropy is a way to measure the amount of information.

- The smaller the entropy, the more informative we have.
Entropy:

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>high</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

Info(T) = \(- \frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2}\) = 1

For attribute Race,

Info(T_{black}) = - \frac{3}{4} \log \frac{3}{4} - \frac{1}{4} \log \frac{1}{4} = 0.8113

Info(T_{white}) = - \frac{3}{4} \log \frac{3}{4} - \frac{1}{4} \log \frac{1}{4} = 0.8113

Info(Race, T) = \frac{1}{2} \times Info(T_{black}) + \frac{1}{2} \times Info(T_{white}) = 0.8113

Gain(Race, T) = Info(T) – Info(Race, T) = 1 – 0.8113 = 0.1887

For attribute Race, Gain(Race, T) = 0.1887
<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>high</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

**Entropy**

\[
\text{Info}(T) = - \frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1
\]

For attribute Income,

\[
\text{Info}(T_{\text{high}}) = - 1 \log 1 - 0 \log 0 = 0
\]

\[
\text{Info}(T_{\text{low}}) = - \frac{1}{3} \log \frac{1}{3} - \frac{2}{3} \log \frac{2}{3} = 0.9183
\]

\[
\text{Info}(\text{Income}, T) = \frac{1}{4} \times \text{Info}(T_{\text{high}}) + \frac{3}{4} \times \text{Info}(T_{\text{low}}) = 0.6887
\]

\[
\text{Gain}(\text{Income}, T) = \text{Info}(T) - \text{Info}(\text{Income}, T) = 1 - 0.6887 = 0.3113
\]

For attribute Race,

\[
\text{Gain}(\text{Race}, T) = 0.1887
\]

For attribute Income,

\[
\text{Gain}(\text{Income}, T) = 0.3113
\]
Info(T) = \( -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} \)  
\[= 1 \]

For attribute Child,

\[\text{Info(T}_{\text{yes}}\text{)} = -1 \log 1 - 0 \log 0 \quad = 0\]

\[\text{Info(T}_{\text{no}}\text{)} = -\frac{1}{5} \log \frac{1}{5} - \frac{4}{5} \log \frac{4}{5} \quad = 0.7219\]

\[\text{Info(Child, T)} = 3/8 \times \text{Info(T}_{\text{yes}}\text{)} + 5/8 \times \text{Info(T}_{\text{no}}\text{)} \quad = 0.4512\]

\[\text{Gain(Child, T)} = \text{Info(T)} - \text{Info(Child, T)} \quad = 1 - 0.4512 \quad = 0.5488\]

For attribute Race,

\[\text{Gain(Race, T)} = 0.1887\]

For attribute Income,

\[\text{Gain(Income, T)} = 0.3113\]

For attribute Child,

\[\text{Gain(Child, T)} = 0.5488\]
Info(T) = \(- \frac{1}{5} \log \frac{1}{5} - \frac{4}{5} \log \frac{4}{5}\)  
= 0.7219

For attribute Race,

\[ \text{Info}(T_{\text{white}}) = -0 \log 0 - 1 \log 1 = 0 \]

\[ \text{Info}(T_{\text{black}}) = -\frac{1}{4} \log \frac{1}{4} - \frac{3}{4} \log \frac{3}{4} = 0.8113 \]

\[ \text{Info}(\text{Race}, T) = \frac{4}{5} \times \text{Info}(T_{\text{black}}) + \frac{1}{5} \times \text{Info}(T_{\text{white}}) = 0.6490 \]

Gain(Race, T) = Info(T) – Info(Race, T) = 0.7219 – 0.6490 = 0.0729

For attribute Race,  
Gain(Race, T) = 0.0729
For attribute Income,

\[
\text{Info}(T_{\text{high}}) = -1 \log 1 - 0 \log 0 = 0
\]

\[
\text{Info}(T_{\text{low}}) = -0 \log 0 - 1 \log 1 = 0
\]

\[
\text{Info}(\text{Income, } T) = \frac{1}{5} \times \text{Info}(T_{\text{high}}) + \frac{4}{5} \times \text{Info}(T_{\text{low}}) = 0
\]

\[
\text{Gain}(\text{Income, } T) = \text{Info}(T) - \text{Info}(\text{Income, } T) = 0.7219 - 0 = 0.7219
\]

For attribute Race,

\[
\text{Gain}(\text{Race, } T) = 0.0729
\]
Gain(Income, T) = 0.7219

Gain(Race, T) = 0.0729

Gain(Income, T) = 0.7219
Suppose there is a new person.

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>high</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>
### Decision Tree

**Termination Criteria?**

- e.g., height of the tree
- e.g., accuracy of each node

<table>
<thead>
<tr>
<th>No.</th>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>black</td>
<td>high</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>white</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>white</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
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).

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

- \text{GainRatio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}(A)}

- Ex. 

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

- \text{gain\_ratio(income)} = 0.029/1.557 = 0.019

- The attribute with the maximum gain ratio is selected as the splitting attribute.
ID3
- Impurity Measurement
  - Gain(A, T)
    - $\text{Gain}(A, T) = \text{Info}(T) - \text{Info}(A, T)$

C4.5
- Impurity Measurement
  - Gain(A, T)
    - $\text{Gain}(A, T) = (\text{Info}(T) - \text{Info}(A, T))/\text{SplitInfo}(A)$
  - where $\text{SplitInfo}(A) = -\sum_{v \in A} p(v) \log p(v)$
Entropy

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>high</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>black</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>white</td>
<td>low</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

\[\text{Info}(T) = - \frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1\]

For attribute Race,
\[\text{Info}(T_{\text{black}}) = - \frac{3}{4} \log \frac{3}{4} - \frac{1}{4} \log \frac{1}{4} = 0.8113\]
\[\text{Info}(T_{\text{white}}) = - \frac{3}{4} \log \frac{3}{4} - \frac{1}{4} \log \frac{1}{4} = 0.8113\]
\[\text{Info}(\text{Race}, T) = \frac{1}{2} \times \text{Info}(T_{\text{black}}) + \frac{1}{2} \times \text{Info}(T_{\text{white}}) = 0.8113\]
\[\text{SplitInfo}(\text{Race}) = - \frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1\]
\[\text{Gain}(\text{Race}, T) = (\text{Info}(T) - \text{Info}(\text{Race}, T))/\text{SplitInfo}(\text{Race}) = (1 - 0.8113)/1 = 0.1887\]

For attribute Race,
\[\text{Gain}(\text{Race}, T) = 0.1887\]
### Entropy

\[
\text{Info}(T) = - \frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1
\]

For attribute Income,
\[
\text{Info}(T_{\text{high}}) = - 1 \log 1 - 0 \log 0 = 0
\]
\[
\text{Info}(T_{\text{low}}) = - \frac{1}{3} \log \frac{1}{3} - 2/3 \log 2/3 = 0.9183
\]
\[
\text{Info}(\text{Income, } T) = \frac{1}{4} \times \text{Info}(T_{\text{high}}) + \frac{3}{4} \times \text{Info}(T_{\text{low}}) = 0.6887
\]
\[
\text{SplitInfo}(\text{Income}) = - \frac{2}{8} \log 2/8 - 6/8 \log 6/8 = 0.8113
\]
\[
\text{Gain}(\text{Income, } T) = (\text{Info}(T) - \text{Info}(\text{Income, } T))/\text{SplitInfo}(\text{Income}) = (1 - 0.6887)/0.8113 = 0.3837
\]

For attribute Race,
\[
\text{Gain}(\text{Race, } T) = 0.1887
\]

For attribute Income,
\[
\text{Gain}(\text{Income, } T) = 0.3837
\]

For attribute Child,
\[
\text{Gain}(\text{Child, } T) = ?
\]
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)
CART

- Impurity Measurement
  - Gini
    \[ I(P) = 1 - \sum_j p_j^2 \]
Info(T) = 1 – (½)^2 – (½)^2
   = ½

For attribute Race,

Info(T_{black}) = 1 – (¾)^2 – (¼)^2  = 0.375
Info(T_{white}) = 1 – (¾)^2 – (¼)^2  = 0.375

Info(Race, T) = ½ x Info(T_{black}) + ½ x Info(T_{white}) = 0.375

Gain(Race, T) = Info(T) – Info(Race, T) = ½ – 0.375  = 0.125

For attribute Race,  Gain(Race, T) = 0.125
Info(T) = 1 – (½)^2 – (½)^2
= ½

For attribute Income,

Info(T_{high}) = 1 – 1^2 – 0^2 = 0
Info(T_{low}) = 1 – (1/3)^2 – (2/3)^2 = 0.444

Info(Income, T) = 1/4 x Info(T_{high}) + 3/4 x Info(T_{low}) = 0.333

Gain(Income, T) = Info(T) – Info(Race, T) = ½ – 0.333 = 0.167

For attribute Race,
Gain(Race, T) = 0.125

For attribute Income,
Gain(Race, T) = 0.167

For attribute Child,
Gain(Child, T) = ?
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
Overfitting and Tree Pruning

- **Overfitting:** An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - 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”
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
- **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 is decision tree induction popular?
  - 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
- RainForest (VLDB’98 — Gehrke, Ramakrishnan & Ganti)
  - Builds an AVC-list (attribute, value, class label)
Scalability Framework for RainForest

- Separates the scalability aspects from the criteria that determine the quality of the tree
- Builds an AVC-list: AVC (Attribute, Value, Class_label)
- AVC-set (of an attribute $X$)
  - Projection of training dataset onto the attribute $X$ and class label where counts of individual class label are aggregated
- AVC-group (of a node $n$)
  - Set of AVC-sets of all predictor attributes at the node $n$
Rainforest: Training Set and Its AVC Sets

### Training Examples

<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>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31…40</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…40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=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>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31…40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31…40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

#### AVC-set on Age

<table>
<thead>
<tr>
<th>Age</th>
<th>Buy_Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>yes</td>
</tr>
<tr>
<td>31..40</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>income</th>
<th>Buy_Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>yes</td>
</tr>
<tr>
<td>low</td>
<td>yes</td>
</tr>
</tbody>
</table>

#### AVC-set on Student

<table>
<thead>
<tr>
<th>student</th>
<th>Buy_Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

#### AVC-set on Credit

<table>
<thead>
<tr>
<th>Credit rating</th>
<th>Buy_Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>excellent</td>
<td>yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>student</th>
<th>Buy_Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>6</td>
</tr>
<tr>
<td>no</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Credit rating</th>
<th>Buy_Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>6</td>
</tr>
<tr>
<td>fair</td>
<td>2</td>
</tr>
<tr>
<td>excellent</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Credit rating</th>
<th>Buy_Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>6</td>
</tr>
<tr>
<td>fair</td>
<td>2</td>
</tr>
<tr>
<td>excellent</td>
<td>3</td>
</tr>
</tbody>
</table>
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- K Nearest Neighbor Classification Methods
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary
Bayesian Classification: Why?

- **A statistical classifier**: performs *probabilistic prediction*, i.e., predicts class membership probabilities
- **Foundation**: Based on Bayes’ Theorem.
- **Performance**: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- **Incremental**: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- **Standard**: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured
Bayes’ Theorem: Basics

- Let $X$ be a data sample ("evidence"): class label is unknown
- Let $H$ be a hypothesis that $X$ belongs to class $C$
- Classification is to determine $P(H|X)$, (*posteriori probability*), the probability that the hypothesis holds given the observed data sample $X$
- $P(H)$ (*prior probability*), the initial probability
  - E.g., $X$ will buy computer, regardless of age, income, ...
- $P(X)$: probability that sample data is observed
- $P(X|H)$ (likelihood), the probability of observing the sample $X$, given that the hypothesis holds
  - E.g., Given that $X$ will buy computer, the prob. that $X$ is 31..40, medium income
Bayes’ Theorem

- Given training data $\mathbf{X}$, posteriori probability of a hypothesis $H$, $P(H|\mathbf{X})$, follows the Bayes’ theorem

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times \frac{P(H)}{P(\mathbf{X})}$$

- Informally, this can be written as

  posteriori = likelihood x prior/evidence

- Predicts $\mathbf{X}$ belongs to $C_2$ iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|\mathbf{X})$ for all the $k$ classes

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost
Towards Naïve Bayes Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector \( X = (x_1, x_2, ..., x_n) \)
- Suppose there are \( m \) classes \( C_1, C_2, ..., C_m \).
- Classification is to derive the maximum posteriori, i.e., the maximal \( P(C_i|X) \)
- This can be derived from Bayes’ theorem

\[
P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}
\]

- Since \( P(X) \) is constant for all classes, only

\[
P(C_i|X) = P(X|C_i)P(C_i)
\]
A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

\[
P(X | C_i) = \prod_{k=1}^{n} P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \ldots \times P(x_n | C_i)
\]

This greatly reduces the computation cost: Only counts the class distribution.

If \( A_k \) is categorical, \( P(x_k | C_i) \) is the # of tuples in \( C_i \) having value \( x_k \) for \( A_k \) divided by \( |C_i, D| \) (# of tuples of \( C_i \) in \( D \)).

If \( A_k \) is continuous-valued, \( P(x_k | C_i) \) is usually computed based on Gaussian distribution with a mean \( \mu \) and standard deviation \( \sigma \):

\[
\begin{align*}
g(x, \mu, \sigma) &= \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\
P(X | C_i) &= g(x_k, \mu_{C_i}, \sigma_{C_i})
\end{align*}
\]
Naïve Bayes Classifier

- Conditional Probability
  - A: a random variable
  - B: a random variable
  - \( P(A \mid B) = \frac{P(AB)}{P(B)} \)
Naïve Bayes Classifier

- **Bayes Rule**
  - A : a random variable
  - B: a random variable

\[
P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}
\]
Independent Assumption

- Each attribute are independent

- e.g.,
  \[ P(X, Y, Z \mid A) = P(X \mid A) \times P(Y \mid A) \times P(Z \mid A) \]
Suppose there is a new person.

For attribute Race,
\[ P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4} \]

For attribute Income,
\[ P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{high} \mid \text{No}) = 0 \]
\[ P(\text{Income} = \text{low} \mid \text{No}) = 1 \]

For attribute Child,
\[ P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Child} = \text{yes} \mid \text{No}) = 0 \]
\[ P(\text{Child} = \text{no} \mid \text{No}) = 1 \]

For attribute Race, $\text{Insurance} = \text{Yes}$

\[ P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4} \]

\[ P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{high} \mid \text{No}) = 0 \]
\[ P(\text{Income} = \text{low} \mid \text{No}) = 1 \]

\[ P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Child} = \text{yes} \mid \text{No}) = 0 \]
\[ P(\text{Child} = \text{no} \mid \text{No}) = 1 \]

Suppose there is a new person.

For attribute Race,
\[ P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4} \]

For attribute Income,
\[ P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{high} \mid \text{No}) = 0 \]
\[ P(\text{Income} = \text{low} \mid \text{No}) = 1 \]

For attribute Child,
\[ P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Child} = \text{yes} \mid \text{No}) = 0 \]
\[ P(\text{Child} = \text{no} \mid \text{No}) = 1 \]

Naïve Bayes Classifier

\[ P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Race} = \text{black}, \text{Income} = \text{low}, \text{Child} = \text{no} \mid \text{No}) = 0 \]

\[ P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{2} \times \frac{1}{2} \times \frac{3}{4} = \frac{3}{32} = 0.09375 \]

\[ P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{No}) = 0 \]

\[ P(\text{Race} = \text{white}, \text{Income} = \text{low}, \text{Child} = \text{no} \mid \text{Yes}) = \frac{3}{4} \times \frac{1}{2} \times \frac{1}{4} = 0.09375 \]

\[ P(\text{Race} = \text{white}, \text{Income} = \text{low}, \text{Child} = \text{no} \mid \text{No}) = 0 \]
Suppose there is a new person.

For attribute Race,

\[ P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4} \]
\[ P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4} \]

For attribute Income,

\[ P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2} \]
\[ P(\text{Income} = \text{high} \mid \text{No}) = 0 \]
\[ P(\text{Income} = \text{low} \mid \text{No}) = 1 \]

For attribute Child,

\[ P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4} \]
\[ P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4} \]
\[ P(\text{Child} = \text{yes} \mid \text{No}) = 0 \]
\[ P(\text{Child} = \text{no} \mid \text{No}) = 1 \]

Insurance = Yes

\[ P(\text{Yes}) = \frac{1}{2} \]
\[ P(\text{No}) = \frac{1}{2} \]

\[ P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{Yes}) \]
\[ = 0.09375 \]
\[ = P(\text{Race} = \text{white} \mid \text{No}) \times P(\text{Income} = \text{high} \mid \text{No}) \]
\[ \times P(\text{Child} = \text{no} \mid \text{No}) \]
\[ = \frac{1}{4} \times 0 \times 1 \]
\[ = 0 \]
Suppose there is a new person.

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>high</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>

For attribute Race, $P(Race = black | Yes) = \frac{1}{4}$
$P(Race = white | Yes) = \frac{3}{4}$
$P(Race = black | No) = \frac{3}{4}$
$P(Race = white | No) = \frac{1}{4}$

For attribute Income, $P(Income = high | Yes) = \frac{1}{2}$
$P(Income = low | Yes) = \frac{1}{2}$
$P(Income = high | No) = 0$
$P(Income = low | No) = 1$

For attribute Child,
$P(Child = yes | Yes) = \frac{3}{4}$
$P(Child = no | Yes) = \frac{1}{4}$
$P(Child = yes | No) = 0$
$P(Child = no | No) = 1$

Naïve Bayes Classifier

\[
P(Race = white, Income = high, Child = no | Yes) = 0.09375
\]

\[
P(Race = white, Income = high, Child = no | No) = P(Race = white | No) 	imes P(Income = high | No) 	imes P(Child = no | No)
\]

\[
= \frac{1}{4} \times 0 \times 1
\]

\[
= 0
\]
Suppose there is a new person. The Naïve Bayes Classifier assigns the class with the highest probability.

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>high</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>

For attribute Race,

- $P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4}$
- $P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4}$
- $P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4}$
- $P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4}$

For attribute Income,

- $P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2}$
- $P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2}$
- $P(\text{Income} = \text{high} \mid \text{No}) = 0$
- $P(\text{Income} = \text{low} \mid \text{No}) = 1$

For attribute Child,

- $P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4}$
- $P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4}$
- $P(\text{Child} = \text{yes} \mid \text{No}) = 0$
- $P(\text{Child} = \text{no} \mid \text{No}) = 1$

For attribute Insurance,

- $P(\text{Insurance} = \text{Yes}) = \frac{1}{2}$
- $P(\text{Insurance} = \text{No}) = \frac{1}{2}$

Naïve Bayes Classifier:

$$P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{Yes}) = 0.09375$$

$$P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{No}) = 0$$
Suppose there is a new person.

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>high</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>

For attribute Race,

\[
P(Race = \text{black} \mid \text{Yes}) = \frac{1}{4}
\]
\[
P(Race = \text{white} \mid \text{Yes}) = \frac{3}{4}
\]
\[
P(Race = \text{black} \mid \text{No}) = \frac{3}{4}
\]
\[
P(Race = \text{white} \mid \text{No}) = \frac{1}{4}
\]

For attribute Income,

\[
P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2}
\]
\[
P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2}
\]
\[
P(\text{Income} = \text{high} \mid \text{No}) = 0
\]
\[
P(\text{Income} = \text{low} \mid \text{No}) = 1
\]

For attribute Child,

\[
P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4}
\]
\[
P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4}
\]
\[
P(\text{Child} = \text{yes} \mid \text{No}) = 0
\]
\[
P(\text{Child} = \text{no} \mid \text{No}) = 1
\]

Naïve Bayes Classifier

\[
P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{Yes}) = 0.09375
\]
\[
P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{No}) = 0
\]
Suppose there is a new person.

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>high</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

For attribute Race,

- \( P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4} \)
- \( P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4} \)
- \( P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4} \)
- \( P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4} \)

For attribute Income,

- \( P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2} \)
- \( P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2} \)
- \( P(\text{Income} = \text{high} \mid \text{No}) = 0 \)
- \( P(\text{Income} = \text{low} \mid \text{No}) = 1 \)

For attribute Child,

- \( P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4} \)
- \( P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4} \)
- \( P(\text{Child} = \text{yes} \mid \text{No}) = 0 \)
- \( P(\text{Child} = \text{no} \mid \text{No}) = 1 \)

P(Yes | Race = white, Income = high, Child = no) = \( \frac{0.09375 x 0.5}{0.046875} = 0.09375 \)

\[
P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} | \text{Yes}) \]

\[
P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} | \text{No}) = 0
\]
Suppose there is a new person.

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>high</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>

For attribute Race,

- \( P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4} \)
- \( P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4} \)
- \( P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4} \)
- \( P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4} \)

For attribute Income,

- \( P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2} \)
- \( P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2} \)
- \( P(\text{Income} = \text{high} \mid \text{No}) = 0 \)
- \( P(\text{Income} = \text{low} \mid \text{No}) = 1 \)

For attribute Child,

- \( P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4} \)
- \( P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4} \)
- \( P(\text{Child} = \text{yes} \mid \text{No}) = 0 \)
- \( P(\text{Child} = \text{no} \mid \text{No}) = 1 \)

Naïve Bayes Classifier

- \( P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{Yes}) = 0.09375 \)
- \( P(\text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no} \mid \text{No}) = 0 \)

\[
P(\text{Yes} \mid \text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no}) = 0.046875
\]

\[
P(\text{No} \mid \text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no}) = 0
\]
Suppose there is a new person.

<table>
<thead>
<tr>
<th>Race</th>
<th>Income</th>
<th>Child</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>high</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>

### Naïve Bayes Classifier

**For attribute Race,**

- $P(\text{Race} = \text{black} \mid \text{Yes}) = \frac{1}{4}$
- $P(\text{Race} = \text{white} \mid \text{Yes}) = \frac{3}{4}$
- $P(\text{Race} = \text{black} \mid \text{No}) = \frac{3}{4}$
- $P(\text{Race} = \text{white} \mid \text{No}) = \frac{1}{4}$

**For attribute Income,**

- $P(\text{Income} = \text{high} \mid \text{Yes}) = \frac{1}{2}$
- $P(\text{Income} = \text{low} \mid \text{Yes}) = \frac{1}{2}$
- $P(\text{Income} = \text{high} \mid \text{No}) = 0$
- $P(\text{Income} = \text{low} \mid \text{No}) = 1$

**For attribute Child,**

- $P(\text{Child} = \text{yes} \mid \text{Yes}) = \frac{3}{4}$
- $P(\text{Child} = \text{no} \mid \text{Yes}) = \frac{1}{4}$
- $P(\text{Child} = \text{yes} \mid \text{No}) = 0$
- $P(\text{Child} = \text{no} \mid \text{No}) = 1$

Since $P(\text{Yes} \mid \text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no}) > P(\text{No} \mid \text{Race} = \text{white}, \text{Income} = \text{high}, \text{Child} = \text{no})$.

we predict the following new person will buy an insurance.
Naïve Bayes Classifier: Training Dataset

Class:
C1: buys_computer = ‘yes’
C2: buys_computer = ‘no’

Data to be classified:
X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

<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>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31…40</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…40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=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>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31…40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31…40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>
Naïve Bayes Classifier: An Example

- **P(C_i)**:  
  - P(buys_computer = “yes”) = 9/14 = 0.643
  - P(buys_computer = “no”) = 5/14 = 0.357

- Compute P(X | C_i) for each class
  - P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222
  - P(age = “<=30” | buys_computer = “no”) = 3/5 = 0.6
  - P(income = “medium” | buys_computer = “yes”) = 4/9 = 0.444
  - P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4
  - P(student = “yes” | buys_computer = “yes”) = 6/9 = 0.667
  - P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2
  - P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667
  - P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

- X = (age <= 30, income = medium, student = yes, credit_rating = fair)
  - P(X | C_i) : P(X | buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
  - P(X | buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019
  - P(X | C_i) * P(C_i) : P(X | buys_computer = “yes”) * P(buys_computer = “yes”) = 0.028
  - P(X | buys_computer = “no”) * P(buys_computer = “no”) = 0.007

Therefore, X belongs to class (“buys_computer = yes”)
Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be non-zero. Otherwise, the predicted prob. will be zero.

\[
P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)
\]

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income=medium (990), and income = high (10).

- Use **Laplacian correction** (or Laplacian estimator)
  - Adding 1 to each case
    - Prob(income = low) = 1/1003
    - Prob(income = medium) = 991/1003
    - Prob(income = high) = 11/1003
  - The “corrected” prob. estimates are close to their “uncorrected” counterparts.
Naïve Bayes Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases

- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc. Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier

- How to deal with these dependencies? Bayesian Belief Networks (Chapter 9)
Bayesian Belief Network

- Naïve Bayes Classifier
  - Independent Assumption
- Bayesian Belief Network
  - Do not have independent assumption
Bayesian Belief Networks

- **Bayesian belief network** (also known as Bayesian network, probabilistic network): allows *class conditional independencies* between *subsets* of variables

- Two components: (1) A *directed acyclic graph* (called a structure) and (2) a set of *conditional probability tables* (CPTs)

- A *(directed acyclic)* graphical model of *causal influence* relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution
    - Nodes: random variables
    - Links: dependency
    - X and Y are the parents of Z, and Y is the parent of P
    - No dependency between Z and P
    - Has no loops/cycles
A Bayesian Network and Some of Its CPTs

CPT: Conditional Probability Tables

| Fire | Smoke | \( \Theta_{s|f} \) |
|------|-------|-----------------|
| True | True  | .90             |
| False| True  | .01             |

| Fire | Tampering | Alarm | \( \Theta_{a|f,t} \) |
|------|-----------|-------|---------------------|
| True | True      | True  | .5                  |
| True | False     | True  | .99                 |
| False| True      | True  | .85                 |
| False| False     | True  | .0001               |

CPT shows the conditional probability for each possible combination of its parents

\[
P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i))
\]

Derivation of the probability of a particular combination of values of \( X \), from CPT:
Bayesian Belief Network

Exercise

Diet

Heartburn

Blood Pressure

Chest Pain

Heart Disease

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Diet</th>
<th>Heartburn</th>
<th>Blood Pressure</th>
<th>Chest Pain</th>
<th>Heart Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Healthy</td>
<td>No</td>
<td>High</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Unhealthy</td>
<td>Yes</td>
<td>Low</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Healthy</td>
<td>Yes</td>
<td>High</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Some attributes are dependent on other attributes.

E.g., doing exercises may reduce the probability of suffering from Heart Disease

Exercise (E) → Heart Disease
Bayesian Belief Network

**Exercise (E)**

<table>
<thead>
<tr>
<th></th>
<th>HD=Yes</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E=Yes</td>
<td>D=Healthy</td>
<td>0.25</td>
<td>E=Yes</td>
</tr>
<tr>
<td>D=Unhealthy</td>
<td>0.45</td>
<td>E=No</td>
<td></td>
</tr>
<tr>
<td>D=Healthy</td>
<td>0.55</td>
<td>E=No</td>
<td></td>
</tr>
<tr>
<td>D=Unhealthy</td>
<td>0.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Diet (D)**

<table>
<thead>
<tr>
<th></th>
<th>Hb=Yes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D=Healthy</td>
<td>0.85</td>
<td>D=Unhealthy</td>
</tr>
</tbody>
</table>

**Heart Disease (HD)**

- HD=Yes: 0.85
- HD=No: 0.2

**Heartburn (Hb)**

<table>
<thead>
<tr>
<th></th>
<th>CP=Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD=Yes</td>
<td>Hb=Yes</td>
</tr>
<tr>
<td>HD=Yes</td>
<td>Hb=No</td>
</tr>
<tr>
<td>HD=No</td>
<td>Hb=Yes</td>
</tr>
<tr>
<td>HD=No</td>
<td>Hb=No</td>
</tr>
</tbody>
</table>

**Blood Pressure (BP)**

<table>
<thead>
<tr>
<th></th>
<th>BP=High</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD=Yes</td>
<td>0.85</td>
</tr>
<tr>
<td>HD=No</td>
<td>0.2</td>
</tr>
</tbody>
</table>

**Chest Pain (CP)**
Let $X$, $Y$, $Z$ be three random variables. 

$X$ is said to be **conditionally independent** of $Y$ given $Z$ if the following holds.

$$P(X \mid Y, Z) = P(X \mid Z)$$

**Lemma:**

If $X$ is conditionally independent of $Y$ given $Z$,

$$P(X, Y \mid Z) = P(X \mid Z) \times P(Y \mid Z)$$
Let $X$, $Y$, $Z$ be three random variables. $X$ is said to be conditionally independent of $Y$ given $Z$ if the following holds:

$$P(X \mid Y, Z) = P(X \mid Z)$$

**Property:** A node is conditionally independent of its non-descendants if its parents are known.

- **e.g.,** $P(BP = \text{High} \mid HD = \text{Yes}, D = \text{Healthy}) = P(BP = \text{High} \mid HD = \text{Yes})$
  
  “$BP = \text{High}$” is conditionally independent of “$D = \text{Healthy}$” given “$HD = \text{Yes}$”

- **e.g.,** $P(BP = \text{High} \mid HD = \text{Yes}, CP=\text{Yes}) = P(BP = \text{High} \mid HD = \text{Yes})$
  
  “$BP = \text{High}$” is conditionally independent of “$CP = \text{Yes}$” given “$HD = \text{Yes}$”
Suppose there is a new person and I want to know whether he is likely to have Heart Disease.

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Diet</th>
<th>Heartburn</th>
<th>Blood Pressure</th>
<th>Chest Pain</th>
<th>Heart Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Healthy</td>
<td>No</td>
<td>High</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Unhealthy</td>
<td>Yes</td>
<td>Low</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>Healthy</td>
<td>Yes</td>
<td>High</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Exercise | Diet | Heartburn | Blood Pressure | Chest Pain | Heart Disease |
|---------|------|-----------|----------------|------------|---------------|

Exercise | Diet | Heartburn | Blood Pressure | Chest Pain | Heart Disease |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
<td>High</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Exercise | Diet | Heartburn | Blood Pressure | Chest Pain | Heart Disease |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Healthy</td>
<td>?</td>
<td>High</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Bayesian Belief Network

Suppose there is a new person and I want to know whether he is likely to have Heart Disease.

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Diet</th>
<th>Heartburn</th>
<th>Blood Pressure</th>
<th>Chest Pain</th>
<th>Heart Disease</th>
</tr>
</thead>
</table>

\[
P(HD = \text{Yes}) = \sum_{x \in \{\text{Yes, No}\}} \sum_{y \in \{\text{Healthy, Unhealthy}\}} P(HD=\text{Yes}|E=x, D=y) \times P(E=x, D=y)
\]

\[
= \sum_{x \in \{\text{Yes, No}\}} \sum_{y \in \{\text{Healthy, Unhealthy}\}} P(HD=\text{Yes}|E=x, D=y) \times P(E=x) \times P(D=y)
\]

\[
= 0.25 \times 0.7 \times 0.25 + 0.45 \times 0.7 \times 0.75 + 0.55 \times 0.3 \times 0.25
+ 0.75 \times 0.3 \times 0.75
\]

\[
= 0.49
\]

\[
P(HD = \text{No}) = 1 - P(HD = \text{Yes})
\]

\[
= 1 - 0.49
\]

\[
= 0.51
\]
Bayesian Belief Network

Suppose there is a new person and I want to know whether he is likely to have Heart Disease.

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Diet</th>
<th>Heartburn</th>
<th>Blood Pressure</th>
<th>Chest Pain</th>
<th>Heart Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
<td>High</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

\[
P(BP = \text{High}) = \sum_{x \in \{\text{Yes, No}\}} P(BP = \text{High}|HD=x) \times P(HD = x)
\]

\[
= 0.85 \times 0.49 + 0.2 \times 0.51
\]

\[
= 0.5185
\]

\[
P(\text{HD = Yes}\mid BP = \text{High}) = \frac{P(BP = \text{High}\mid HD=\text{Yes}) \times P(HD = \text{Yes})}{P(BP = \text{High})}
\]

\[
= \frac{0.85 \times 0.49}{0.5185}
\]

\[
= 0.8033
\]

\[
P(\text{HD = No}\mid BP = \text{High}) = 1 - P(\text{HD = Yes}\mid BP = \text{High})
\]

\[
= 1 - 0.8033
\]

\[
= 0.1967
\]
Bayesian Belief Network

Suppose there is a new person and I want to know whether he is likely to have Heart Disease.

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Diet</th>
<th>Heartburn</th>
<th>Blood Pressure</th>
<th>Chest Pain</th>
<th>Heart Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Healthy</td>
<td>?</td>
<td>High</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

\[
P(HD = \text{Yes} | BP = \text{High}, D = \text{Healthy}, E = \text{Yes}) = \frac{P(BP = \text{High} | HD = \text{Yes}, D = \text{Healthy}, E = \text{Yes}) \times P(HD = \text{Yes} | D = \text{Healthy}, E = \text{Yes})}{P(BP = \text{High} | D = \text{Healthy}, E = \text{Yes})} = \frac{P(BP = \text{High} | HD = \text{Yes}) P(HD = \text{Yes} | D = \text{Healthy}, E = \text{Yes})}{\sum_{x \in \{\text{Yes, No}\}} P(BP = \text{High} | HD = x) P(HD = x | D = \text{Healthy}, E = \text{Yes})} = \frac{0.85 \times 0.25}{0.85 \times 0.25 + 0.2 \times 0.75} = 0.5862
\]

\[
P(HD = \text{No} | BP = \text{High}, D = \text{Healthy}, E = \text{Yes}) = 1 - P(HD = \text{Yes} | BP = \text{High}, D = \text{Healthy}, E = \text{Yes}) = 1 - 0.5862 = 0.4138
\]
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- K Nearest Neighbor Classification Methods
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary
Model Evaluation and Selection

- Evaluation metrics: How can we measure accuracy? Other metrics to consider?
- Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier’s accuracy:
  - Holdout method, random subsampling
  - Cross-validation
  - Bootstrap
- Comparing classifiers:
  - Confidence intervals
  - Cost-benefit analysis and ROC Curves
Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

<table>
<thead>
<tr>
<th>Actual class\Predicted class</th>
<th>$C_1$</th>
<th>$\neg C_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>True Positives (TP)</td>
<td>False Negatives (FN)</td>
</tr>
<tr>
<td>$\neg C_1$</td>
<td>False Positives (FP)</td>
<td>True Negatives (TN)</td>
</tr>
</tbody>
</table>

Example of Confusion Matrix:

<table>
<thead>
<tr>
<th>Actual class\Predicted class</th>
<th>buy_computer = yes</th>
<th>buy_computer = no</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>buy_computer = yes</td>
<td>6954</td>
<td>46</td>
<td>7000</td>
</tr>
<tr>
<td>buy_computer = no</td>
<td>412</td>
<td>2588</td>
<td>3000</td>
</tr>
<tr>
<td>Total</td>
<td>7366</td>
<td>2634</td>
<td>10000</td>
</tr>
</tbody>
</table>

- Given $m$ classes, an entry, $CM_{i,j}$ in a confusion matrix indicates # of tuples in class $i$ that were labeled by the classifier as class $j$.
- May have extra rows/columns to provide totals.
Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

- **Classifier Accuracy**, or recognition rate: percentage of test set tuples that are correctly classified
  
  $\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{All}}$

- **Error rate**: $1 - \text{accuracy}$, or
  
  $\text{Error rate} = \frac{\text{FP} + \text{FN}}{\text{All}}$

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class

- **Sensitivity**: True Positive recognition rate
  
  $\text{Sensitivity} = \frac{\text{TP}}{\text{P}}$

- **Specificity**: True Negative recognition rate
  
  $\text{Specificity} = \frac{\text{TN}}{\text{N}}$

---

<table>
<thead>
<tr>
<th>A\P</th>
<th>C</th>
<th>¬C</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>¬C</td>
<td>FP</td>
<td>TN</td>
</tr>
<tr>
<td>P'</td>
<td>N'</td>
<td>All</td>
</tr>
</tbody>
</table>
Classifier Evaluation Metrics:
Precision and Recall, and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

- **Recall**: completeness – what % of positive tuples did the classifier label as positive?

- Perfect score is 1.0

- Inverse relationship between precision & recall

- **F measure** ($F_1$ or **F-score**): harmonic mean of precision and recall,

- $F_\beta$: weighted measure of precision and recall
  - assigns $\beta$ times as much weight to recall as to precision

\[
\text{precision} = \frac{TP}{TP + FP}
\]

\[
\text{recall} = \frac{TP}{TP + FN}
\]

\[
F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

\[
F_\beta = \frac{(1 + \beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}
\]
Classifier Evaluation Metrics: Example

- **Precision** = $90/230 = 39.13\%$
- **Recall** = $90/300 = 30.00\%$

<table>
<thead>
<tr>
<th>Actual Class \ Predicted class</th>
<th>cancer = yes</th>
<th>cancer = no</th>
<th>Total</th>
<th>Recognition(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer = yes</td>
<td>90</td>
<td>210</td>
<td>300</td>
<td>30.00 (sensitivity)</td>
</tr>
<tr>
<td>cancer = no</td>
<td>140</td>
<td>9560</td>
<td>9700</td>
<td>98.56 (specificity)</td>
</tr>
<tr>
<td>Total</td>
<td>230</td>
<td>9770</td>
<td>10000</td>
<td>96.40 (accuracy)</td>
</tr>
</tbody>
</table>
Evaluating Classifier Accuracy: Holdout & Cross-Validation Methods

- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained

- **Cross-validation** (k-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At i-th iteration, use D_i as test set and others as training set
  - Leave-one-out: k folds where k = # of tuples, for small sized data
  - **Stratified cross-validation**: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data
Evaluating Classifier Accuracy: Bootstrap

- **Bootstrap**
  - Works well with small data sets
  - Samples the given training tuples uniformly *with replacement*
    - i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
  - Several bootstrap methods, and a common one is **0.632 bootstrap**
    - A data set with \(d\) tuples is sampled \(d\) times, with replacement, resulting in a training set of \(d\) samples. The data tuples that did not make it into the training set end up forming the test set. About 63.2% of the original data end up in the bootstrap, and the remaining 36.8% form the test set (since \((1 - 1/d)^d \approx e^{-1} = 0.368\))
    - Repeat the sampling procedure \(k\) times, overall accuracy of the model:
      \[
      Acc(M) = \frac{1}{k} \sum_{i=1}^{k} \left(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set}\right)
      \]
Estimating Confidence Intervals: Classifier Models $M_1$ vs. $M_2$

- Suppose we have 2 classifiers, $M_1$ and $M_2$, which one is better?
- Use 10-fold cross-validation to obtain $\text{err}(M_1)$ and $\text{err}(M_2)$
- These mean error rates are just estimates of error on the true population of future data cases
- What if the difference between the 2 error rates is just attributed to chance?
  - Use a test of statistical significance
  - Obtain confidence limits for our error estimates
Estimating Confidence Intervals: Null Hypothesis

- Perform 10-fold cross-validation
- Assume samples follow a t distribution with $k-1$ degrees of freedom (here, $k=10$)
- Use t-test (or Student’s t-test)
- Null Hypothesis: $M_1$ & $M_2$ are the same
- If we can reject null hypothesis, then
  - we conclude that the difference between $M_1$ & $M_2$ is statistically significant
  - Chose model with lower error rate
Estimating Confidence Intervals: t-test

- If only 1 test set available: pairwise comparison
  - For \(i^{th}\) round of 10-fold cross-validation, the same cross partitioning is used to obtain \(\text{err}(M_1)_i\) and \(\text{err}(M_2)_i\)
  - Average over 10 rounds to get \(\overline{\text{err}(M_1)}\) and \(\overline{\text{err}(M_2)}\)
  - t-test computes t-statistic with \(k-1\) degrees of freedom:
    \[
    t = \frac{\overline{\text{err}(M_1)} - \overline{\text{err}(M_2)}}{\sqrt{\text{var}(M_1 - M_2)/k}}
    \]
    where
    \[
    \text{var}(M_1 - M_2) = \frac{1}{k} \sum_{i=1}^{k} \left[ \text{err}(M_1)_i - \text{err}(M_2)_i - (\overline{\text{err}(M_1)} - \overline{\text{err}(M_2)}) \right]^2
    \]

- If two test sets available: use non-paired t-test
  
  where
  \[
  \text{var}(M_1 - M_2) = \sqrt{\frac{\text{var}(M_1)}{k_1} + \frac{\text{var}(M_2)}{k_2}},
  \]
  where \(k_1\) & \(k_2\) are \# of cross-validation samples used for \(M_1\) & \(M_2\), resp.
Estimating Confidence Intervals:

Table for t-distribution

- **Symmetric**
- **Significance level**, e.g., *sig = 0.05* or 5% means $M_1$ & $M_2$ are **significantly different** for 95% of population
- **Confidence limit**, $z = \frac{\text{sig}}{2}$

### Table B: $t$-Distribution Critical Values

<table>
<thead>
<tr>
<th>df</th>
<th>.25</th>
<th>.20</th>
<th>.15</th>
<th>.10</th>
<th>.05</th>
<th>.025</th>
<th>.02</th>
<th>.01</th>
<th>.005</th>
<th>.0025</th>
<th>.001</th>
<th>.0005</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
<td>1.376</td>
<td>1.636</td>
<td>1.812</td>
<td>2.060</td>
<td>2.326</td>
<td>2.576</td>
<td>2.807</td>
<td>3.078</td>
<td>3.356</td>
<td>3.690</td>
<td>4.032</td>
</tr>
<tr>
<td>4</td>
<td>.727</td>
<td>.941</td>
<td>1.190</td>
<td>1.399</td>
<td>1.533</td>
<td>1.711</td>
<td>1.944</td>
<td>2.179</td>
<td>2.414</td>
<td>2.632</td>
<td>2.841</td>
<td>3.078</td>
</tr>
<tr>
<td>5</td>
<td>.711</td>
<td>.896</td>
<td>1.119</td>
<td>1.240</td>
<td>1.405</td>
<td>1.532</td>
<td>1.746</td>
<td>1.968</td>
<td>2.179</td>
<td>2.390</td>
<td>2.571</td>
<td>2.764</td>
</tr>
<tr>
<td>6</td>
<td>.700</td>
<td>.857</td>
<td>1.043</td>
<td>1.153</td>
<td>1.319</td>
<td>1.443</td>
<td>1.628</td>
<td>1.807</td>
<td>2.000</td>
<td>2.183</td>
<td>2.343</td>
<td>2.508</td>
</tr>
<tr>
<td>7</td>
<td>.679</td>
<td>.816</td>
<td>1.008</td>
<td>1.088</td>
<td>1.259</td>
<td>1.365</td>
<td>1.504</td>
<td>1.660</td>
<td>1.812</td>
<td>1.984</td>
<td>2.137</td>
<td>2.306</td>
</tr>
<tr>
<td>8</td>
<td>.668</td>
<td>.775</td>
<td>0.983</td>
<td>1.036</td>
<td>1.211</td>
<td>1.270</td>
<td>1.422</td>
<td>1.554</td>
<td>1.691</td>
<td>1.843</td>
<td>1.984</td>
<td>2.120</td>
</tr>
<tr>
<td>9</td>
<td>.660</td>
<td>.737</td>
<td>0.956</td>
<td>0.983</td>
<td>1.172</td>
<td>1.180</td>
<td>1.353</td>
<td>1.475</td>
<td>1.587</td>
<td>1.716</td>
<td>1.833</td>
<td>1.950</td>
</tr>
<tr>
<td>10</td>
<td>.650</td>
<td>.699</td>
<td>0.929</td>
<td>0.951</td>
<td>1.134</td>
<td>1.108</td>
<td>1.291</td>
<td>1.401</td>
<td>1.510</td>
<td>1.626</td>
<td>1.736</td>
<td>1.841</td>
</tr>
<tr>
<td>20</td>
<td>.606</td>
<td>.606</td>
<td>0.812</td>
<td>0.866</td>
<td>1.000</td>
<td>0.978</td>
<td>1.171</td>
<td>1.314</td>
<td>1.432</td>
<td>1.541</td>
<td>1.645</td>
<td>1.736</td>
</tr>
<tr>
<td>50</td>
<td>.500</td>
<td>.500</td>
<td>0.711</td>
<td>0.785</td>
<td>0.929</td>
<td>0.912</td>
<td>1.106</td>
<td>1.274</td>
<td>1.385</td>
<td>1.475</td>
<td>1.554</td>
<td>1.626</td>
</tr>
<tr>
<td>100</td>
<td>.500</td>
<td>.500</td>
<td>0.606</td>
<td>0.675</td>
<td>0.765</td>
<td>0.745</td>
<td>0.912</td>
<td>1.088</td>
<td>1.186</td>
<td>1.264</td>
<td>1.333</td>
<td>1.401</td>
</tr>
</tbody>
</table>

Confidence level $C$
Estimating Confidence Intervals: Statistical Significance

- Are $M_1$ & $M_2$ significantly different?
  - Compute $t$. Select significance level (e.g. $sig = 5\%$)
  - Consult table for t-distribution: Find $t$ value corresponding to $k-1$ degrees of freedom (here, 9)
  - t-distribution is symmetric: typically upper % points of distribution shown → look up value for confidence limit $z = sig/2$ (here, 0.025)
  - If $t > z$ or $t < -z$, then $t$ value lies in rejection region:
    - Reject null hypothesis that mean error rates of $M_1$ & $M_2$ are same
    - Conclude: statistically significant difference between $M_1$ & $M_2$
  - Otherwise, conclude that any difference is chance
Model Selection: ROC Curves

- **ROC (Receiver Operating Characteristics) curves**: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model
- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model

- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line
- A model with perfect accuracy will have an area of 1.0
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- K Nearest Neighbor Classification Methods
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary
Ensemble Methods: Increasing the Accuracy

- **Ensemble methods**
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, $M_1, M_2, \ldots, M_k$, with the aim of creating an improved model $M^*$

- **Popular ensemble methods**
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers
  - Ensemble: combining a set of heterogeneous classifiers
Bagging: Bootstrap Aggregation

- Analogy: Diagnosis based on multiple doctors’ majority vote
- Training
  - Given a set D of $d$ tuples, at each iteration $i$, a training set $D_i$ of $d$ tuples is sampled with replacement from D (i.e., bootstrap)
  - A classifier model $M_i$ is learned for each training set $D_i$
- Classification: classify an unknown sample $X$
  - Each classifier $M_i$ returns its class prediction
  - The bagged classifier $M^*$ counts the votes and assigns the class with the most votes to $X$
- Prediction: can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple
- Accuracy
  - Often significantly better than a single classifier derived from D
  - For noise data: not considerably worse, more robust
  - Proved improved accuracy in prediction
Boosting

- Analogy: Consult several doctors, based on a combination of weighted diagnoses—weight assigned based on the previous diagnosis accuracy
- How boosting works?
  - **Weights** are assigned to each training tuple
  - A series of k classifiers is iteratively learned
  - After a classifier $M_i$ is learned, the weights are updated to allow the subsequent classifier, $M_{i+1}$, to **pay more attention to the training tuples that were misclassified** by $M_i$
  - The final $M^*$ **combines the votes** of each individual classifier, where the weight of each classifier's vote is a function of its accuracy
- Boosting algorithm can be extended for numeric prediction
- Comparing with bagging: Boosting tends to have greater accuracy, but it also risks overfitting the model to misclassified data
Adaboost (Freund and Schapire, 1997)

- Given a set of $d$ class-labeled tuples, $(x_1, y_1), \ldots, (x_d, y_d)$
- Initially, all the weights of tuples are set the same (1/d)
- Generate $k$ classifiers in $k$ rounds. At round $i$,
  - Tuples from $D$ are sampled (with replacement) to form a training set $D_i$ of the same size
  - Each tuple’s chance of being selected is based on its weight
  - A classification model $M_i$ is derived from $D_i$
  - Its error rate is calculated using $D_i$ as a test set
  - If a tuple is misclassified, its weight is increased, o.w. it is decreased
- Error rate: $err(x_j)$ is the misclassification error of tuple $x_j$. Classifier $M_i$ error rate is the sum of the weights of the misclassified tuples:
  \[
  error(M_i) = \sum_{j}^{d} w_j \times err(x_j)
  \]
- The weight of classifier $M_i$’s vote is
  \[
  \log \frac{1 - error(M_i)}{error(M_i)}
  \]
Random Forest (Breiman 2001)

- Random Forest:
  - Each classifier in the ensemble is a *decision tree* classifier and is generated using a random selection of attributes at each node to determine the split
  - During classification, each tree votes and the most popular class is returned

- Two Methods to construct Random Forest:
  - Forest-RI (*random input selection*): Randomly select, at each node, $F$ attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
  - Forest-RC (*random linear combinations*): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)

- Comparable in accuracy to Adaboost, but more robust to errors and outliers
- Insensitive to the number of attributes selected for consideration at each split, and faster than bagging or boosting
Classification of Class-Imbalanced Data Sets

- Class-imbalance problem: Rare positive example but numerous negative ones, e.g., medical diagnosis, fraud, oil-spill, fault, etc.
- Traditional methods assume a balanced distribution of classes and equal error costs: not suitable for class-imbalanced data
- Typical methods for imbalance data in 2-class classification:
  - **Oversampling**: re-sampling of data from positive class
  - **Under-sampling**: randomly eliminate tuples from negative class
  - **Threshold-moving**: moves the decision threshold, \( t \), so that the rare class tuples are easier to classify, and hence, less chance of costly false negative errors
  - Ensemble techniques: Ensemble multiple classifiers introduced above
- Still difficult for class imbalance problem on multiclass tasks
Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- K Nearest Neighbor Classification Methods
- Decision Tree Induction
- Bayes Classification Methods
- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary
Classification is a form of data analysis that extracts models describing important data classes.

Effective and scalable methods have been developed for decision tree induction, Naive Bayesian classification, rule-based classification, and many other classification methods.

Evaluation metrics include: accuracy, sensitivity, specificity, precision, recall, $F$ measure, and $F_β$ measure.

Stratified k-fold cross-validation is recommended for accuracy estimation. Bagging and boosting can be used to increase overall accuracy by learning and combining a series of individual models.
Summary (II)

- Significance tests and ROC curves are useful for model selection.
- There have been numerous comparisons of the different classification methods; the matter remains a research topic.
- No single method has been found to be superior over all others for all data sets.
- Issues such as accuracy, training time, robustness, scalability, and interpretability must be considered and can involve trade-offs, further complicating the quest for an overall superior method.
References (1)

- C. M. Bishop. **Neural Networks for Pattern Recognition.** Oxford University Press, 1995
- P. K. Chan and S. J. Stolfo. **Learning arbiter and combiner trees from partitioned data for scaling machine learning.** KDD’95
- H. Cheng, X. Yan, J. Han, and C.-W. Hsu, **Discriminative Frequent Pattern Analysis for Effective Classification**, ICDE'07
- H. Cheng, X. Yan, J. Han, and P. S. Yu, **Direct Discriminative Pattern Mining for Effective Classification**, ICDE'08
- W. Cohen. **Fast effective rule induction.** ICML'95
- G. Cong, K.-L. Tan, A. K. H. Tung, and X. Xu. **Mining top-k covering rule groups for gene expression data.** SIGMOD'05


W. Li, J. Han, and J. Pei, *CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules*, ICDM'01.


M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. EDBT'96.


R. Rastogi and K. Shim. **Public: A decision tree classifier that integrates building and pruning.** VLDB’98.

J. Shafer, R. Agrawal, and M. Mehta. **SPRINT: A scalable parallel classifier for data mining.** VLDB’96.


P. Tan, M. Steinbach, and V. Kumar. **Introduction to Data Mining.** Addison Wesley, 2005.


X. Yin and J. Han. **CPAR: Classification based on predictive association rules.** SDM'03

H. Yu, J. Yang, and J. Han. **Classifying large data sets using SVM with hierarchical clusters.** KDD'03.
Issues: Evaluating Classification Methods

- **Accuracy**
  - classifier accuracy: predicting class label
  - predictor accuracy: guessing value of predicted attributes

- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)

- **Robustness**: handling noise and missing values

- **Scalability**: efficiency in disk-resident databases

- **Interpretability**
  - understanding and insight provided by the model

- **Other measures**, e.g., goodness of rules, such as decision tree size or compactness of classification rules
Predictor Error Measures

- Measure predictor accuracy: measure how far off the predicted value is from the actual known value
- **Loss function**: measures the error betw. $y_i$ and the predicted value $y_i'$
  - Absolute error: $| y_i - y_i' |$
  - Squared error: $(y_i - y_i')^2$
- Test error (generalization error): the average loss over the test set
  - Mean absolute error: $\frac{1}{d} \sum_{i=1}^{d} | y_i - y_i' |$
  - Mean squared error: $\frac{1}{d} \sum_{i=1}^{d} (y_i - y_i')^2$
  - Relative absolute error: $\frac{1}{\sum_{i=1}^{d} | y_i - \bar{y} |} \sum_{i=1}^{d} | y_i - y_i' |$
  - Relative squared error: $\frac{1}{\sum_{i=1}^{d} (y_i - \bar{y})^2} \sum_{i=1}^{d} (y_i - y_i')^2$

The mean squared-error exaggerates the presence of outliers

Popularly use (square) root mean-square error, similarly, root relative squared error
Scalable Decision Tree Induction Methods

- **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)
  - Builds an AVC-list (attribute, value, class label)
- **BOAT** (PODS’99 — Gehrke, Ganti, Ramakrishnan & Loh)
  - Uses bootstrapping to create several small samples
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