For written notes on this lecture, please read chapter 3 of The Practical Bioinformatician. Alternatively, please read “Rule-Based Data Mining Methods for Classification Problems in Biomedical Domains”, a tutorial at PKDD04 by Jinyan Li and Limsoon Wong, September 2004. http://www.comp.nus.edu.sg/~wongls/talks/pkdd04/

Bioinformatics and Biomarker Discovery

Part 2: Tools

Limsoon Wong
9 September 2011

Outline

• Overview of Supervised Learning

• Decision Trees Ensembles
  – Bagging

• Other Methods
  – K-Nearest Neighbour
  – Bayesian Approach
Overview of Supervised Learning

Computational Supervised Learning

• Also called classification

• Learn from past experience, and use the learned knowledge to classify new data

• Knowledge learned by intelligent algorithms

• Examples:
  – Clinical diagnosis for patients
  – Cell type classification
Data

- Classification application involves > 1 class of data. E.g.,
  - Normal vs disease cells for a diagnosis problem

- Training data is a set of instances (samples, points) with known class labels

- Test data is a set of instances whose class labels are to be predicted

Process

\[ f(X) \]

Training data: \( X \)

Class labels \( Y \)

\[ f(\bullet) \text{: A classifier, a mapping, a hypothesis} \]

\[ f(U) \]

Test data: \( U \)

Predicted class labels
Relational Representation of Patient Data

$n$ features (order of 1000)

\[
\begin{array}{ccccccc}
\text{gene}_1 & \text{gene}_2 & \text{gene}_3 & \text{gene}_4 & \cdots & \text{gene}_n \\
\hline
x_{11} & x_{12} & x_{13} & x_{14} & \cdots & x_{1n} \\
x_{21} & x_{22} & x_{23} & x_{24} & \cdots & x_{2n} \\
x_{31} & x_{32} & x_{33} & x_{34} & \cdots & x_{3n} \\
\vdots & \cdots & \cdots & \cdots & \ddots & \cdots \\
x_{m1} & x_{m2} & x_{m3} & x_{m4} & \cdots & x_{mn} \\
\end{array}
\]

$m$ samples

Class

Requirements of Biomedical Classification

- High accuracy/sensitivity/specificity/precision
- High comprehensibility
Importance of Rule-Based Methods

• Systematic selection of a small number of features used for the decision making
  ⇒ Increase the comprehensibility of the knowledge patterns

• C4.5 and CART are two commonly used rule induction algorithms—a.k.a. decision tree induction algorithms

Structure of Decision Trees

• Every path from root to a leaf forms a decision rule
  – If $x_1 > a_1$ & $x_2 > a_2$, then it’s A class
• C4.5, CART, two of the most widely used
• Easy interpretation, but accuracy generally unattractive
A Simple Dataset

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp</th>
<th>Humidity</th>
<th>Windy</th>
<th>class.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>75</td>
<td>70</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>Sunny</td>
<td>80</td>
<td>90</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>Sunny</td>
<td>85</td>
<td>85</td>
<td>false</td>
<td>Don’t</td>
</tr>
<tr>
<td>Sunny</td>
<td>72</td>
<td>95</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>Sunny</td>
<td>69</td>
<td>70</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>Overcast</td>
<td>72</td>
<td>90</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>Overcast</td>
<td>83</td>
<td>78</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>Overcast</td>
<td>64</td>
<td>65</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>Overcast</td>
<td>81</td>
<td>75</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>Rain</td>
<td>71</td>
<td>80</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>Rain</td>
<td>65</td>
<td>70</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>Rain</td>
<td>75</td>
<td>80</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>Rain</td>
<td>68</td>
<td>80</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>Rain</td>
<td>70</td>
<td>96</td>
<td>false</td>
<td>Play</td>
</tr>
</tbody>
</table>

9 Play samples
5 Don’t
A total of 14.

A Decision Tree

- Construction of a tree is equivalent to determination of the root node of the tree and the root node of its sub-trees

Exercise: What is the accuracy of this tree?
An Example

Source: Anthony Tung

Most Discriminatory Feature

- Every feature can be used to partition the training data
- If the partitions contain a pure class of training instances, then this feature is most discriminatory
Example of Partitions

- **Categorical feature**
  - Number of partitions of the training data is equal to the number of values of this feature

- **Numerical feature**
  - Two partitions

<table>
<thead>
<tr>
<th>Instance #</th>
<th>Outlook</th>
<th>Temp</th>
<th>Humidity</th>
<th>Windy</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunny</td>
<td>75</td>
<td>70</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>2</td>
<td>Sunny</td>
<td>80</td>
<td>90</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>3</td>
<td>Sunny</td>
<td>85</td>
<td>85</td>
<td>false</td>
<td>Don’t</td>
</tr>
<tr>
<td>4</td>
<td>Sunny</td>
<td>72</td>
<td>95</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>5</td>
<td>Sunny</td>
<td>69</td>
<td>70</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>6</td>
<td>Overcast</td>
<td>72</td>
<td>90</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>7</td>
<td>Overcast</td>
<td>83</td>
<td>78</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>8</td>
<td>Overcast</td>
<td>64</td>
<td>65</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>9</td>
<td>Overcast</td>
<td>81</td>
<td>75</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>10</td>
<td>Rain</td>
<td>71</td>
<td>80</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>11</td>
<td>Rain</td>
<td>65</td>
<td>70</td>
<td>true</td>
<td>Don’t</td>
</tr>
<tr>
<td>12</td>
<td>Rain</td>
<td>75</td>
<td>80</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>13</td>
<td>Rain</td>
<td>68</td>
<td>80</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>14</td>
<td>Rain</td>
<td>70</td>
<td>96</td>
<td>false</td>
<td>Play</td>
</tr>
</tbody>
</table>
A categorical feature is partitioned based on its number of possible values.

A numerical feature is generally partitioned by choosing a “cutting point.”
Steps of Decision Tree Construction

- Select the “best” feature as the root node of the whole tree

- Partition the dataset into subsets using this feature so that the subsets are as “pure” as possible

- After partition by this feature, select the best feature (wrt the subset of training data) as the root node of this sub-tree

- Recursively, until the partitions become pure or almost pure

Measures to Evaluate Which Feature is Best

- Gini index

- Information gain

- Information gain ratio

- T-statistics

- $\chi^2$

- ...
Gini Index

\[
gini(S) = \frac{\text{diff of two arbitrary specimen in } S}{\text{mean specimen in } S} \\
= \text{prob(getting two specimen of diff class in } S) \\
= 1 - \text{prob(getting two specimen of same class in } S) \\
= 1 - \sum_i \text{prob(getting specimen of class } i \text{ in } S)^2
\]

- Gini index is the expected value of the ratio of the difference of two arbitrary specimens to the mean value of all specimens.
- Closer to 0, means the samples are “pure”; thus the feature is “unexpected”.

Let \( D = \{C_1, ..., C_k\} \) be all the classes. Suppose we are currently at a node and \( D \) is the set of those samples that have been moved to this node. Let \( f \) be a feature and \( d[f] \) be the value of the feature \( f \) in a sample \( d \). Let \( S \) be a range of values that the feature \( f \) can take. Then the Gini index for \( f \) in \( D \) for the range \( S \) is defined as:

\[
gini_f^S(S) = 1 - \sum_{c_i \in D} \left( \frac{|\{d \in D | d \in C_i, d[f] \in S\}|}{|D|} \right)^2
\]

The purity of a split of the value range \( S \) of an attribute \( f \) by some split-point into subranges \( S_1 \) and \( S_2 \) is then defined as:

\[
gini_f^{S_1,S_2}(S_1,S_2) = \sum_{S \in (S_1,S_2)} \frac{|\{d \in D | d[f] \in S\}|}{|D|} \cdot gini_f^S(S)
\]

we choose the feature \( f \) and the split-point \( v \) that minimizes \( gini_f^{S_1,S_2}(S) \) over all possible alternative features and split-points.

- Gini index of a node if the weighted average of the purity (measured by Gini) of each subtree of the node.
**Decision Tree Ensembles**

- $h_1, h_2, h_3$ are independent classifiers with accuracy = 60%
- $C_1, C_2$ are the only classes
- $t$ is a test instance in $C_1$
- $h(t) = \arg\max_{C \in \{C_1, C_2\}} \{h_j \in \{h_1, h_2, h_3\} | h_j(t) = C\}$
- Then $\text{prob}(h(t) = C_1)$
  
  $= \text{prob}(h_1(t) = C_1, h_2(t) = C_1, h_3(t) = C_1) +$
  $\text{prob}(h_1(t) = C_1, h_2(t) = C_1, h_3(t) = C_2) +$
  $\text{prob}(h_1(t) = C_1, h_2(t) = C_2, h_3(t) = C_1) +$
  $\text{prob}(h_1(t) = C_2, h_2(t) = C_1, h_3(t) = C_1)$
  
  $= 0.60 \times 0.60 \times 0.60 + 0.60 \times 0.60 \times 0.40 +$
  $0.60 \times 0.40 \times 0.60 + 0.40 \times 0.60 \times 0.60 = 64.8\%$
Bagging

- Proposed by Breiman (1996)
- Also called Bootstrap aggregating
- Make use of randomness injected to training data

Main Ideas

Original training set: 50 p + 50 n

- Draw 100 samples with replacement
- A base inducer such as C4.5

A committee $H$ of classifiers:

$h_1$, $h_2$, ..., $h_k$
Decision Making by Bagging

Given a new test sample $T$

$$\text{bagged}(T) = \arg \max_{C_j \in \mathcal{U}} |\{ h_i \in \mathcal{H} \mid h_i(T) = C_j \}|$$

where $\mathcal{U} = \{C_1, \ldots, C_r\}$

Exercise: What does the above formula mean?

Summary of Ensemble Classifiers

Bagging          Random Forest
Rules may not be correct when applied to training data

AdaBoost.M1

Randomization Trees          CS4
Rules correct

Exercise: Describe the 3 decision tree ensemble classifiers not explained in this ppt
Concluding Remarks…

What have we learned?

• Decision Trees

• Decision Trees Ensembles
  – Bagging
References

- L. Breiman, Bagging predictors, Machine Learning, 24:123--140, 1996
- L. Breiman, Random forests, Machine Learning, 45:5-32, 2001
- Jinyan Li et al., Data Mining Techniques for the Practical Bioinformatician, The Practical Bioinformatician. Chapter 3, pages 35—70, WSPC, 2004

Exercise: Download a copy of WEKA. What are the names of classifiers in WEKA that correspond to C4.5 and SVM?