Data Mining with Customer Relationship Management

(For Class Use Only; Updated Fall 2014)

Part 1
Data Warehouse, Data Mining, and CRM
Why do we need a data warehouse?

- A data warehouse stores all our transaction data, customer data, employee data, product data, marketing data, partner data. The data can be used for data mining (knowledge creation and sharing).
- A data warehouse acts as a unifier of a multitude of isolated, distributed databases and aggregates, cleans, scrubs, and consolidate disparate data.

Web-Enabled Warehouses

- Web-enabled warehouses, known as data web houses, deliver real-time data storage.
- By aggregating data across multiple sources, they give decision makers the ability to run online queries, slide-and-dice data, and most importantly, improve information quality and accuracy.
Data Mining

Data mining is the process of extracting and presenting implicit knowledge, previously unknown, from large volumes of databases for actionable decisions.

Data mining can convert data into information and knowledge, such that the right decisions can be made.

Provides the mechanisms to deploy knowledge into operational systems, such that the right action occurs.

Data Mining and CRM

Data mining can do the customer/employee/partner relationship management analysis and forecasting that discovers the knowledge.
Customer Relationship

- Customer Propensity Analysis
- Market-Basket Analysis
- Customer Segmentation Identification
- Target Marketing Analysis
- Customer Retention Analysis
- Customer Profitability Forecasting
- Dynamic Pricing
- Cross-sell and up-sell.

Partner Relationship

- Sales/Revenue/Demand Forecasting
- Inventory Control, Monitoring and Forecasting
- Risk Analysis and Management
- Process/Quality Control and Monitoring
- Channel Analysis and Forecasting.
Employee Relationship

Employee performance measurement and reward program analysis.

Data mining for maintaining customer life cycle

Key stages in the customer lifecycle:

- **Prospects**: people who are not yet customers but are in the target market
  - **Responders**: prospects who show an interest in a product or service
- **Active customers**: people who are currently using the product or service
- **Former customers**: may be “bad” customers who did not pay their bills or who incurred high costs.
The most frequently used summary statistics include the following:

- **MAX** – the maximum value for a given predictor
- **MIN** – the minimum value for a given predictor
- **MEAN** – the average value for a given predictor.
**Example**

<table>
<thead>
<tr>
<th>Customer</th>
<th>Name</th>
<th>Prediction</th>
<th>Age</th>
<th>Balance</th>
<th>Income</th>
<th>Eyes</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Amy</td>
<td>No</td>
<td>62</td>
<td>$0</td>
<td>Medium</td>
<td>Brown</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>Al</td>
<td>No</td>
<td>53</td>
<td>$1,800</td>
<td>Medium</td>
<td>Green</td>
<td>M</td>
</tr>
<tr>
<td>3</td>
<td>Betty</td>
<td>No</td>
<td>47</td>
<td>$16,543</td>
<td>High</td>
<td>Brown</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>Bob</td>
<td>Yes</td>
<td>32</td>
<td>$45</td>
<td>Medium</td>
<td>Green</td>
<td>M</td>
</tr>
<tr>
<td>5</td>
<td>Carla</td>
<td>Yes</td>
<td>21</td>
<td>$2,300</td>
<td>High</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>6</td>
<td>Carla</td>
<td>No</td>
<td>27</td>
<td>$5,400</td>
<td>High</td>
<td>Brown</td>
<td>M</td>
</tr>
<tr>
<td>7</td>
<td>Donna</td>
<td>Yes</td>
<td>50</td>
<td>$165</td>
<td>Low</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>Don</td>
<td>Yes</td>
<td>46</td>
<td>$0</td>
<td>High</td>
<td>Blue</td>
<td>M</td>
</tr>
<tr>
<td>9</td>
<td>Edna</td>
<td>Yes</td>
<td>27</td>
<td>$500</td>
<td>Low</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>10</td>
<td>Edward</td>
<td>No</td>
<td>68</td>
<td>$1,200</td>
<td>Low</td>
<td>Blue</td>
<td>M</td>
</tr>
</tbody>
</table>

**Histogram as a Visualization Tool**

This diagram shows the number of customers of different ages and quickly tells the viewer that the majority of customers are over the age of 50.
## Two Classes of Data Mining Methods

There are two classes of knowledge discovery-driven data mining used in CRM:

- **Description**
  - Clustering
  - Association
  - Sequential association

- **Prediction**
  - Classification
  - Regression.
Description using Clustering

Clustering is to find groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

Since the class information is not known in advance, hence unsupervised learning is used.

Clustering Algorithm: \( k \)-Means

1. Specify in advance how many clusters are being sought, \( k \)
2. Choose \( k \) points (randomly) as cluster centers
3. Start with the \( k \) cluster centers,
   a) Instances are assigned to their closest cluster center according to the Euclidean distance. The distance between point \((x_i, y_i)\) to cluster center \((x_c, y_c)\) is calculated as:
   \[
   d_{i,c} = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}
   \]
   b) The centroid, or mean, of all instances in each cluster is calculated. These centroids are taken to be the new center values of their respective clusters.
   c) If the centroid in each clusters remains unchanged (which means the cluster centers have stabilized),
      return the \( k \) clusters
      else
      go to 3(a) with the new cluster center.
A 4-Point Example and Related Issues

Assume $X_1(1,1)$, $X_2(1,4)$, $X_3(10,4)$, $X_4(10,1)$ and $k=2$

What is your clustering outcome?

Why 2 different outcomes?

How to get the better one?

How to select the best $k$?

Applications

We often group the population by demographic information into segments that are useful for direct marketing and sales

To build these groupings, we use information such as income, age, occupation, housing, and race collected in the U.S. census

Then, we assign memorable nicknames to the clusters.
### Example: Segments

<table>
<thead>
<tr>
<th>Name</th>
<th>Income</th>
<th>Age</th>
<th>Education</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Blood Estates</td>
<td>Wealthy</td>
<td>35-54</td>
<td>College</td>
<td>Claritas Prizm</td>
</tr>
<tr>
<td>Shotguns and Pickups</td>
<td>Middle</td>
<td>35-64</td>
<td>High School</td>
<td>Claritas Prizm</td>
</tr>
<tr>
<td>Southside City</td>
<td>Poor</td>
<td>Mix</td>
<td>Grade School</td>
<td>Claritas Prizm</td>
</tr>
<tr>
<td>Living Off the Land</td>
<td>Middle-Poor</td>
<td>School Age Families</td>
<td>Low</td>
<td>Equifax MicroVision</td>
</tr>
<tr>
<td>University USA</td>
<td>Very low</td>
<td>Young-Mix</td>
<td>Medium to High</td>
<td>Equifax MicroVision</td>
</tr>
<tr>
<td>Sunset Years</td>
<td>Medium</td>
<td>Seniors</td>
<td>Medium</td>
<td>Equifax MicroVision</td>
</tr>
</tbody>
</table>

### Example: Data

<table>
<thead>
<tr>
<th>Customer</th>
<th>Name</th>
<th>Prediction</th>
<th>Age</th>
<th>Balance</th>
<th>Income</th>
<th>Eyes</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Amy</td>
<td>No</td>
<td>62</td>
<td>$0</td>
<td>Medium</td>
<td>Brown</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>Al</td>
<td>No</td>
<td>53</td>
<td>$1,800</td>
<td>Medium</td>
<td>Green</td>
<td>M</td>
</tr>
<tr>
<td>3</td>
<td>Betty</td>
<td>No</td>
<td>47</td>
<td>$16,543</td>
<td>High</td>
<td>Brown</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>Bob</td>
<td>Yes</td>
<td>32</td>
<td>$45</td>
<td>Medium</td>
<td>Green</td>
<td>M</td>
</tr>
<tr>
<td>5</td>
<td>Carla</td>
<td>Yes</td>
<td>21</td>
<td>$2,300</td>
<td>High</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>6</td>
<td>Carla</td>
<td>No</td>
<td>27</td>
<td>$5,400</td>
<td>High</td>
<td>Brown</td>
<td>M</td>
</tr>
<tr>
<td>7</td>
<td>Donna</td>
<td>Yes</td>
<td>50</td>
<td>$165</td>
<td>Low</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>Don</td>
<td>Yes</td>
<td>46</td>
<td>$0</td>
<td>High</td>
<td>Blue</td>
<td>M</td>
</tr>
<tr>
<td>9</td>
<td>Edna</td>
<td>Yes</td>
<td>27</td>
<td>$500</td>
<td>Low</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>10</td>
<td>Edward</td>
<td>No</td>
<td>68</td>
<td>$1,200</td>
<td>Low</td>
<td>Blue</td>
<td>M</td>
</tr>
</tbody>
</table>
If you were a pragmatic person, you might create three clusters because you think that marital happiness is mostly dependent on financial compatibility.
Is there another correct way to cluster?

- We might note some incompatibilities between 46-year old Don and 21-year old Carla (even though they both make very good incomes).
- We might instead consider age and some physical characteristics to be most important in creating clusters of friends.
- Another way you could cluster your friends would be based on their ages and on the color of their eyes.

---

More Romantic Clustering

<table>
<thead>
<tr>
<th>Customer</th>
<th>Name</th>
<th>Prediction</th>
<th>Age</th>
<th>Balance</th>
<th>Income</th>
<th>Eyes</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Carla</td>
<td>Yes</td>
<td>21</td>
<td>$2,300</td>
<td>High</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>9</td>
<td>Edna</td>
<td>Yes</td>
<td>27</td>
<td>$500</td>
<td>Low</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>6</td>
<td>Carla</td>
<td>No</td>
<td>27</td>
<td>$5,400</td>
<td>High</td>
<td>Brown</td>
<td>M</td>
</tr>
<tr>
<td>4</td>
<td>Bob</td>
<td>Yes</td>
<td>32</td>
<td>$45</td>
<td>Medium</td>
<td>Green</td>
<td>M</td>
</tr>
<tr>
<td>8</td>
<td>Don</td>
<td>Yes</td>
<td>46</td>
<td>$0</td>
<td>High</td>
<td>Blue</td>
<td>M</td>
</tr>
<tr>
<td>7</td>
<td>Donna</td>
<td>Yes</td>
<td>50</td>
<td>$165</td>
<td>Low</td>
<td>Blue</td>
<td>F</td>
</tr>
<tr>
<td>10</td>
<td>Edward</td>
<td>No</td>
<td>68</td>
<td>$1,200</td>
<td>Low</td>
<td>Blue</td>
<td>M</td>
</tr>
<tr>
<td>3</td>
<td>Betty</td>
<td>No</td>
<td>47</td>
<td>$16,543</td>
<td>High</td>
<td>Brown</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>Al</td>
<td>No</td>
<td>53</td>
<td>$1,800</td>
<td>Medium</td>
<td>Green</td>
<td>M</td>
</tr>
<tr>
<td>1</td>
<td>Amy</td>
<td>No</td>
<td>62</td>
<td>$0</td>
<td>Medium</td>
<td>Brown</td>
<td>F</td>
</tr>
</tbody>
</table>
Association analysis is to find the rules that will predict the occurrence of an item based on the occurrences of other items in the transaction. It is also known as Market Basket Analysis.

For example,

- “the customer will buy the bread if they buy the milk” with a high confidence and a high support.

Example

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Frozen Pizza, Cola, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Milk, Potato Chips</td>
</tr>
<tr>
<td>3</td>
<td>Cola, Frozen Pizza</td>
</tr>
<tr>
<td>4</td>
<td>Milk, Pretzels</td>
</tr>
<tr>
<td>5</td>
<td>Cola, Pretzels</td>
</tr>
</tbody>
</table>
If a customer purchases Frozen Pizza, then they will probably purchase Cola.
If a customer purchases Cola, then they will probably purchase Frozen Pizza.

### Association Analysis: 2 Steps

1. Discover the large itemsets that have transaction support (frequency) above a pre-specified minimum support, $minsup$.

2. Use each large itemset $I$ to generate rules $A \rightarrow B$ that have confidence above a pre-specified minimum confidence, $minconf$.
   - $X \subseteq I, Y \subseteq I$
   - $X \cup Y = I$
   - $X \cap Y = \emptyset$
   - $Conf(A \rightarrow B) = \frac{Sup(A \cup B)}{Sup(A)}$
Apriori Algorithm w/ Minisup

1. In the 1st iteration,
   a) Compose candidate 1-itemset, \( C_1 = \{ \text{all items in } D \} \), where \( D \) is the set of transactions
   b) The transactions in \( D \) are scanned to count the number of occurrences of each item in \( C_1 \)
   c) Filter the items in \( C_1 \) and keep those items with a support level higher than or equal to \( \text{minisup} \) in \( L_1 \). (\( \text{Minisup} \) refers to the minimum support level.)

2. In the \( k \)th iteration,
   a) Generate candidate \( k \)-itemset, \( C_k \), from \( L_{k-1} \)
   b) The transactions in \( D \) are scanned to test the support for each candidate in \( C_k \)
   c) Filter the candidates in \( C_k \) and keep those candidates with a support level higher than or equal to \( \text{minisup} \) in \( L_k \)

3. If \( L_k \) is empty, then there is no need to try \( L_{k+1} \).

Exercise: Association Analysis for Associative Rules

- An example run: modified from TKDE, December 1996, page 870
- TID Items
- -------+----------
- 100    A B C D
- 200    B C E
- 300    A B C E
- 400    B E
- \( \text{minisup} = 40\% \), \( \text{miniconf} = 70\% \)
- \( L_1: \{A\}, \{B\}, \{C\}, \{E\} \)
- \( C_2: \)
- \( L_2: \{A,B\}, \{A,C\}, \{B,C\}, \{B,E\}, \{C,E\} \)
- \( C_3: \{A,B,C\}, \{B,C,E\} \)
- \( L_3: \{A B C\}, \{B C E\} \)
- With \( \{B C E\} \) in \( L_3 \), rules: ??
  * \( \text{Conf}(X \rightarrow Y) = \frac{\text{Supp}(X \cup Y)}{\text{Supp}(X)} \)
- \( \{C, E\} \rightarrow B \).
Description using Sequential Association Rules

- Sequential association rules are used to find the association that if event \( X \) occurs, then event \( Y \) will occur within \( T \) events.
- The goal is to find patterns in one or more sequences that precede the occurrence of patterns in other sequences.
- So, the process of mining sequential association rules is composed mainly by the discovery of frequent patterns.

Advanced Topics for Association Analysis

- Association for classification
- Utility-based association analysis
- More copies of each item
- Negative association analysis
- One-scan of databases.
Prediction has two sub-subclasses
- Classification
- Regression.
Classification

Classification includes

- \(k\)NN
- Decision trees
- Rule induction
- Neural networks

Customer classification with Narex (in Colorado)

- A 101-person puzzle.

Classification using \(k\)-Nearest-Neighbors (\(k\)NN)

- Clustering and the \(k\)-Nearest-Neighbor classification technique are among the oldest techniques used in data mining
- With \(k\)NN, the \(k\) nearest neighbors are found and their majority class is used for the classification of a new instance
- Thus, \(k\)NN is a prediction approach.
Clustering like $k$ Nearest Neighbors

The $k$NN algorithm is basically a refinement of clustering, in the sense that they both use distance in some feature space to create their structure in the data or prediction.

Classification using Decision Trees

A decision tree is a predictive model that can be viewed as a tree

Specifically, each branch of the tree is a classification question, and the leaves of the tree are partitions of the dataset with their classification.
2.2 Decision Tree Construction

<table>
<thead>
<tr>
<th>ORDER</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>a</td>
<td>#1</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>a</td>
<td>a0</td>
<td>0</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>b</td>
<td>c1</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>b</td>
<td>b0</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>a</td>
<td>c1</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>b</td>
<td>a0</td>
<td>#1</td>
<td>T</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>e</td>
<td>c0</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>e</td>
<td>b1</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>e</td>
<td>b0</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>a</td>
<td>a0</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>e</td>
<td>c1</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>e</td>
<td>a0</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>a</td>
<td>b0</td>
<td>0</td>
<td>F</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>a</td>
<td>a1</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>b</td>
<td>a1</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Decision Tree Construction (2)

S1: $T \leftarrow$ the whole training set. Create a $T$ node.

S2: If all examples in $T$ are positive examples of a specific class, create a ‘yes’ node with $T$ as its parent and stop; If all examples in $T$ are negative, create a ‘no’ node with $T$ as its parent and stop.

S3: Select an attribute $X$ with values $V_1, ..., V_N$ and partition $T$ into subsets $T_1, ..., T_N$ according to their values on $X$. Create $N$ $T_i$ nodes ($i = 1, ..., N$) with $T$ as their parent and $X = V_i$ as the label of the branch from $T$ to $T_i$.

S4: For each $T_i$ do: $T \leftarrow T_i$ and goto S2.
Exercise: Construct a Decision Tree with Golf Data

Table 4.1. Cases of Play and Don’t Play (adapted from [Quinlan 86a])

<table>
<thead>
<tr>
<th>Order</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rain</td>
<td>hot</td>
<td>high</td>
<td>true</td>
<td>Don’t Play</td>
</tr>
<tr>
<td>2</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>Don’t Play</td>
</tr>
<tr>
<td>3</td>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>4</td>
<td>overcast</td>
<td>mild</td>
<td>normal</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>5</td>
<td>rain</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>6</td>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>Play</td>
</tr>
<tr>
<td>7</td>
<td>sunny</td>
<td>hot</td>
<td>normal</td>
<td>true</td>
<td>Don’t Play</td>
</tr>
<tr>
<td>8</td>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>Don’t Play</td>
</tr>
<tr>
<td>9</td>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>10</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>11</td>
<td>rain</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>Play</td>
</tr>
<tr>
<td>12</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>Don’t Play</td>
</tr>
<tr>
<td>13</td>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>Don’t Play</td>
</tr>
<tr>
<td>14</td>
<td>rain</td>
<td>mild</td>
<td>normal</td>
<td>true</td>
<td>Don’t Play</td>
</tr>
</tbody>
</table>

ID3 and C4.5: Key ideas

Suppose \( T = PE \cup NE \) where \( PE \) is the set of positive examples and \( NE \) is the set of negative examples, \( p = |PE| \) and \( n = |NE| \).

\[
I(p, n) = \begin{cases} 
-\frac{p}{p+n} \log_{2} \frac{p}{p+n} - \frac{n}{p+n} \log_{2} \frac{n}{p+n} & \text{when } p \neq 0 \& n \neq 0 \\
0 & \text{otherwise} 
\end{cases}
\]  

(1)

If attribute \( X \) with value domain \( \{v_1, \ldots, v_N\} \) is used for the root of the decision tree, it will partition \( T \) into \( \{T_1, \ldots, T_N\} \) where \( T_i \) contains those examples in \( T \) that have value \( v_i \) of \( X \). Let \( T_i \) contain \( p_i \) examples of \( PE \) and \( n_i \) of \( NE \). The expected information required for the subtree for \( T_i \) is \( I(p_i, n_i) \).
ID3 and C4.5: Key ideas (2)

The expected information required for the tree with $X$ as root, $EI(X)$, is then obtained as weighted average.

$$EI(X) = \sum_{i=1}^{N} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$  \hspace{1cm} (2)

where the weight for the $i$-th branch is the proportion of the examples in $T$ that belong to $T_i$. The information gained by branching on $X$, $G(X)$, is therefore

$$G(X) = I(p, n) - EI(X).$$  \hspace{1cm} (3)

ID3 and C4.5: Key ideas (3)

ID3 examines all candidate attributes, chooses $X$ to maximize $G(X)$, and then uses the same process recursively to construct decision trees for residual subsets $T_1, ..., T_N$.

C4.5 adopts a new heuristic, the gain ratio criterion, instead of $G(X)$, for selecting tests in decision tree generation. In the gain ratio criterion, $G(X)/IV(X)$ is used to replace $G(X)$ where

$$IV(X) = \sum_{i=1}^{N} \frac{p_i + n_i}{p + n} \log_2 \left( \frac{p_i + n_i}{p + n} \right)$$  \hspace{1cm} (4)
Viewing Decision Trees as Segmentation with a Purpose

- From a business perspective, decision trees can be viewed as creating a segmentation of the original dataset (each segment would be one of the leaves of the tree)
- Segmentation of customers, products, and sales regions is something that marketing managers have been doing for many years
- This segmentation has been performed in order to get a high-level view of a large amount of data – the records within each segmentation were somewhat similar.

Other topics in ID3-like algorithms

- Incremental induction: The windowing technique.
- Noise handling: Halt tree growth when no more significant information gain can be found.
- Sensible mechanisms for handling missing information.
- Post-pruning of decision trees.
- Decompiling decision trees into production rules.
- Binarization of decision trees.
- Structured induction.
- Real-valued attributes: Discretization.
- Classification of more than 2 classes with a decision tree.
Classification using Rule Induction

A rule-based system (RBS) embeds existing rules of thumb and heuristics to facilitate fast and accurate decision making.

Five conditions must be met for them to work well:
1. You must know the varieties in your problem/question
2. You must be able to express them in hand, including numerical terms (e.g. dollars)
3. The rules specified must cover all these variables
4. There is not much overlap between rules
5. The rules must have been validated

RBS system require continuous manual change and conflict resolution.

2.1 Rule Induction: A Simplified Example

Rules produced by HCV:

\begin{align*}
\text{ORDER} & \quad X_1 & \quad X_2 & \quad X_3 & \quad X_4 & \quad \text{CLASS} \\
1 & 1 & a & b & F \\
2 & 1 & a & a & F \\
3 & 1 & b & c & F \\
4 & 0 & b & b & T \\
5 & 0 & a & c & F \\
6 & 1 & b & a & F \\
7 & 1 & c & c & F \\
8 & 1 & c & b & T \\
9 & 0 & c & b & T \\
10 & 0 & a & a & T \\
11 & 0 & c & e & T \\
12 & 0 & c & a & T \\
13 & 1 & a & b & F \\
14 & 0 & a & a & T \\
15 & 0 & b & a & T \\
\end{align*}

\[ X_2 = b \]
\[ \lor \quad X_1 = 0 \quad \Rightarrow \quad X_4 = 0 \quad \Rightarrow \quad \text{The T class.} \]
\[ X_1 = 0 \quad \lor \quad X_2 = [c,a] \]
\[ \Rightarrow \quad \text{The F class.} \]

Advantages:
1. Implied but not explicit in the database
2. Smaller in volume
3. More general in description
C4.5 Rules: Steps

1. Traverse a decision tree to obtain a number of conjunctive rules
   - Each part from the root to a leaf in the tree corresponds to a conjunctive rule with the leaf as its conclusion

2. Check each condition in each conjunctive rule to see if it can be dropped without more misclassification than expected on the original training examples or new test examples

3. If some conjunctive rules are the same after Step 2, then keep only one of them.

Rule Induction with 1R ("1-rule")

Generates a one-level decision tree, which is expressed in the form of a set of rules that all test one particular attribute.

For each attribute Ai,
   For each value Vij of Ai, make a rule as follows:
   count how often each class appears
   find the most frequent class Ck
   make a rule "If Ai = Vij then class = Ck"
Choose an Ai with the smallest error rate for its rules
An Example Run with Table 4.1

<table>
<thead>
<tr>
<th>attribute</th>
<th>rules</th>
<th>errors</th>
<th>total errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  outlook</td>
<td>rain -&gt; no</td>
<td>3/6 (*)</td>
<td>4/14 (**)</td>
</tr>
<tr>
<td></td>
<td>overcast -&gt; yes</td>
<td>0/3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sunny -&gt; no</td>
<td>1/5</td>
<td></td>
</tr>
<tr>
<td>2  temperature</td>
<td>hot -&gt; no</td>
<td>2/5</td>
<td>6/14</td>
</tr>
<tr>
<td></td>
<td>mild -&gt; yes</td>
<td>2/5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cool -&gt; no</td>
<td>2/4 (*)</td>
<td></td>
</tr>
<tr>
<td>3  humidity</td>
<td>high -&gt; no</td>
<td>3/6 (*)</td>
<td>7/14</td>
</tr>
<tr>
<td></td>
<td>normal -&gt; no</td>
<td>4/8</td>
<td></td>
</tr>
<tr>
<td>4  windy</td>
<td>true -&gt; no</td>
<td>2/6</td>
<td>4/14 (**)</td>
</tr>
<tr>
<td></td>
<td>false -&gt; yes</td>
<td>4/8</td>
<td></td>
</tr>
</tbody>
</table>

• **Break the ties**
  (1) 3/6 and 3/6 for "outlook: rain -> no": take any consistent way
  (2) 4/14 and 4/14 at total errors: Occam's razor

• **Windy is chosen with the following rules.**
  Windy:
  true ==> not play
  false ==> play

• **Simple, cheap, and ("astonishingly – even embarrassingly") often comes up with good rules!**

Regression

- **Statistical regression** is a supervised learning technique that generalizes a set of numerical data by creating a mathematical question relating one or more input attributes to a single numeric output attribute
- **E.g. Life insurance promotion** is the attribute whose value is to be determined by a linear combination of attributes such as credit card insurance and sex.
Linear Regression

- Linear regression is similar to the task of finding the line that minimizes the total distance to a set of data.

Predictor and Prediction

- The line will take a given value for a predictor and map it into a given value for prediction.
- The actual equation would look something like:
  \[ \text{Prediction} = a + b \times \text{Predictor} \]
- This is just the equation for a line:
  \[ Y = a + b \times X. \]
### What if the pattern in a higher dimensional space?

- Adding more predictors can produce more complicated lines that take more information into account hence make a better prediction.
- This is called multiple line regression and might have an equation like the following if 5 predictors were used (X₁, X₂, X₃, X₄, X₅):
  \[ Y = a + b₁(X₁) + b₂(X₂) + b₃(X₃) + b₄(X₄) + b₅(X₅). \]

### What if the pattern in my data doesn’t look like a Straight line?

- By transforming the predictors by squaring, cubing, or taking their square root, create more complex models that are no longer simple shapes as lines.
- This is called non-linear regression.
  \[ Y = a + b₁(X₁) + b₂(X₁)(X₁). \]
Part 5

Artificial Neural Networks

Neural Networks

[Diagram of a neuron showing input, dendrite, soma, axon, output, and synapses.]

- Information enters the nerve cell at the synaptic site on the dendrite.
- Propagated action potentials leave the soma-dendrite complex to travel to the axon terminals.
- The axon terminal carries information to other cells.
- Synapse 1 and Synapse 2 are shown.
**An Artificial Neuron**

- Receives inputs $X_1, X_2, \ldots, X_p$ from other neurons or environment
- Inputs fed-in through connections with “weights”
- Total input $I = \sum w_i X_i$ where $w_i$ are weights
- Transfer function (activation function) $f$ converts the inputs to output
- Output $V$ goes to other neurons or environment.

**Neural Networks – What and When**

- A number of processors are interconnected, suggestive of the connections between neurons in a human brain
- Learning by a process of trial and error
- They are especially useful when very limited inputs from domain experts exist or when you have the data but lack experts to make judgments about it.
Artificial Neuron Configuration

A Simple Example

<table>
<thead>
<tr>
<th>Mobile</th>
<th>MP3</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Artificial Neuron Configuration with Bias

\[ \sum_{i=1}^{m} bias + (w^i x^i) \]

A Simple Model with Bias

\[ A = \sum_{n=1}^{N+1} w_i * I_i \]
There are various choices for Transfer / Activation functions.

**Tanh**
\[ f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]

**Logistic**
\[ f(x) = \frac{e^x}{1 + e^x} \]

**Threshold**
\[ f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 1 \end{cases} \]
### Number of Hidden Layers can be

- **None**
- **One**
- **More**

### Prediction

#### Input: $X_1, X_2, X_3$

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>Output: $Y$</th>
<th>Model: $Y = f(X_1, X_2, X_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>2</td>
<td></td>
<td>$0.2 = 0.5 \times 1 - 0.1 \times (-1) - 0.2 \times 2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$f(x) = \frac{e^x}{1 + e^x}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$f(0.2) = \frac{e^{0.2}}{1 + e^{0.2}} = 0.55$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Predicted $Y = 0.478$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suppose Actual $Y = 2$, Then</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Prediction Error = $(2 - 0.478) = 1.522$</td>
</tr>
</tbody>
</table>
A Collection of Neurons form a “Layer”

Input Layer
- Each neuron gets ONLY one input, directly from outside

Hidden Layer
- Connects Input and Output layers

Output Layer
- Output of each neuron directly goes to outside

How to build the model?

Input: X₁ X₂ X₃  
Output: Y  
Model: Y = f(X₁ X₂ X₃)

# Input Neurons = # Inputs = 3  
# Output Neurons = # Outputs = 1

# Hidden Layer = ???  
Try 1  
No fixed strategy

# Neurons in Hidden Layer = ???  
Try 2  
By trial and error

Architecture is now defined … How to get the weights ???

Given the architecture, there are 8 weights to decide:
W = (W₁, W₂, ..., W₈)

Training data: (Yᵢ, X₁ᵢ, X₂ᵢ, ..., Xₚᵢ)  i= 1,2,...,n
Given a particular choice of W, we will get predicted Y's
(V₁, V₂, ..., Vₙ)

They are function of W.
Choose W such that over all prediction error E is minimized

E = Σ (Yᵢ - Vᵢ)²
How to train the model?

- Start with a random set of weights.
- Feed forward the first observation through the net $X_1 \rightarrow \text{Network} \rightarrow V_1$; Error $= (Y_1 - V_1)$
- Adjust the weights so that this error is reduced (network fits the first observation well)
- Feed forward the second observation. Adjust weights to fit the second observation well
- Keep repeating till you reach the last observation
- This finishes one CYCLE through the data
- Perform many such training cycles till the overall prediction error $E$ is small.

$E = \Sigma (Y_i - V_i)^2$

Weight Adjustment during Back Propagation

$V_i$ – the prediction for $i^{th}$ observation – is a function of the network weight vector $W = (W_1, W_2, \ldots)$

Hence, $E$, the total prediction error is also a function of $W$

$E(W) = \Sigma [Y_i - V_i(W)]^2$
Training Algorithm

Decide the network architecture
(# Hidden layers, #Neurons in each Hidden Layer)

Decide the Learning parameter and Momentum

Initialize the Network with random weights

Do till Convergence criterion is met

For I = 1 to # Training Data points

Feed forward the I-th observation thru the Net
Compute the prediction error on I-th observation
Back propagate the error and adjust weights

Next I

Check for Convergence

End Do

\[ E = \sum (Y_i - V_i)^2 \]

An Example Application

<table>
<thead>
<tr>
<th>Age</th>
<th>Hidden Layer</th>
<th>Output Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age &lt; 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age between 30 and 50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age &gt; 50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>House</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Car</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kowloon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Input Nodes

Car Magazine
House Magazine
Sports Magazine
Music Magazine
Comic Magazine
Exercise: Problem

[Diagram showing scatter plot with age on the x-axis and some data points highlighted]

Exercise: Model

[Diagram showing a model with age on the x-axis and categories on the y-axis]
A Brief History of ANN

- McCulloch-Pitts's theory: Warren McCulloch and Walter Pitts in 1943.
- 1949 with a book, "The Organization of Behavior" written by Donald Hebb.
- 1956: the Dartmouth workshop on AI.
- 1969: Minsky and Papert (1969) at MIT.
- In 1982 John Hopfield of Caltech presented Hopfield model.
Purely numerical methods have been supplemented with visual methods.
This has led to the emergence of “Visual Data Mining”.
**Definition**: Visual data mining is a collection of interactive reflective methods that support exploration of data sets by dynamically adjusting parameters to see how they affect the information being presented.
Data Mining and CRM

Data Collection → Data Preprocessing

Building Data Mining Models → Variable Selection

Model Evaluation & Selection → Calculation

Data Mining Process

- **Data Preprocessing**
  - Problem definition
  - Data selection
  - Data preparation

- **Model Design**
  - Field selection
  - Technique selection
  - Segmentation

- **Data Analysis**
  - Data transformation
  - Model creation
  - Model testing
  - Interpretation / evaluation

- **Output Generation**
  - Reports
  - Monitors / Agents
  - Models / application
  - Data Visualization.
Roadmap of Building a CRM Data Warehouse for KCRM

1. The Planning Phase
2. The Design and Implementation Phase
3. Usage, Support, and Enhancement Phase.

(1) The Planning Phase

Four processes in this phase:
- The Business Discovery Process
- The Information Discovery Process
- Logical Data Model Design Process
- The Data Warehouse Architecture Design Process.
The Business Discovery Process

- The Business Discovery Process determines practical, information-based solutions to issues that the organization is facing.
- The emphasis is on the **business issues**, not technology.

The Information Discovery Process

- The Information Discovery Process helps enterprise refine the solution requirements by **validating critical business needs and information requirements**.
Data Modeling

Data modeling is used to show the customer how data can be turned into useful information and used to address the critical issues.

Data Modeling (con’t)

Run a walk-through to help identify the type of data warehouse best suited for your organizational plans and requirements, which can be comprised of dynamic presentations and interactive interviews.

Educate decision-makers, the user community, and technical or operations personnel about the advantages and disadvantages of the different data warehouse architectures, database engines, access methods, management methods and data sourcing methods is necessary.
The Logical Data Model Design Process

Logical Data Model Design process produces a logical data model for particular solutions, including the confirmation of requirements, creation of a project-plan, and the generation of the logical data model showing relationships and attributes.
### The Data Warehouse Architecture Design Process

- The Data Warehouse Architecture Design Process designs the specific architecture for the customer-defined environment and specifies the data warehouse location (centralized, distributed), network requirements, types of users, access, and so on.

### (2) The Design and Implementation Phase

- Develops full-scale design and implementation plans for the construction of the data warehouse.
Technology Assessment

- Technology Assessment ensures that no technical issues could prevent the implementation of a desired solution.
- It assesses the hardware, network and software environment, and analyzes the requirements for remote data access, data sharing, and backup/restart/recovery.
- It defines and prioritizes issues that could impede solution implementation, and produces a follow-up plan to eliminate issues.

Data and Function Assessments

- Data and Function Assessments evaluate the existing data structures and their characteristics to ensure that data warehouse sourcing requirements are met and that the data models supporting the solution truly address the business requirements.
- The Function Assessment identifies the technology and business processes that will be supported by a data warehouse, and confirms that the system under consideration will meet business needs.
Change Readiness Assessment

- Change Readiness Assessment defines how people using the data warehouse are affected by it and how they might react to the changes caused by implementing a data warehouse.
- It should explore barriers to success caused by cultural issues and identifies potential education to address the problems.
- It focuses on the impact of the proposed solution on the technical and user communities and their disposition to accept the changes.

Education and Support Assessments

- Education and Support Assessments will organize and plan education for project participants and end-users to support the integration of a data warehouse into their environment.
- The support assessment identifies the requirements for ongoing support of a data warehouse.
Knowledge Discovery Model Development

- Knowledge Discovery Model Development begins when conventional database access methods such as Structure Query Language (SQL) or Online Analytical Processing (OLAP) are not enough to solve certain business problems.
- Knowledge Discovery utilizes predictive data modeling to make more informed decisions about these problems.

Data Mining and Analytical Application

- Data Mining and Analytical Application select data mining tools or analytical application that best fits the business problem defined by the Knowledge Discovery service.
Client/Server Application Process

Client/Server Application Process includes both client/server design and development
- The client/server design provides for specific applications and the query interface required, and includes an iterative definition and prototyping of the desired applications.
- The results of the design process is a working prototype of the applications, a specification of the technical requirements of the design, specific deliverables, and development tools to support implementation, as well as complete plans for the required applications.

The Application Development Process

The Application Development Process implements the query interface for a data warehouse solution
- It uses the prototypes, specifications, and recommended tool sets and other outputs of the design service to develop and validate the applications that give users the data and information they need to perform their business functions.
Data Warehouse Physical Database Design Process

- Data Warehouse Physical Database Design process provides the customer with the physical database design and implementation optimized for a data warehouse.
- The primary activities in this service include translating the logical data model to a physical database design, database construction, design optimization, and functional testing of the constructed database.
- It also provides design guidelines appropriate for the environment and for the specific database platform used in this project.

Data Transformation Process

- The Data Transformation Process designs and develops the utilities and programming that allow the data warehouse database to be loaded and maintained.
The Data Warehouse Management Process

- The Data Warehouse Management Process implements the data, network, systems, and operations management procedures need to successfully manage a data warehouse.
- Included are data maintenance routines to update, load, backup, archive, administer, and restore/recover data, ensuring consistency and compatibility with existing procedures.

(3) The Usage, Support and Enhancement Phase

- There are five services and programs in this phase.
Enterprise System Support

Enterprise System Support provides three tiers of integrated system support for all solution components, such as the database, tools, applications, basic software and hardware.

Data Warehouse Logical Data Model and Physical Design Review

- Data Warehouse Logical Data Model Review and Physical Design Review process adds skills to your organization to allow use of your own staff
- Your external consultants review the requirements of the users along with their model construction, and offer analysis and suggestions for improvement.
Data Warehouse Tuning

Data Warehouse Tuning would typically be engaged when performance problems are encountered.

Capacity Planning

Capacity Planning helps enterprises plan for the initial definitions of capacity and sizing, and then for the expansion of their data warehouse.

Expansion includes the addition of applications, users, data, remote applications, and operations.
Data Warehouse Audit

- Data Warehouse Audit helps ascertain the business value of an existing data warehouse by validating its current state against its business “best practices”
- The areas of assessment include flexibility and extensibility of the database design, meta-data richness, consistency, and consumer access.

References