Statistical Methods in Data Mining

Decision Trees

Professor Dr. Gholamreza Nakhaeizadeh
content

Decision Trees

- Introduction
- Example: Credit Rating
- Example: Computer buyers
- Attribute selection measure in Decision Trees
- Construction of Decision Trees
- Gain Ratio
- Gini Index
- Overfitting
- Pruning
Decision Trees (DT)

Introduction

- DT are classification tools
- Class Variable (Target Variable): Nominal
- Attributes: Nominal or continuous-valued
- Top Down construction based on heuristic methods by using training data (Greedy instead of completely search: tends to find good solutions quickly, but not always optimal ones)
Simple fictive example; Credit Rating in a Bank

<table>
<thead>
<tr>
<th>Income &gt;2000</th>
<th>Car</th>
<th>Gender</th>
<th>Credit Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>no</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 2</td>
<td>no statement</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 3</td>
<td>no statement</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 4</td>
<td>no</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 5</td>
<td>yes</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 6</td>
<td>yes</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 7</td>
<td>no statement</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 8</td>
<td>yes</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 9</td>
<td>no statement</td>
<td>no</td>
<td>M</td>
</tr>
<tr>
<td>Customer 10</td>
<td>no</td>
<td>no</td>
<td>F</td>
</tr>
</tbody>
</table>
Simple fictive example; Credit Rating in a Bank

<table>
<thead>
<tr>
<th>Income &gt;2000</th>
<th>Car</th>
<th>Gender</th>
<th>Credit Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>no</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 2</td>
<td>no statement</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 3</td>
<td>no statement</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 4</td>
<td>no</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 5</td>
<td>yes</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 6</td>
<td>yes</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 7</td>
<td>no statement</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 8</td>
<td>yes</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 9</td>
<td>no statement</td>
<td>no</td>
<td>M</td>
</tr>
<tr>
<td>Customer 10</td>
<td>no</td>
<td>no</td>
<td>F</td>
</tr>
</tbody>
</table>
## Simple fictive example; Credit Rating in a Bank

<table>
<thead>
<tr>
<th></th>
<th>Income &gt;2000</th>
<th>Car</th>
<th>Gender</th>
<th>Credit Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>no</td>
<td>yes</td>
<td>F</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 2</td>
<td>no statement</td>
<td>no</td>
<td>F</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 3</td>
<td>no statement</td>
<td>yes</td>
<td>M</td>
<td>good</td>
</tr>
<tr>
<td>Customer 4</td>
<td>no</td>
<td>yes</td>
<td>M</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 5</td>
<td>yes</td>
<td>yes</td>
<td>M</td>
<td>good</td>
</tr>
<tr>
<td>Customer 6</td>
<td>yes</td>
<td>yes</td>
<td>F</td>
<td>good</td>
</tr>
<tr>
<td>Customer 7</td>
<td>no statement</td>
<td>yes</td>
<td>F</td>
<td>good</td>
</tr>
<tr>
<td>Customer 8</td>
<td>yes</td>
<td>no</td>
<td>F</td>
<td>good</td>
</tr>
<tr>
<td>Customer 9</td>
<td>no statement</td>
<td>no</td>
<td>M</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 10</td>
<td>no</td>
<td>no</td>
<td>F</td>
<td>bad</td>
</tr>
</tbody>
</table>
## Simple fictive example; Credit Rating in a Bank

<table>
<thead>
<tr>
<th>Customer</th>
<th>Income &gt;2000</th>
<th>Car</th>
<th>Gender</th>
<th>Credit Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>no</td>
<td>yes</td>
<td>F</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 2</td>
<td>no statement</td>
<td>no</td>
<td>F</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 3</td>
<td>no statement</td>
<td>yes</td>
<td>M</td>
<td>good</td>
</tr>
<tr>
<td>Customer 4</td>
<td>no</td>
<td>yes</td>
<td>M</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 5</td>
<td>yes</td>
<td>yes</td>
<td>M</td>
<td>good</td>
</tr>
<tr>
<td>Customer 6</td>
<td>yes</td>
<td>yes</td>
<td>F</td>
<td>good</td>
</tr>
<tr>
<td>Customer 7</td>
<td>no statement</td>
<td>yes</td>
<td>F</td>
<td>good</td>
</tr>
<tr>
<td>Customer 8</td>
<td>yes</td>
<td>no</td>
<td>F</td>
<td>good</td>
</tr>
<tr>
<td>Customer 9</td>
<td>no statement</td>
<td>no</td>
<td>M</td>
<td>bad</td>
</tr>
<tr>
<td>Customer 10</td>
<td>no</td>
<td>no</td>
<td>F</td>
<td>bad</td>
</tr>
</tbody>
</table>
# Simple fictive example; Credit Rating in a Bank

<table>
<thead>
<tr>
<th>Income &gt;2000</th>
<th>Car</th>
<th>Gender</th>
<th>Credit Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>no</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 2</td>
<td>no statement</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 3</td>
<td>no statement</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 4</td>
<td>no</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 5</td>
<td>yes</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 6</td>
<td>yes</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 7</td>
<td>no statement</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 8</td>
<td>yes</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 9</td>
<td>no statement</td>
<td>no</td>
<td>M</td>
</tr>
<tr>
<td>Customer 10</td>
<td>no</td>
<td>no</td>
<td>F</td>
</tr>
</tbody>
</table>
# Simple fictive example; Credit Rating in a Bank

<table>
<thead>
<tr>
<th>Income &gt;2000</th>
<th>Car</th>
<th>Gender</th>
<th>Credit Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>no</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 2</td>
<td>no statement</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 3</td>
<td>no statement</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 4</td>
<td>no</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 5</td>
<td>yes</td>
<td>yes</td>
<td>M</td>
</tr>
<tr>
<td>Customer 6</td>
<td>yes</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 7</td>
<td>no statement</td>
<td>yes</td>
<td>F</td>
</tr>
<tr>
<td>Customer 8</td>
<td>yes</td>
<td>no</td>
<td>F</td>
</tr>
<tr>
<td>Customer 9</td>
<td>no statement</td>
<td>no</td>
<td>M</td>
</tr>
<tr>
<td>Customer 10</td>
<td>no</td>
<td>no</td>
<td>F</td>
</tr>
</tbody>
</table>
Simple fictive example; Credit Rating in a Bank

### Classifier

<table>
<thead>
<tr>
<th>Condition</th>
<th>Credit Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Income &gt; 2000 = yes</td>
<td>good</td>
</tr>
<tr>
<td>Income &gt; 2000 = no</td>
<td>bad</td>
</tr>
<tr>
<td>Income = no statement &amp; Car = yes</td>
<td>good</td>
</tr>
<tr>
<td>Income = no statement &amp; Car = no</td>
<td>bad</td>
</tr>
</tbody>
</table>

This classifier can be regarded as an Inductive expert systems

### Rating new Customers

- Rating a new customer with income 3000 = good
- Rating a new customer who has no car and made no income statement = bad
- ...
Credit rating: decision tree construction

Root

Income > 2000
?

yes

no state.

no

Leaf

Node
car

good

bad

good

bad

10 custom.
totally: 5 good rate 5 bad rate

4 custom.
2 good rate 2 bad rate

3 custom.
3 bad rate 0 good rate

3 custom.
3 good rate 0 bad rate

2 custom.
0 bad rate 2 good rate

2 custom.
2 bad rate 0 good rate
Credit rating: pruned decision tree

Income > 2000 ?

- yes
  - good
    - 3 custom.
      - 3 good rate
      - 0 bad rate
  - good
    - 4 custom.
      - 2 good rate
      - 2 bad rate

- no
  - no state.
  - bad
    - 3 custom.
      - 3 bad rate
      - 0 good rate

Perhaps due to background knowledge of credit officer
• Clementine Demo

Credit_toy2.str
• Clementine Demo

German-credit1.str
## Example: Computer buyers

Table 1

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Age</th>
<th>Income</th>
<th>Student?</th>
<th>Credit Rating</th>
<th>Buys Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>middle aged</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>senior</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>middle aged</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>youth</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>youth</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>senior</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>youth</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>middle aged</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>middle aged</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>senior</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>
Attribute selection measure in Decision Trees

Information Gain (IG)

IG is based on information theory due to Shannon

Example 1:
Finding a certain number between 1 and 1000 by asking question

1. Alternative

Choose randomly a number between 1 and 1000 and ask whether it is the right one. No optimal method, because in the worst case 999 questions are needed to find the right number.

In this case, if the first answer is “no”, the IG has been very little. Because, there are still 999 alternative numbers between them the number we are looking for is.
Attribute selection measure in Decision Trees

**information Gain (IG)** (continues)

**Second alternative**

The first question should be: is the number $\leq 500$?

The IG of this question is too higher because after the answer we have to search between 500 numbers instead of 1000.

If the answer of this question is positive, the next question is, as it may be expected: Is the number $\leq 250$? and so on.

In this example, IG of each new question is equal to the amount of information one gains by asking this question.

*Higher the IG of a question (attribute) $\rightarrow$ quicker to reach the goal.*
Attribute selection measure in Decision Trees

Information Gain (IG) (continues)

Definition of Entropy

\[ Y: \text{a random variable and } P(Y=b_1)=p_1, \ldots, P(Y=b_m)=p_m \]

Entropy of \( Y \):

\[ I(Y) = - \sum_{i=1}^{m} p_i \log p_i \]  \hspace{1cm} (1)

\( I(Y) \) is the expected information needed to find a certain value in distribution of \( Y \)
Attribute selection measure in Decision Trees

Information Gain (IG) (continues)

Remark 1

Definition (1) is due to Shannon in conjunction with information theory and aims to find the number of needed bits to communicate a message (for this reason the base of used logarithm is 2)

Remark 2

( In example for \( m=1000 \) we get \( \pi = 1/1000 \) ) and \( I(Y) \) in (1) is equal nearly to 10 which is the average number of the question that one needs to find a certain number between 1 and 1000
### Example 2: Computer buyers

- Two classes: Buys Computer (yes or no)
- C1: class 1 (yes), C2: class 2 (no)
- N1: Numbers of tuples in C1 = 9
- N2: Numbers of tuples in C2 = 5
- N = N1 + N2 = 14
- p1: probability that a tuple belongs to C1
- p2: probability that a tuple belongs to C2
- Probability should be approximated by the portions
- Thus: $p1 = \frac{N1}{N} = \frac{9}{14}$ and $p2 = \frac{N2}{N} = \frac{5}{14}$

Using relation (1) results to

$$I(Y) = - \frac{9}{14} \log_2 \left( \frac{9}{14} \right) - \frac{5}{14} \log_2 \left( \frac{5}{14} \right) = 0.94$$

This is the Expected information (entropy) needed to classify a tuple.
Attribute selection measure in Decision Trees

Information Gain (IG) (continues)

Conditional Entropy

Example: X Income Y Football Fan

Using relation (1) results to:
I(X) = 1.5 and I(Y) = 1
Moreover:
P (Football Fan = yes) = 0.5
P (Income= high) = 0.5
P (Income high and Football Fan = no) = 0.25
P (Football Fan = yes | Income = medium ) = 0

Using relation (1) results to:
I(X) = 1.5 and I(Y) = 1
Moreover:
P (Football Fan = yes) = 0.5
P (Income= high) = 0.5
P (Income high and Football Fan = no) = 0.25
P (Football Fan = yes | Income = medium ) = 0

Entropy of Y for X = b:  \( I( Y \mid X = b ) \)

Using this definition and (1) leads to:
\[
\begin{align*}
I ( Y \mid X = \text{high} ) &= 1 \\
I ( Y \mid X = \text{medium} ) &= 0 \\
I ( Y \mid X = \text{low} ) &= 0
\end{align*}
\]

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>no</td>
</tr>
<tr>
<td>low</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>low</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>no</td>
</tr>
<tr>
<td>high</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 2: Football Fan
Attribute selection measure in Decision Trees

**information Gain (IG)** (continues)

**Conditional Entropy**

Generally:

\[
I(Y|X) = \sum_i p(X = b_i) I(Y | X = b_i)
\]

(3) Called average conditional entropy

From Table 2 and relation (2) we can get:

And from this table:

\[
I(Y|X) = 0.5 \times 1 + 0.25 \times 0 + 0.25 \times 0 = 0.5
\]

we have seen already \(I(Y) = 1\) and now by using the values of \(X\) we have got \(I(Y|X) = 0.5\) the needed information reduced to half and we have got \(1 - 0.5 = 0.5\) “Information Gain”
Attribute selection measure in Decision Trees

**Information Gain (IG)** (continues)

Generally:

\[ IG(Y|X) = I(Y) - I(Y|X) \quad (4) \]

(4) called information gained by using X

Inserting (3) in (4) leads to:

\[ IG(Y|X) = I(Y) - \sum p_i (X = bi) I(Y | X = bi) \quad (5) \]

In the relation (5), like before \( p(x=bi) \) can be approximated by \( N_i/N \), where \( N_i \) is the frequency of the value \( x_i \) in \( X \).

(5) Is one of the measures that has been used for attribute selection in Decision trees. The Decision Tree algorithms ID3 e.g. uses this measure.
Construction of Decision Trees

Root selection: using the attribute with highest information gain

Example: Computer Buyers

In the following we show the target variable (computer Buyers) with Y and the attributes (Age, Income..) with X

Now we calculate $\text{IG}(Y)$ regarding attribute age

$$\text{IG}(Y) = \text{I}(Y) - \text{I}(Y \mid \text{age}) \quad (6)$$

with

$$\text{I}(Y \mid \text{age}) = p(\text{youth}) \times \text{I}(Y \mid \text{age} = \text{youth}) + p(\text{middle aged}) \times \text{I}(Y \mid \text{age} = \text{middle aged}) + p(\text{senior}) \times \text{I}(Y \mid \text{age} = \text{senior})$$

we have seen already that $\text{I}(Y) = 0.94$ and for Attribute age we have

$p(\text{youth}) = 5/14$, $p(\text{senior}) = 5/14$ and $p(\text{middle aged}) = 4/14$
On the other hand:

\[
p(Y=\text{no} \mid \text{age=young}) = \frac{3}{5} \\
p(Y=\text{yes} \mid \text{age=young}) = \frac{2}{5}
\]

It means.

\[
I(Y \mid \text{age=young}) = -\frac{3}{5} \log \left( \frac{3}{5} \right) - \frac{2}{5} \log \left( \frac{2}{5} \right) = 0.968
\]

From (7) and (8) we get:

\[
P(\text{young}) \cdot I(Y \mid \text{age=young}) = \frac{5}{14} \times 0.968 = 0.346
\]

In the same way we can calculate the other components of (6):

IG (Y) = I(Y) – I(Y \mid \text{age}) = 0.246

and for the other attributes:

IG (Y) = 0.029

IG (Y) = 0.151

IG (Y) = 0.048

IG of the attribute age is at the highest; Splitting of the DT starts by using this attribute as the root and its values (senior, middle aged, and youth) as the first branches of the tree.
Construction of Decision Trees

Fig 1, first splitting of DC

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Age</th>
<th>Income</th>
<th>Student?</th>
<th>Credit Rating</th>
<th>Buys Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>middle aged</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>senior</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>middle aged</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>youth</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>youth</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>senior</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>youth</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>middle aged</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>middle aged</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>senior</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>
### Construction of Decision Trees

#### Splitting (continues)

**Table:**

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Age</th>
<th>Income</th>
<th>Student?</th>
<th>Credit Rating</th>
<th>Buys Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>youth</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>middle aged</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>senior</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>senior</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>middle aged</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>youth</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>youth</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>senior</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>youth</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>middle aged</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>middle aged</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>senior</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

**Diagram:**

- **Age:** youth, middle aged, senior
- **Student:** no, yes
- **Credit Rating:** fair, excellent, no
- **Buyer:** no, yes

**Notice:** Attribute income wasn't use
Decision Trees as classifiers for new tuples

1. Age 37 = buyer
2. Age 55 with excellent credit rating = no buyer
3. Age 18 but no student = no buyer
4. .....
Construction of Decision Trees

Splitting of continuous-valued attributes

Example: monthly income of 10 individuals

<table>
<thead>
<tr>
<th>Income</th>
<th>700</th>
<th>300</th>
<th>200</th>
<th>200</th>
<th>300</th>
<th>200</th>
<th>500</th>
<th>300</th>
<th>700</th>
<th>500</th>
</tr>
</thead>
</table>

**Step 1:** Sort the attribute values increasing and calculate the average of each two neighbors as possible threshold

<table>
<thead>
<tr>
<th>Attribute value</th>
<th>200</th>
<th>300</th>
<th>500</th>
<th>700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>250</td>
<td>400</td>
<td>600</td>
<td></td>
</tr>
</tbody>
</table>

**Step 2:** Split the node using the averages as alternative thresholds

**Step 3:** Calculate for each split the IG (income) and choose the one with highest IG

```
income

250 > 250
3 Persons 7 Persons

income

400 > 400
6 Persons 4 Persons

income

600 > 600
8 Persons 2 Persons
```
For this case  \( I(Y | A) = \sum 1/N \log (Y I ai ) = 0; \) Regarding: \( IG(A) = I(Y) - I(Y|A) \)

\( IG \) has a significant drawback: it does not take into account the number of attribute values.

Suppose that we have just \( N \) tuples and between the attributes we have a discrete-valued attribute \( A \) with values \( a1, a2, ... ai, .. aj, .. aN \) with \( ai \neq aj \) for all \( i \) and \( j \). In this case we would have by splitting using \( A \), so many partitions as tuples namely \( N \):

\[
\begin{array}{c}
A \\
\vdots \\
a1 \\
a2 \\
ai \\
aj \\
aN
\end{array}
\]

For this case  \( I(Y | A) = - \sum\limits_{i} 1/N \log (Y I ai ) = 0; \) Regarding: \( IG(A) = I(Y) - I(Y|A) \)

This extreme case shows very well that the IG prefers selection attributes with a large number of partitions.
Attribute selection measure in Decision Trees

To overcome this problem Quinlan suggests for C4.5 (extension of ID3 algorithm) using Gain Ratio instead of Information Gain.

Gain Ratio is defined as: \( GR(A) = \frac{IG(A)}{I(A)} \)

\( I(A) = - \sum_{i} \frac{n_i}{N} \cdot \log \left( \frac{n_i}{N} \right) \)

\( n_i/N \) is the portion of the tuples with attribute value \( a_i \)

\[ n_1/N = 4/14 \]
\[ n_2/N = 6/14 \]
\[ n_3/N = 4/14 \]

\[ I(A) = - \frac{4}{14} \cdot \log \frac{4}{14} - \frac{6}{14} \cdot \log \frac{6}{14} - \frac{4}{14} \cdot \log \frac{4}{14} = 0.926 \]

we had calculated \( IG(\text{income}) = 0.029 \) thus: \( GR(A) = 0.029/0.926 = 0.031 \)
Attribute selection measure in Decision Trees

Gini Index (Gini)

Impurity in the nodes

GINI Index: Measure of Impurity

Class 1: 5  
Class 2: 5
High impurity

Class 1: 9  
Class 2: 1
Low impurity

Attribute value
Attribute selection measure in Decision Trees

**Gini Index (Gini)**

\[ Gini(k) = 1 - \sum_{j} \left( \frac{n_j}{n|k} \right)^2 \]

- \( n \) = Number of tuples at the node \( k \)
- \( n_j \) = Number of the tuples belong to the class \( j \) at the node \( k \)
- \( \frac{n_j}{n} \) = relative frequently of the class \( j \) at the node \( k \)

For two classes:

- \( n_1 = 0 \), \( n_2 = n \) \( \rightarrow \) Gini = 0 \( \rightarrow \) lowest impurity
- \( \frac{n_1}{n} = \frac{1}{2} \), \( \frac{n_2}{n} = \frac{1}{2} \) \( \rightarrow \) Gini = 1/2 \( \rightarrow \) highest impurity

Further examples:

- \( \frac{n_1}{n} = \frac{1}{6} \), \( \frac{n_2}{n} = \frac{5}{6} \) \( \rightarrow \) Gini = \( 1 - \left( \frac{1}{6} \right)^2 - \left( \frac{5}{6} \right)^2 \) = 0.278
Attribute selection measure in Decision Trees

**Gini Index**

Gini Index as attribute selection measure

Parent node p

\( N \): Number of the tuples at p

\[ \begin{align*}
\text{Gini (A)} &= \sum_{i=1}^{n} \left( \frac{N_i}{N} \right) \text{Gini}(k_i) \\
\text{Splitting will be done for the attribute with minimal GINI} \\
\text{Some of DT algorithms among them CART use Gini Index}
\end{align*} \]

\text{Attribute A}

\( a_1 \)

\( a_i \)

\( a_n \)

child \( k_1 \)

child \( k_i \)

child \( k_n \)

\( N_1 \): Number of the tuples at \( k_1 \)

\( N_i \): Number of the tuples at \( k_i \)

\( N_n \): Number of the tuples at \( k_n \)
Attribute selection measure in Decision Trees

Gini Index

Example

Parent node p
N : Number of the tuples 100

Attribute A

N₁ = 40
C₁ = 20, C₂ = 20

N₂ = 60
C₁ = 10 C₂ = 50

Gini (A) = \sum_{i} (N_i/N) \text{Gini}(k_i)

Gini (A) = 0.4 * 0.5 + 0.6 * 0.278 = 0.367

\text{Gini } (k) = 1 - \sum_{j} \left( \frac{n_j}{n} \right)^2

\text{Gini (1)} = 1 - \left( \frac{20}{40} \right)^2 - \left( \frac{20}{40} \right)^2 = 0.5

\text{Gini (2)} = 1 - \left( \frac{10}{60} \right)^2 - \left( \frac{50}{60} \right)^2 = 0.278
Construction of Decision Trees

Overfitting

Overfitting means: DT can classify the training data with a relatively high accuracy rate but not the test data. It means the DC is not able to generalize.

Solution: Tree Pruning

- Pre-pruning
- Post-pruning

• Pre-pruning: Stop growing of the tree in the early stages

  Stop Criteria:
  • at pure nodes or nodes with high degree of purity
  • small number of tuples at a node
  • no more improving of accuracy rate by more growing
  …..
Construction of Decision Trees

Overfitting

Post-pruning:

• Produce a full grown tree
• Prune this tree in different depths to produce a set of pruned trees
• Select the best one using a “validation” data set
Weakness and Strength of Decision Trees

**Strength**
- Produce understandable classification rules with reasonable accuracy rates
- Decision trees can be constructed relatively fast
- Decision trees indicate clearly which attributes are most important for classification

**Weakness**
- By using of decision trees only descriptive analysis of data is possible
- Discretization of continuous-valued is necessary
- They are not appropriate for time series analysis and prediction
Part Four: Decision Trees

- Introduction
- Example: Credit Rating
- Example: Computer buyers
- Attribute selection measure in Decision Trees
- Construction of Decision Trees
- Gain Ratio
- Gini Index
- Overfitting
- Pruning