제5장: 규칙기반 시스템과 학습
(인공지능 강의 슬라이드)
(교재) 장병탁, 인공지능 개론, 2017

장 병 탁
서울대학교 컴퓨터공학부
http://bi.snu.ac.kr/~btzhang/
목차

5.1 규칙기반 시스템  .......................... 3
5.2 전문가 시스템  .............................. 12
5.3 의사결정트리  .............................. 22
5.4 결정트리 학습  .............................. 30
5.1 규칙기반 시스템
Rule-Based Systems: Production Rules

• A production rule consists of two parts:
  – condition (antecedent) part
  – conclusion (action, consequent)

• IF (conditions) THEN (actions)

• Example
  IF ((Gauge is OK) AND (Temperature > 120))
  THEN (Cooling system is in the state of overheating)
Human-Readable Format

IF the stain of the organism is gram negative
AND the morphology of the organism is rod
AND the aerobiocity of the organism is gram anaerobic
THEN there is strong evidence (0.8)
that the class of the organism is enterobacteriaceae

MYCIN Format

IF (AND (SAME CNTEXT GRAM GRAMNEG)
             (SAME CNTEXT MORPH ROD)
             (SAME CNTEXT AIR AEROBIC)
THEN (CONCLUDE CNTEXT CLASS ENTEROBACTERIACEAE
             TALLY .8)
Rule Grammar

<rules. ::=  
   <premise> <action>

<premise> ::=  
   ($AND <condition> ... <condition>)

<condition> ::=  
   ($OR <condition> ... <condition>)
   (<predicate> <context> <parameter> <value>) |

<action> ::=  
   <conclusion> | <instruction>
Production System (PS)

• A production system consists of:
  – Working memory (facts memory)
  – Production memory (rules memory)
  – Inference engine, it cycles through three steps:
    • Match facts against rules
    • Select a rule
    • Execute the rule
Production System Cycle

1. Matching facts against rules
2. Select rule(s)
3. Execute the rule(s)

Working memory (list of facts)
Production memory (list of rules)
The user's program
Change rules

Assert, retract, modify facts
The execution cycle of a PS
Agenda
Input/output data; Halt;
Figure 1: Forward-Chaining Procedure
Advantages and Limitations of the Production Systems (PS)

- PS are universal computational mechanism
- PS are universal function approximators
- Readability
- Explanation
- Expressiveness
- Modularity
5.2 전문가 시스템
Early Expert Systems

- DENDRAL – used in chemical mass spectroscopy to identify chemical constituents
  *interprets molecular structure (chemistry expert system)
- MYCIN – Diagnose/remedy bacterial infection (Medical expert system)
- DIPMETER – geological data analysis for oil (Geology Expert System)
- PROSPECTOR – geological data analysis for minerals (Geology Expert System)
- XCON/R1 – configuring computer systems
Expert System Architecture: Modules

A typical ES architecture consists of:

- **Knowledge base module**
- Working memory module (for the current data)
- **Inference engine**
  - Forward chaining (inductive, data driven)
  - Backward chaining (deductive, goal driven)
- User interface (possibly an NLI and menu)
- **Explanation** module
- **Knowledge acquisition** module
Inference Models

• **Forward-chaining** – starts from a set of conditions and moves towards some conclusion

• **Backward-chaining** – starts with a list of goals and the works backwards to see if there is any data that will allow it to conclude any of these goals.

• Both problem-solving methods are built into inference engines or inference procedures
Knowledge-Based System (KBS) Architecture (1/4)
KBS Architecture (2/4)
KBS Architecture (3/4)
KBS Architecture (4/4)
Explanation in Expert System

- `How' and `Why' explanations in ES

Concrete facts:
- temperature = 128°
5.3 의사결정트리
Decision Tree and Rule-Based System
Decision Tree as Rules

R1 • IF Outlook = Sunny
AND Humidity = High,
THEN PlayTennis = No

R2 • IF Outlook = Sunny
AND Humidity = Normal,
THEN PlayTennis = Yes

R3 • IF Outlook = Overcast,
THEN PlayTennis = Yes

R4 • IF Outlook = Rain
AND Wind = Weak,
THEN PlayTennis = Yes

R5 • IF Outlook = Rain
AND Wind = Strong,
THEN PlayTennis = No
Decision Trees

Nodes: attributes
Edges: values
Terminal nodes: class labels
Learning Algorithm: C4.5

\[ \text{Gain}(S, A) \equiv \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \]
Main Idea

• **Classification by Partitioning Example Space**
• **Goal** : Approximating discrete-valued target functions
• **Appropriate Problems**
  – Examples are represented by attribute-value pairs.
  – The target function has discrete output value.
  – Disjunctive description may be required.
  – The training data may contain missing attribute values.
### Example Problem: Play Tennis

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Example Space

Yes
(Outlook = Overcast)

No
(Outlook = Sunny & Humidity = High)

Yes
(Outlook = Sunny & Humidity = Normal)

Yes
(Outlook = Rain & Wind = Weak)

No
(Outlook = Rain & Wind = Strong)
Decision Tree Representation

- Outlook
  - Sunny
  - Overcast
  - Rain
    - High
      - NO
    - Normal
      - YES
  - YES
  - NO

5.4 결정트리 학습
Basic Decision Tree Learning

• Which Attribute is Best?
  – Select the attribute that is most useful for classifying examples.
  – Quantitative Measure
    • Information Gain
    • For Attribute $A$, relative to a collection of data $D$
      \[
      \text{Gain}(D, A) \equiv \text{Entropy}(D) - \sum_{v \in \text{Values}(A)} \frac{|D_v|}{|D|} \text{Entropy}(D_v)
      \]
    • Expected Reduction of Entropy
Entropy

**Entropy** is a measure of the uncertainty in a random variable.

The term **Entropy**, usually refers to the **Shannon entropy**, which quantifies the expected value of the information contained in a message.

Given a random variable \( v \) with value \( V_k \), the entropy of \( v \) is defined by

\[
H(v) = - \sum_{k} P(v_k) \log_2 P(v_k)
\]
Entropy

- Impurity of an Arbitrary Collection of Examples

\[ \text{Entropy}(D) \equiv \sum_{i=1}^{c} -p_i \log p_i \]

- Minimum number of bits of information needed to encode the classification of an arbitrary member of \( D \)

Constructing Decision Tree

ID3(Examples, Target attribute, Attributes)
Examples are the training examples. Target attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a Root node for the tree
- If all Examples are positive, Return the single-node tree Root, with label = +
- If all Examples are negative, Return the single-node tree Root, with label = -
- If Attributes is empty, Return the single-node tree Root, with label = most common value of Target attribute in Examples
- Otherwise Begin
  - A ← the attribute from Attributes that best* classifies Examples
  - The decision attribute for Root ← A
  - For each possible value, v_i, of A,
    - Add a new tree branch below Root, corresponding to the test A = v_i
    - Let Examples_{v_i} be the subset of Examples that have value v_i for A
    - If Examples_{v_i} is empty
      - Then below this new branch add a leaf node with label = most common value of Target attribute in Examples
      - Else below this new branch add the subtree ID3(Examples_{v_i}, Target attribute, Attributes – {A})
  - End
- Return Root

* The best attribute is the one with highest information gain, as defined in Equation (3.4).
Example: Play Tennis (1)

- Entropy of $D$

\[
Entropy(D) = Entropy([9+, 5-])
= -\frac{9}{14} \log\left(\frac{9}{14}\right) - \frac{5}{14} \log\left(\frac{5}{14}\right)
= 0.940
\]
Example: Play Tennis (2)

- Attribute Wind
  - $D = [9+,5-]$
  - $D_{weak} = [6+,2-]$
  - $D_{strong} = [3+,3-]$

$$Gain(D,\text{Wind}) = \text{Entropy}(D) - \sum_{v\in\{\text{weak, strong}\}} \frac{|D_v|}{|D|} \text{Entropy}(D_v)$$

$$= \text{Entropy}(D) - \frac{8}{14} \text{Entropy}(D_{\text{weak}}) - \frac{6}{14} \text{Entropy}(D_{\text{strong}})$$

$$= 0.940 - \frac{8}{14} 0.811 - \frac{6}{14} 1.00$$

$$= 0.048$$

Example: Play Tennis (3)

• **Attribute Humidity**
  - \( D_{\text{high}} = [3+,4-] \)
  - \( D_{\text{normal}} = [6+,1-] \)

\[
\text{Gain}(D, \text{Wind}) = \text{Entropy}(D) - \sum_{v \in \{ \text{high, normal} \}} \frac{|D_v|}{|D|} \text{Entropy}(D_v)
\]

\[
= \text{Entropy}(D) - \frac{7}{14} \text{Entropy}(D_{\text{high}}) - \frac{7}{14} \text{Entropy}(D_{\text{normal}})
\]

\[
= 0.940 - \frac{7}{14} 0.985 - \frac{7}{14} 0.592
\]

\[
= 0.151
\]

\( [9+,5-] : E = 0.940 \)

\( [3+,4-] : E = 0.985 \)

\( [6+,1-] : E = 0.592 \)

Example: Play Tennis (4)

- Best Attribute?
  - $\text{Gain}(D, \text{Outlook}) = 0.246$
  - $\text{Gain}(D, \text{Humidity}) = 0.151$
  - $\text{Gain}(D, \text{Wind}) = 0.048$
  - $\text{Gain}(D, \text{Temperature}) = 0.029$

\[ [9+,5-] : E = 0.940 \]

\[ [2+,3-] : \{D1, D2, D8, D9, D11\} \]
\[ [4+,0-] : \{D3, D7, D12, D13\} \]
\[ [3+,2-] : \{D4, D5, D6, D10, D14\} \]

Example: Play Tennis (5)

- Entropy $D_{\text{sunny}}$

\[
\text{Entropy}(D_{\text{sunny}}) = \text{Entropy}([2+, 3-])
\]
\[
= -\frac{2}{5} \log \left( \frac{2}{5} \right) - \frac{3}{5} \log \left( \frac{3}{5} \right)
\]
\[
= 0.971
\]
Example: Play Tennis (6)

- **Attribute Wind**
  - $D_{weak} = [1+, 2-]
  - $D_{strong} = [1+, 1-]

$Gain(D, Wind) = Entropy(D_{sunny}) - \sum_{v \in \{weak, strong\}} \frac{|D_v|}{|D|} Entropy(D_v)

\[
= Entropy(D_{sunny}) - \frac{3}{5} Entropy(D_{weak}) - \frac{2}{5} Entropy(D_{strong})
\]

\[
= 0.971 - \frac{3}{5} 0.918 - \frac{2}{5} 1.00
\]

\[
= 0.020
\]
Example: Play Tennis (7)

- **Attribute** *Humidity*
  - $D_{\text{high}} = [0+,3-]$
  - $D_{\text{normal}} = [2+,0-]$

\[
\text{Gain}(D, \text{Wind}) = \text{Entropy}(D_{\text{sunny}}) - \sum_{v \in \{\text{high,normal}\}} \frac{|D_v|}{|D|} \text{Entropy}(D_v)
\]

\[
= \text{Entropy}(D_{\text{sunny}}) - \frac{3}{5} \text{Entropy}(D_{\text{high}}) - \frac{2}{5} \text{Entropy}(D_{\text{normal}})
\]

\[
= 0.971 - \frac{3}{5} 0.00 - \frac{2}{5} 0.00
\]

\[
= 0.971
\]

[2+,3-] : $E = 0.971$

[0+,3-] : $E = 0.00$

[2+,0-] : $E = 0.00$

Example: Play Tennis (8)

• Best Attribute?
  – $\text{Gain}(D_{\text{sunny}}, \text{Humidity}) = 0.971$
  – $\text{Gain}(D_{\text{sunny}}, \text{Wind}) = 0.020$
  – $\text{Gain}(D_{\text{sunny}}, \text{Temperature}) = 0.571$

[9+, 5-] : $E = 0.940$

Example: Play Tennis (9)

• Entropy $D_{\text{rain}}$

$\text{Entropy}(D_{\text{sunny}}) = \text{Entropy}([3+,2-])$

$$=-\frac{3}{5}\log\left(\frac{3}{5}\right) - \frac{2}{5}\log\left(\frac{2}{5}\right)$$

$$= 0.971$$

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

Example: Play Tennis (10)

- **Attribute** \textit{Wind}
  - \( D_{\text{weak}} = [3+,0-] \)
  - \( D_{\text{strong}} = [0+,2-] \)

\[
\text{Gain}(D,\text{Wind}) = \text{Entropy}(D_{\text{rain}}) - \sum_{v \in \{\text{weak}, \text{strong}\}} \frac{|D_v|}{|D|} \text{Entropy}(D_v)
\]

\[
= \text{Entropy}(D_{\text{rain}}) - \frac{3}{5} \text{Entropy}(D_{\text{weak}}) - \frac{2}{5} \text{Entropy}(D_{\text{strong}})
\]

\[
= 0.971 - \frac{3}{5} \times 0.00 - \frac{2}{5} \times 1.00
\]

\[
= 0.971
\]

\( \copyright 2007, \text{SNU Biointelligence Lab, http://bi.snu.ac.kr/} \)
Example: Play Tennis (11)

- **Attribute** *Humidity*
  
  \[ D_{\text{high}} = [1+,1-] \]
  
  \[ D_{\text{normal}} = [2+,1-] \]

\[
\text{Gain}(D,\text{Wind}) = \text{Entropy}(D_{\text{rain}}) - \sum_{v \in \{\text{high, normal}\}} \frac{|D_v|}{|D|} \text{Entropy}(D_v)
\]

\[
= \text{Entropy}(D_{\text{rain}}) - \frac{3}{5} \text{Entropy}(D_{\text{high}}) - \frac{2}{5} \text{Entropy}(D_{\text{normal}})
\]

\[
= 0.971 - \frac{2}{5} \times 1.00 - \frac{3}{5} \times 0.918
\]

\[
= 0.020
\]

Example: Play Tennis (12)

- Best Attribute?
  - $Gain(D_{\text{rain}}, \text{Humidity}) = 0.020$
  - $Gain(D_{\text{rain}}, \text{Wind}) = 0.971$
  - $Gain(D_{\text{rain}}, \text{Temperature}) = 0.020$

Avoiding Overfitting Data

• Definition
  – Given a hypothesis space $H$, a hypothesis $h$ is said to **overfit** the data if there exists some alternative hypothesis $h'$ such that $h$ has smaller error than $h'$ over the training examples, but $h'$ has a smaller error than $h$ over entire distribution of instances.

• Occam’s Razor
  – Prefer the simplest hypothesis that fits the data.
Avoiding Overfitting Data

Solutions to Overfitting

1. Partition examples into training, test, and validation set.

2. Use all data for training, but apply a statistical test to estimate whether expanding (or pruning) a particular node is likely to produce an improvement beyond the training set.

3. Use an explicit measure of the complexity for encoding the training examples and the decision tree, halting growth of the tree when this encoding is minimized.
Summary

• Decision trees are a tree representation of production rules. A decision tree is a collection of production rules.
• Thus, decision tree learning is a rule induction process and can be used for automatic knowledge acquisition for rule-based expert systems.

• Formally, decision trees provide a practical method for concept learning and discrete-valued functions in tree structure.
• ID3 searches a complete hypothesis space of the decision trees or rule sets.
• Overfitting is an important issue in decision tree learning.