Expert Systems

- Knowledge representation
- Knowledge acquisition
- Machine learning
- ID3 & C4.5
Knowledge Representation

Recall:

- Knowledge engineering
  - Knowledge acquisition
    - Knowledge elicitation
  - Knowledge representation
    - Production rules
    - Semantic networks
    - Frames
Knowledge Representation

- Representation is more than just encoding (encrypting)
- Coding preserves structural ambiguity
- Communication assumes prior knowledge
- Representation implies organization
Knowledge Representation

- Representation
  - A set of syntactic and semantic conventions that make it possible to describe things (Winston)

- Description
  - makes use of the conventions of a representation to describe some particular thing

- Syntax v. semantics
Knowledge Representation

- STRIPS
  - Predicate-argument expressions
    - at (robot, roomA)
  - World models
  - Operator tables
    - push (X, Y, Z)
      - Preconditions: at (robot, Y), at (X, Y)
      - Delete list: at (robot, Y), at (X, Y)
      - Add list: at (robot, Z), at (X, Z)
Knowledge Representation

- STRIPS
  - maintained lists of goals
  - selected goal to work on next
  - searched for applicable operators
  - matched goals against formulas in add lists
  - set up preconditions as sub-goals
  - used means-end analysis
Knowledge Representation

- STRIPS - lessons
  - Heuristic search
  - Uniform representation
  - Problem reduction

- Procedural semantics
Knowledge Representation

- **MYCIN**
  - Assists physicians who are not experts in the field of antibiotics in treating blood infections
  - Consists of
    - Knowledge base
    - Dynamic patient database
    - Consultation program
    - Explanation program
    - Knowledge acquisition program
Knowledge Representation

MYCIN

- Production rules
  - Premises
    - Conjunctions of conditions
  - Actions
    - Conclusions or instructions
- Patient information stored in context tree
- Certainty factors for uncertain reasoning
- Backward chaining control structure (based on AND/OR tree)
Knowledge Representation

- MYCIN
  - Evaluation
    - Panel of experts approved 72% of recommendations
    - Good as experts
    - Better than non-experts
    - Knowledge base incomplete (400 rules)
    - Required more computing power than available in hospitals
    - Doctors did not like the user interface
Knowledge Acquisition

- Stages
  - Identification
  - Conceptualization
  - Formalization
  - Implementation
  - Testing

- KADS

- Ontological analysis
Knowledge Acquisition

- Expert system shells
  - EMYCIN
  - TEIRESIAS
  - Rule models (meta-rules)
  - Schemas for data types
  - Domain-specific knowledge
  - Representation-specific knowledge
  - Representation-independent knowledge
  - Explain-Test-Review
Knowledge Acquisition

Methods and tools

- Structured interview
- Unstructured interview
- Case studies
  - Retrospective v. observational
  - Familiar v. unfamiliar
- Concurrent protocols
  - Verbalization, “thinking aloud”
- Tape recording
- Video recording
Knowledge Acquisition

Methods and tools

- Automated knowledge acquisition
  - Domain models
  - Graphical interfaces
  - Visual programming language
Knowledge Acquisition

Different types of knowledge

- Procedural knowledge
  - Rules, strategies, agendas, procedures

- Declarative knowledge
  - Concepts, objects, facts

- Meta-knowledge
  - Knowledge about other types of knowledge and how to use them

- Structural knowledge
  - Rules sets, concept relationships, concept to object relationships
Knowledge Acquisition

Sources of knowledge

- Experts
- End-users
- Multiple experts (panels)
- Reports
- Books
- Regulations
- Guidelines
Knowledge Acquisition

- Major difficulties with elicitation
  - Expert may
    - be unaware of the knowledge used
    - be unable to verbalize the knowledge used
    - provide irrelevant knowledge
    - provide incomplete knowledge
    - provide incorrect knowledge
    - provide inconsistent knowledge
Knowledge Acquisition

“The more competent domain experts become, the less able they are to describe the knowledge they used to solve problems”
(Waterman)
Detailed guidelines for conducting structured and unstructured interviews and both retrospective and observational case studies are given in Durkin (Chapter 17)
# Knowledge Acquisition

## Technique Capabilities

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Interviews</th>
<th>Case Studies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Unstructured</strong></td>
<td><strong>Structured</strong></td>
</tr>
<tr>
<td>Facts</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td>Concepts</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>Objects</td>
<td>Good</td>
<td>Excellent</td>
</tr>
<tr>
<td>Rules</td>
<td>Fair</td>
<td>Average</td>
</tr>
<tr>
<td>Strategies</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>Heuristics</td>
<td>Fair</td>
<td>Average</td>
</tr>
<tr>
<td>Structures</td>
<td>Fair</td>
<td>Excellent</td>
</tr>
</tbody>
</table>
Knowledge Acquisition

Analyzing the knowledge collected

- Producing transcripts
- Interpreting transcripts
  - Chunking
- Analyzing transcripts
  - Knowledge dictionaries
  - Graphical techniques
    - Cognitive maps
    - Inference networks
    - Flowcharts
    - Decision trees
Machine Learning

- Rote learning
- Supervised learning
  - Induction
    - Concept learning
    - Descriptive generalization
- Unsupervised learning
Machine Learning

- META-DENDRAL
  - RULEMOD
    - Removing redundancy
    - Merging rules
    - Making rules more specific
    - Making rules more general
    - Selecting final rules
Machine Learning

- META-DENDRAL
  - Version spaces
    - Partial ordering
    - Boundary sets
    - Candidate elimination algorithm
  - Monotonic, non-heuristic
  - Results independent of order of presentation
  - Each training instance examine only once
  - Discarded hypotheses never reconsidered
  - Learning is properly incremental
Decision trees and production rules

* Decision trees are an alternative way of structuring rules
* Efficient algorithms exist for constructing decision trees
* There is a whole family of such learning systems:
  - CLS (1966)
  - ID3 (1979)
  - ACLS (1982)
  - ASSISTANT (1984)
  - IND (1990)
  - C4.5 (1993) - and C5.0
* Decision trees can be converted to rules later
Let $X$ be a variable with states $x_1 - - - x_n$

Define the entropy of $X$ by

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2(p(x_i))$$

N.B. $\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)} = \frac{\ln(x)}{\ln(2)}$
Entropy

Consider flipping a perfect coin:

e.g., \( n = 2 \)

\[
X : x_1, x_2
\]

\[
p(x_1) = p(x_2) = 1/2
\]
Entropy

\[ H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 (p(x_i)) \]

\[ = -\left[ \frac{1}{2} \log_2 \left( \frac{1}{2} \right) + \frac{1}{2} \log_2 \left( \frac{1}{2} \right) \right] \]

\[ = -\left[ \frac{1}{2} (-1) + \frac{1}{2} (-1) \right] = 1 \]
Entropy

Consider $n$ equiprobable outcomes

\[ H(X) = - \sum_{i=1}^{n} p(x_i) \log_2 (p(x_i)) \]

\[ = - \sum_{i=1}^{n} \left( \frac{1}{n} \right) \log_2 \left( \frac{1}{n} \right) \]

\[ = \sum_{i=1}^{n} \frac{1}{n} \log_2 (n) = \log_2 (n) \]
Entropy

- Consider flipping a totally biased coin:
  - e.g., \( n = 2 \)
  
  \( X : x_1, x_2 \)
  
  \( p(x_1) = 1 \) \hspace{1cm} \( p(x_2) = 0 \)
Entropy

\[ H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 (p(x_i)) \]

\[ = -\left[ \log_2 (1) + 0 \log_2 (0) \right] \]

\[ = -\left[ 0 + 0 \log_2 (0) \right] = 0 \]

(by L’Hôpital’s rule)
Entropy is a measure of chaos or disorder.

$H(X)$ is maximum for equiprobable outcomes.
Entropy

\( X: x_1 \ldots x_m \) and \( Y: y_1 \ldots y_n \) be two variables

\[
H(X, Y) = -\sum_{i=1}^{m} \sum_{j=1}^{n} p(x_i, y_j) \log_2 (p(x_i, y_j))
\]

* If \( X \) and \( Y \) are independent

\[
H(X, Y) = H(X) + H(Y)
\]
Conditional Entropy

* Partial conditional entropy of $Y$ given $X$ is in state $x_i$:

$$H(Y|x_i) = - \sum_{j=1}^{n} p(y_j|x_i) \log_2 (p(y_j|x_i))$$

* Full conditional entropy of $Y$ given $X$:

$$H(Y|X) = \sum_{i=1}^{m} p(x_i) \cdot H(Y|x_i)$$
# Machine Learning

## Binary Logarithms

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>1.5850</td>
</tr>
<tr>
<td>4</td>
<td>2.0000</td>
</tr>
<tr>
<td>5</td>
<td>2.3219</td>
</tr>
<tr>
<td>6</td>
<td>2.5850</td>
</tr>
<tr>
<td>7</td>
<td>2.8074</td>
</tr>
<tr>
<td>8</td>
<td>3.0000</td>
</tr>
</tbody>
</table>
Machine Learning

- **ID3**

  - Builds a decision tree first, then rules
  - Given a set of attributes, and a decision, recursively selects attributes to be the root of the tree based on Information Gain:

    \[ H(\text{decision}) - H(\text{decision} | \text{attribute}) \]

  - Favors attributes with many outcomes
  - Is not guaranteed to find the simplest decision tree
  - Is not incremental
C4.5

* Selects attributes based on Information gain ratio:

\[
\frac{H(\text{decision}) - H(\text{decision} \mid \text{attribute})}{H(\text{attribute})}
\]

* Uses pruning heuristics to simplify decision trees
  - to simplify
  - to reduce dependence on training set

* Tunes the resulting rule(s)
C4.5 rule tuning

- Derive initial rules by enumerating paths through the decision tree
- Generalize the rules by possibly deleting unnecessary conditions
- Group rules according to target classes and delete any that do not contribute to overall performance on the class
- Order the sets of rules for the target classes and choose a default class
Rule tuning

- Rule tuning may be useful for rules derived by a variety of other means besides C4.5
  - Evaluate the contribution of individual rules
  - Evaluate the performance of the rule set as a whole
### A data set for classification (Quinlan)

<table>
<thead>
<tr>
<th>Height</th>
<th>Hair</th>
<th>Eyes</th>
<th>Attractiveness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>short</td>
<td>blond</td>
<td>blue</td>
</tr>
<tr>
<td>2</td>
<td>tall</td>
<td>blond</td>
<td>brown</td>
</tr>
<tr>
<td>3</td>
<td>tall</td>
<td>red</td>
<td>blue</td>
</tr>
<tr>
<td>4</td>
<td>short</td>
<td>dark</td>
<td>blue</td>
</tr>
<tr>
<td>5</td>
<td>tall</td>
<td>dark</td>
<td>blue</td>
</tr>
<tr>
<td>6</td>
<td>tall</td>
<td>blond</td>
<td>blue</td>
</tr>
<tr>
<td>7</td>
<td>tall</td>
<td>dark</td>
<td>brown</td>
</tr>
<tr>
<td>8</td>
<td>short</td>
<td>blond</td>
<td>brown</td>
</tr>
</tbody>
</table>
A data set for classification (Quinlan)

\[ H(\text{decision}) = H(\text{Attractiveness}) \]

\[ = - \frac{3}{8} \log_2 \left( \frac{3}{8} \right) - \frac{5}{8} \log_2 \left( \frac{5}{8} \right) = 0.955 \]
A data set for classification (Quinlan)

- **Height:**
  - short: 1, 4, 8 \( p(\text{+}|\text{short}) = 1/3 \) \( p(-|\text{short}) = 2/3 \)
  - tall: 2, 3, 5, 6, 7 \( p(\text{+}|\text{tall}) = 2/5 \) \( p(-|\text{tall}) = 3/5 \)

\[
H(\text{decision}|\text{attribute}) = H(\text{Attractiveness}|\text{Height}) = \\
\frac{3}{8} \left[ -\frac{1}{3} \log_2 \left( \frac{1}{3} \right) - \frac{2}{3} \log_2 \left( \frac{2}{3} \right) \right] + \frac{5}{8} \left[ -\frac{2}{5} \log_2 \left( \frac{2}{5} \right) - \frac{3}{5} \log_2 \left( \frac{3}{5} \right) \right] = 0.951
\]

- **Information gain** = \( 0.955 - 0.951 = 0.004 \)
A data set for classification (Quinlan)

- **Hair:**
  - blond: 1, 2, 6, 8 \( p(+)\text{|blond} = \frac{2}{4} \) \( p(-)\text{|blond} = \frac{2}{4} \)
  - red: 3 \( p(+)\text{|red} = \frac{1}{1} \) \( p(-)\text{|red} = \frac{0}{1} \)
  - dark: 4, 5, 7 \( p(+)\text{|dark} = \frac{0}{3} \) \( p(-)\text{|dark} = \frac{3}{3} \)

- **H(decision\text{|attribute}) = H(Attractiveness\text{|Hair}) =**

\[
\frac{4}{8}[1] + \frac{1}{8}[0] + \frac{3}{8}[0] = 0.500
\]

- **Information gain =** \( 0.955 - 0.500 = 0.455 \)
A data set for classification (Quinlan)

* Eyes:
  - blue: 1, 3, 4, 5, 6  \( p(+) = 3/5 \)  \( p(-) = 2/5 \)
  - brown: 2, 7, 8  \( p(+) = 0/3 \)  \( p(-) = 3/3 \)

* H(decision|attribute) = H(Attractiveness|Eyes) =

\[
\frac{5}{8} \left[ \frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) \right] + \frac{3}{8} [0] = 0.607
\]

* Information gain = 0.955 - 0.607 = 0.348
A data set for classification (Quinlan)

- Hence Hair is chosen as the best choice for the root of the tree
- Now we recursively repeat this process for the (three) resulting branches
- In this case, the branches for Hair: red and Hair: dark are already completely classified, and we need to work only on the sub-table for Hair: blond
### A data set for classification (Quinlan)

<table>
<thead>
<tr>
<th>Height</th>
<th>Eyes</th>
<th>Attractiveness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 short</td>
<td>blue</td>
<td>+</td>
</tr>
<tr>
<td>2 tall</td>
<td>brown</td>
<td>-</td>
</tr>
<tr>
<td>6 tall</td>
<td>blue</td>
<td>+</td>
</tr>
<tr>
<td>8 short</td>
<td>brown</td>
<td>-</td>
</tr>
</tbody>
</table>

\[
\star H(\text{decision}) = H(\text{Attractiveness})
\]

\[
\begin{align*}
&= -\frac{2}{4} \log_2 \left( \frac{2}{4} \right) - \frac{2}{4} \log_2 \left( \frac{2}{4} \right) \\
&= 1
\end{align*}
\]
A data set for classification (Quinlan)

- **Height:**
  - short: 1, 8  \( p(+|\text{short}) = 1/2 \quad p(-|\text{short}) = 1/2 \)
  - tall: 2, 6 \( p(+|\text{tall}) = 1/2 \quad p(-|\text{tall}) = 1/2 \)

- **H(\text{decision}|\text{attribute}) = H(\text{Attractiveness}|\text{Height}) =**

\[
\frac{2}{4} [1] + \frac{2}{4} [1] = 1
\]

- **Information gain =** 1-1=0
A data set for classification (Quinlan)

* Eyes:
  - blue: 1, 6 \( p(\text{+|blue}) = 2/2 \) \( p(-|\text{blue}) = 0/2 \)
  - brown: 2, 8 \( p(\text{+|brown}) = 0/2 \) \( p(-|\text{brown}) = 2/2 \)

\[ H(\text{decision|attribute}) = H(\text{Attractiveness|Eyes}) = \]
\[ \frac{2}{4} [0] + \frac{2}{4} [0] = 0 \]

* Information gain = 1-0=1
A data set for classification (Quinlan)

* Hence Eyes is chosen as the best root of this subtree
* The final tree is

```
   Hair
  /  \
blond red dark
   \
Eyes 3
  / \  \
blue brown
  1, 6 2, 8
     + +  - -
     + +  - -
```
A data set for classification (Quinlan)

* We may now build rules from this decision tree
  - R1: (Hair, dark) --> (Attractiveness, -)
  - R2: (Hair, red) --> (Attractiveness, +)
  - R3: (Hair, blond) & (Eyes, blue) --> (Attractiveness, +)
  - R4: (Hair, blond) & (Eyes, brown) --> (Attractiveness, -)

* Note that height is irrelevant
A data set for classification (Quinlan)

- Dropping conditions from rules
  - Rules 1 and 2 have only one condition
  - Rule 3: neither condition can be dropped (case 5 needs the first condition and case 2 needs the second condition)
  - Rule 4: we can drop the first condition
  - R4': (Eyes, brown) --> (Attractiveness, -)
A data set for classification (Quinlan)

Dropping conditions from rules

- Linear
  - Scan rule left to right
  - Try to drop conditions one at a time
  - If possible, drop for good
  - Iterate (n conditions, n attempts)

- Exponential
  - Scan rule left to right
  - Try to drop conditions one at a time
  - Then try to drop pairs, triples, etc. (n conditions, $2^n-2$ attempts)
A data set for classification (Quinlan)

Now consider Information gain ratio

For initial root of tree we already know

\[ H(\text{decision}) = H(\text{Attractiveness}) \]

\[ = -\frac{3}{8} \log_2 \left( \frac{3}{8} \right) - \frac{5}{8} \log_2 \left( \frac{5}{8} \right) = 0.955 \]
A data set for classification (Quinlan)

* $H(\text{decision}|\text{attribute}) = H(\text{Attractiveness}|\text{Height}) = \frac{3}{8} \left[ -\frac{1}{3} \log_2 \left( \frac{1}{3} \right) - \frac{2}{3} \log_2 \left( \frac{2}{3} \right) \right] + \frac{5}{8} \left[ -\frac{2}{5} \log_2 \left( \frac{2}{5} \right) - \frac{3}{5} \log_2 \left( \frac{3}{5} \right) \right] = 0.951$

* $H(\text{decision}|\text{attribute}) = H(\text{Attractiveness}|\text{Hair}) = \frac{4}{8} [1] + \frac{1}{8} [0] + \frac{3}{8} [0] = 0.500$

* $H(\text{decision}|\text{attribute}) = H(\text{Attractiveness}|\text{Eyes}) = \frac{5}{8} \left[ -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) \right] + \frac{3}{8} [0] = 0.607$
Machine Learning

A data set for classification (Quinlan)

\[ H(\text{attribute}) = H(\text{Height}) = \]
\[ -\frac{3}{8} \log_2 \left( \frac{3}{8} \right) - \frac{5}{8} \log_2 \left( \frac{5}{8} \right) = 0.955 \]

\[ H(\text{attribute}) = H(\text{Hair}) = \]
\[ -\frac{4}{8} \log_2 \left( \frac{4}{8} \right) - \frac{1}{8} \log_2 \left( \frac{1}{8} \right) - \frac{3}{8} \log_2 \left( \frac{3}{8} \right) = 1.406 \]

\[ H(\text{attribute}) = H(\text{Eyes}) = \]
\[ -\frac{5}{8} \log_2 \left( \frac{5}{8} \right) - \frac{3}{8} \log_2 \left( \frac{3}{8} \right) = 0.955 \]
A data set for classification (Quinlan)

- Hence the Information gain ratios are
  - Height: 0.004
  - Hair: 0.324
  - Eyes: 0.364

- By this criterion, Eyes is chosen as the best root available

- The branch for Eyes: brown is already completely classified, and we need to work only on the sub-table for Eyes: blue
### Machine Learning

A data set for classification (Quinlan)

<table>
<thead>
<tr>
<th>Attributes</th>
<th>---</th>
<th>Decision</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Height</strong></td>
<td><strong>Hair</strong></td>
<td><strong>Attractiveness</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>short</td>
<td>blond</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>tall</td>
<td>red</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>short</td>
<td>dark</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>tall</td>
<td>dark</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>tall</td>
<td>blond</td>
<td>+</td>
</tr>
</tbody>
</table>

\[
H(\text{decision}) = H(\text{Attractiveness}) = -\frac{3}{5} \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.971
\]
### Machine Learning

#### A data set for classification (Quinlan)

- **Height:**
  - short: 1, 4 \( p(+|\text{short}) = 1/2 \) \( p(-|\text{short}) = 1/2 \)
  - tall: 3, 5, 6 \( p(+|\text{tall}) = 2/3 \) \( p(-|\text{tall}) = 1/3 \)

- \( H(\text{decision}|\text{attribute}) = H(\text{Attractiveness}|\text{Height}) = \)

  \[
  \frac{2}{5} \left[ -\frac{1}{2} \log_2 \left( \frac{1}{2} \right) - \frac{1}{2} \log_2 \left( \frac{1}{2} \right) \right] + \frac{3}{5} \left[ -\frac{2}{3} \log_2 \left( \frac{2}{3} \right) - \frac{1}{3} \log_2 \left( \frac{1}{3} \right) \right] = 0.951
  \]

- \( H(\text{Height}) = \frac{2}{5} \log_2 \left( \frac{2}{5} \right) - \frac{3}{5} \log_2 \left( \frac{3}{5} \right) = 0.971 \)
A data set for classification (Quinlan)

- Hair:
  - blond: 1, 6       \( p(+) \text{blond} = 2/2 \)   \( p(-) \text{blond} = 0/2 \)
  - red: 3           \( p(+) \text{red} = 1/1 \)   \( p(-) \text{red} = 0/1 \)
  - dark: 4,5        \( p(+) \text{dark} = 0/2 \)   \( p(-) \text{dark} = 2/2 \)

- \( H(\text{decision}|\text{attribute}) = H(\text{Attractiveness}|\text{Hair}) = \)
  \[
  \frac{2}{5} [0] + \frac{1}{5} [0] + \frac{2}{5} [0] = 0
  \]

- \( H(\text{Hair}) = -\frac{2}{5} \log_2 \left( \frac{2}{5} \right) - \frac{1}{5} \log_2 \left( \frac{1}{5} \right) - \frac{2}{5} \log_2 \left( \frac{2}{5} \right) = 0.793 \)
A data set for classification (Quinlan)

Hence the Information gain ratios are
- Height: 0.021
- Hair: 1.224

By this criterion, Hair is chosen as the best root available.
A data set for classification (Quinlan)

* We may now build rules from this decision tree
  * R1: (Eyes, brown) --> (Attractiveness, -)
  * R2: (Eyes, blue) & (Hair, blond) --> (Attractiveness, +)
  * R3: (Eyes, blue) & (Hair, red) --> (Attractiveness, +)
  * R4: (Eyes, blue) & (Hair, dark) --> (Attractiveness, -)

* These are different rules

* Note that after dropping conditions, however, they are the same - this is NOT generally true
Machine Learning

ID3 & C4.5

- What if too many cases?
  - Windowing
- What if the data is incomplete?
- What if the data is inconsistent?
- What if the data is continuous?
  - Binarization
  - Discretization
- Incremental algorithms?
- Pruning?