CSE 473
Artificial Intelligence

Review
Logistics

• Problem Set due at midnight

• Exam next Wed 8:30—10:30
  Regular classroom
  Closed book
  Cover all quarter’s material
  Emphasis on material not covered on midterm
  • Even more emphasis on material not on any PS
Abalone
Defining AI

<table>
<thead>
<tr>
<th></th>
<th>human-like</th>
<th>rational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems that think</td>
<td>Systems think like humans</td>
<td>Systems think rationally</td>
</tr>
<tr>
<td>Systems that act</td>
<td>Systems act like humans</td>
<td>Systems act rationally</td>
</tr>
</tbody>
</table>
Goals of this Course

• To introduce you to a set of key:
  Paradigms &
  Techniques
• Teach you to identify when & how to use
  Heuristic search
  Constraint satisfaction
  Machine learning
  Logical inference
  Bayesian inference
  Policy construction
Theme I

• Problem Spaces & Search

How to specify PS?

Two kinds of search?
Learning as Search
SUPPLE – Adapting UIs
Adapting to Device Characteristics

Func Interface Spec + Device Model → Custom Interface Rendering

Hierarchy of State vars + Methods

Screen size
Available widgets
Interaction modes
Interface Adaptation as Search

Func Interface Spec + Device Model → Custom Interface Rendering

© Daniel S. Weld
Theme II

• In the knowledge lies the power

• Adding knowledge to search?
Heuristics

• How to generate?

• Admissibility?
Constraint Satisfaction?

• How to Specify?

• Why Effective?

SEND
+
MORE
------
MONEY
Backjumping (BJ)

• Similar to BT, but more efficient when no consistent instantiation can be found for the current var

• Instead of backtracking to most recent var...
  BJ reverts to deepest var which was c-checked against the current var

BJ Discovers
(2, 5, 3, 6) inconsistent with x₆
No sense trying other values of x₅
Shuttle Repair Scheduling
Probabilistic Representations

• How encode knowledge here?

In the knowledge lies the power
Theme III
Importance of Representation
- Features in ML
- Reformulation

© Daniel S. Weld
## Propositional Logic vs. First Order

<table>
<thead>
<tr>
<th><strong>Ontology</strong></th>
<th>Facts (P, Q)</th>
<th>Objects, Properties, Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syntax</strong></td>
<td>Atomic sentences, Connectives</td>
<td>Variables &amp; quantification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sentences have structure: terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>father-of(mother-of(X)))</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td>Truth Tables</td>
<td>Interpretations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Much more complicated)</td>
</tr>
<tr>
<td><strong>Inference</strong></td>
<td>DPLL, GSAT</td>
<td>Unification</td>
</tr>
<tr>
<td><strong>Algorithm</strong></td>
<td>Fast in practice</td>
<td>Forward, Backward chaining</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prolog, theorem proving</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>NP-Complete</td>
<td>Semi-decidable</td>
</tr>
</tbody>
</table>
Nested Quantifiers: 
Order matters!

$$\forall x \exists y \ P(x,y) \neq \exists y \ \forall x \ P(x,y)$$

Every dog has a tail

$$\forall d \exists t \ \text{has}(d,t) \ ? \ \exists t \ \forall d \ \text{has}(d,t)$$
Logical Inference as Search
Skolemization

• Existential quantifiers aren’t necessary!
  Existential variables can be replaced by
  • Skolem functions (or constants)
  • Args to function are all surrounding $\forall$ vars

• $\forall d \exists t \ has(d, t)$
  $\forall d \ has(d, f(d) )$

• $\exists x \forall y \ loves(y, x)$
  $\forall y \ loves(y, f() )$
  $\forall y \ loves(y, f_{97} )$
Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when PREJUDICE meets DATA!

Learning a “FOO”
Planning

- **Extend to Durative Actions**
  - Simultaneous actions
  - Minimize make-span

- How???
Uncertainty

- Joint Distribution
- Prior & Conditional Probability
- Bayes Rule
- [Conditional] Independence
- Bayes Net

<table>
<thead>
<tr>
<th></th>
<th>B=t</th>
<th>B=f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr(B)</td>
<td>0.05</td>
<td>0.95</td>
</tr>
<tr>
<td>Pr(A</td>
<td>E,B)</td>
<td>e,b</td>
</tr>
<tr>
<td></td>
<td>0.9 (0.1)</td>
<td>0.2 (0.8)</td>
</tr>
<tr>
<td></td>
<td>0.85 (0.15)</td>
<td>0.01 (0.99)</td>
</tr>
</tbody>
</table>
Dynamic Bayesian Network
State Estimation
Specifying a MDP

\[ S = \text{set of states set} \ (|S| = n) \]

\[ A = \text{set of actions} \ (|A| = m) \]

\[ Pr = \text{transition function } Pr(s,a,s') \]
represented by set of \( m \times n \) stochastic matrices (factored into DBNs)
each defines a distribution over \( S \times S \)

\[ R(s) = \text{bounded, real-valued reward function} \]
represented by an \( n \)-vector
Finding a Policy

- Value Iteration
- Policy Iteration
- Modified Policy Iteration
Bellman Backup

\[ Q_{n+1}(s,a) \]

\[ V_{n+1}(s) \rightarrow a_1 \rightarrow V_n \]
\[ V_{n+1}(s) \rightarrow a_2 \rightarrow V_n \]
\[ V_{n+1}(s) \rightarrow a_3 \rightarrow V_n \]
Q-Learning

- Maintain $Q(a, s)$ for visited states, tried actions
- As execute actions, do backup on per-action basis

$$Q(a, s) \leftarrow Q(a, s) + \alpha (R(s) + \gamma \max_{a'} Q(a', s') - Q(a, s))$$

- Or compute weighted average over all future states (not just immediate successor)
- Do updates at end of epoch

- Approximating Q function
And More

• Specific search & CSP algorithms
• Adversary Search
• Inference in Propositional & FO Logic
• Specific Learning Algorithms
  DT Induction, Ensembles, Naïve Bayes
• EM, DBNs
• Lots of details