Modeling and Decision Making

1/24/18
Modeling Dimensions

• Discreteness
• Planning horizon
• Observability
• Uncertainty
• Dynamism
• Number of agents
Discreteness

- Does the agent model the environment as:
  - Discrete
    - Some software agents may truly live in a discrete world.
  - Continuous
    - The real world is continuous, but a discrete model can often improve agent reasoning.
    - Discrete and continuous modules can co-exist, e.g. discrete route planning and continuous motor control.

Temperature is continuous, but a discrete state model simplifies the thermostat.

States:
- Too cold
- Comfortable
- Too hot
Planning Horizon

- non-planning
  - thermostat
- fixed finite horizon
  - tic-tac-toe player
- indefinite finite horizon
  - chess player
- infinite horizon
  - smart home

This difference is an adaptation to computational constraints.

Different components of the same system may operate on different horizons.
Observability

Does the agent know everything about the world that is relevant to its decisions?

Full observability
- Bird’s eye view of a maze
- Chess

Partial observability
- Agent dropped into a maze
- Poker
Uncertainty

When an agent acts, does it know all the consequences of that action?

In deterministic environments

• Agents can make a plan they know will succeed
• Optimize for the simplest or cheapest plan

In uncertain environments

• Agents may need contingent plans
• Agents may need to reason probabilistically
• Consider risk/reward and optimize expected outcomes
Dynamic Environment

• If the world is modeled as static, we assume that the environment only changes as a result of the agent’s actions.

• In a dynamic environment, the world can change on its own.

How is this different from uncertainty/observability?
Number of Agents

Additional agents can be modeled as:

- part of the environment
  - This will always make the environment dynamic.
- competitors
  - The agent will need to reason about their intentions with game theory.
- collaborators
  - The agent may be able to offload some of its physical or computational work on others.
  - The agent may need to assist other agents.
## How should we model these environments?

<table>
<thead>
<tr>
<th></th>
<th>Is the world discrete or continuous?</th>
<th>What is the planning horizon?</th>
<th>Is the environment fully observable?</th>
<th>Do actions have deterministic or uncertain consequences?</th>
<th>Is the environment static or dynamic?</th>
<th>Is the one agent? If there are many, are they cooperative or competitive?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rubick's cube</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mars rover</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stock trading</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Frameworks for Decision-Making

1. Goal-directed planning
   • Agents want to accomplish some goal.
   • The agent will use search to devise a plan.

2. Utility maximization
   • Agents ascribe a utility to various outcomes.
   • The agent attempts to maximize expected utility.
Goal-Directed Planning

Examples:
• Lab 0: maze search
• Lab 1: heuristic search
• Lab 3, 4: game playing

Approach:
• Search for a sequence of actions that achieves a goal.
State space problems have

- A set of discrete **states**
- A distinguished **start state**
- A set of **actions** available to the agent in each state;
- An **action function** that, given a state and an action, returns a new state
- A set of **goal states**, often specified as a function
- A way to measure **solution quality**
Utility Maximization

Examples:
• Lab 2: optimization/local search
• Lab 4, 5: nondeterministic planning

Key ideas:
• Assign utility value to various outcomes
• Reason about the probability of each outcome occurring
• Maximize expected value
Utility considerations

Planning horizon: does the agent get utility at the end, or accumulate it along the way?

• Discounting

\[ EU = \sum_{i=1}^{n} D_i P_i \]

Expected value:

• Act in a way that maximizes the sum over outcomes of the probability of that outcome times its utility.
What are the advantages and disadvantages of each framework for these tasks?

<table>
<thead>
<tr>
<th></th>
<th>Goal-directed planning</th>
<th>Utility maximization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rubick’s cube</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mars rover</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stock trading</td>
<td></td>
<td></td>
</tr>
<tr>
<td>chess playing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>