Decision Making in Intelligent Systems:

Partially observable Markov decision processes

14 April 2008

Frans Oliehoek
Regular MDPs

- Up to now...

State = state of environment!
Regular MDPs

- Up to now...

Agents observe the state (of the environment) -> Fully Observable MDP
Partially observable MDPs

- Now: Partially observable environment
  - agent can't observe the full state.
  - ...but observation gives hint about the true state.
RL vs Planning

- In this course: focus on reinforcement learning (RL).
- RL = learning the model + planning
  - Planning is `using the model'
  - explicit: `model-based RL'
  - implicit: Q-learning etc.

- In this lecture: only planning!
  - We assume we have a perfect model of the (partially observable) world.
Partially observable MDPs

- States – $s_1 \ldots s_n$
- Transitions – $P(s' | s, a)$
- Rewards - $R(s, a)$
- Observations – $o_1 \ldots o_m$
- Observation probs – $P(o | a, s')$
POMDP: an example

- Where am I?
Partial observability

- When is an agent's environment partially observable?
  - Real world: almost always.

- Types of partial observability
  - Noise
    - Sensors have measurement errors.
    - Sensor (or other part of the agent) can fail.
  - Perceptual aliasing
    - When multiple situations can't be discriminated. I.e., multiple states give the same observation.
      - e.g. what is behind a wall?
Example: predator-prey

- Fully observable
- o=s=(-3,4)
Example: predator-prey

- Partially observable – perceptual aliasing
- $o = \text{Null}$
Example: predator-prey

- Partially observable – (noise?)
- \( o=(-1,1) \)
Policies under partial observability?

- Now given that the agent only gets some observations, what policy should he follow?
  - How does such a policy look?
Policies under partial observability?

• Now given that the agent only gets some observations, what policy should he follow?
  – How does such a policy look?

• No more Markovian signal (i.e. the state) directly available to the agent...
  ➔ In general: should use all information!
  ➔ The full history of observations.

• We will do something smarter in a moment...
A full POMDP: the Tiger problem

- **States**: left / right (50% prob.)
- **Actions**: Open left, open right, listen
- **Observation**: Hear left, Hear right
- **Transitions**: static, but opening resets.
- **Rewards**:
  - correct door +10,
  - wrong door -100
  - listen -1
- **Observations** are correct 85% of the time.
  - \( P(\text{HearLeft} | \text{Listen}, \text{State=left} ) = 0.85 \)
  - \( P(\text{HearRight} | \text{Listen}, \text{State=left} ) = 0.15 \)
The Tiger problem

• When do you open...?

• At the beginning?

• After HL?

• After HL, HL?

• After HL, HL, HL?
Beliefs

- As promised: there is something smarter than trying all possible policies.
  - mappings from obs. histories -> actions is approx. \( A^{O^t} \)

- Maintain the probability of all states.
  - Use that to make your decisions.
    - Did you estimate the probability of the states for the tiger problem?
  - The probability distribution over states at some time step, is called the belief \( b \).
    - For all \( s \): \( b(s) = \Pr(s) \)
    - Sufficient statistic for the history.
Beliefs: an example

- For the hallway problem
Calculating the belief

- A POMDP is often specified with an initial belief.
  - So we want to keep track of the probs. of the states.
  - I.e., given \( b, a \) and \( o \), we want to find the new belief \( b'_{ao} \).
  - Process is called belief update.

**DO not forget:** the term `belief` can be misleading.

**Not:** `something that one agent can belief, but some other agent would not'
**But:** The actual probability of the states, given the history.
Belief update – prerequisites

- $b'_{ao}$ can be calculated from $b$ and $T$, $O$... (resp. the transition, observation model)
- ...using Bayes' rule.

Bayes rule:

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$
Belief update

- substituting relevant vars in Bayes' rule.

\[
P(s'|o) = \frac{P(o|s') P(s')}{P(o)}
\]

- adding same arguments to `given`

\[
P(s'|b,a,o) = \frac{P(o|b,a,s') P(s'|b,a)}{P(o|b,a)}
\]

- expanding \( P(s'|b,a) \) gives the belief update:

\[
b'_{ao}(s') = \frac{P(o|a,s') \sum_s P(s'|s,a) b(s)}{P(o|b,a)}
\]

with \( P(o|b,a) = \sum_{s'} P(o|a,s') \sum_s P(s'|s,a) b(s) \)
POMDPs: making decisions

- Now we know how to maintain a belief over states...
  - but what decisions should we make?

- We treat 3 methods
  - Approximate
    - most likely state (MLS)
    - $Q_{MDP}$
  - Exact, given the initial belief
    - Solving the `belief MDP'
Most likely state

- Take the action that would seem best in...
  ...the most likely state $s_{ml}$.
  - I.e., state with highest probability.
  - $b = (0.1 \ 0.3 \ 0.5 \ 0.1)^T$ -> state 3

- But what is the best action in $s_{ml}$?
  - Solve the `underlying MDP'.
    - pretend there are no observations.
    - Solve the MDP.
    - Result: the MDP policy $\pi_{MDP}$
  - Perform action $\pi_{MDP} (s_{ml})$. 
Q-MDP

- Also uses solution of the `underlying MDP'
  - but now uses the found Q values, not the policy.

- Find the MDP Q(s,a)-values
  - E.g., using value iteration.

- Given the current belief b, for each action compute
  \[
  Q(b, a) = \sum_s Q(s, a) b(s)
  \]

- select the action with highest Q-value
  \[
  a_{Qmdp} = \text{arg max}_a Q(b, a)
  \]
Solve the beliefs MDP

• For a finite (and not too large) horizon...
• and given an initial belief...
  ➔ we can compute all possible beliefs.
  – `belief tree'
• Propagate back the expected reward

\[
V(b) = \max_a \left( R(b, a) + \sum_o P(o|b, a) V(b_{ao}) \right)
\]

with

\[
R(b, a) = \sum_s R(s, a) b(s)
\]

– The optimal action \(a^*\) is the one that maximizes the above expression.
Pros and cons

• Exact (‘belief MDP’).
  - Gives the optimal policy.
  - Only applicable to fairly small problems.
    • Few actions and observations.
    • Small horizon.

• Approximate (MLS, Q-MDP)
  - Scales to larger problems.
    • Solving the underlying MDP is the hardest.
    • Also selecting the final action can be done on-line.
  - Not optimal:
    • Too positive.
    • Information gaining actions are undervaluated.
Solving for ANY initial belief

- In some cases no initial belief $b^0$ available.
  - Perform planning for all possible initial beliefs.

- This is possible because of special property of the POMDP value function:
  - Piecewise-linear and convex (PWLC)

- Like VI for MDPs: use a backup operator $H$
  - $V_{k+1} = HV_k$
  - inf. horizon: $V^* = HV^*$

$$V(b) = \max_a \left( R(b, a) + \gamma \sum_o P(o|b, a)V(b_{ao}) \right)$$
**PWLC-property**

- $V_k$ is PWLC (when $k$ is finite)
  - Can be represented by a set of vectors.

\[
V_n(b) = \max_i b \cdot \alpha_n^i
\]

![Graph showing PWLC-property](image)
PWLC in 3D

- 3 states

- Generalizes to arbitrary number of states.
  - Although hard to visualize.
A numeric example

- $V_0$ given by the immediate rewards

<table>
<thead>
<tr>
<th>$R(s, a)$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$s_2$</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>
Constructing $V_{k+1}$ from $V_k$

- Basic procedure for a particular belief $b$
  - for all $a$
    - $\alpha_{\text{temp}} = (0 \ldots 0)^T$
    - for all $o$
      - calculate $b_{ao}$
      - Select $\alpha_{ao}$ the maximizing vector from $V_k$ at $b_{ao}$
      - $\alpha_{\text{temp}} += P(o|b,a) \times \alpha_{ao}$
    - create a new vector: $\alpha_{a} = R_a + \alpha_{\text{temp}}$
      - Select the action that maximizes $\alpha_{a} \cdot b$

- However, need to do this for all beliefs...
  - Just generate all possible vectors.
Summary

- Planning in a partially observable world.
- In such a setting an agent can maintain a belief over states.
  - using Bayes' Rule
- We considered 3 planning methods for use with an initial belief:
  - Exact: `solving the belief MDP`
  - Approximate: MLS and Q-MDP
- When no initial belief:
  - use PWLC property to generate a value function.
- PWLC property also basis for more advanced algorithms.