Robot grocery shopping in partially observable settings

Rodrigo Gomes, Xiaomin Wang, Dustin Tran

May 13, 2015

MIT, 6.834j Cognitive Robotics

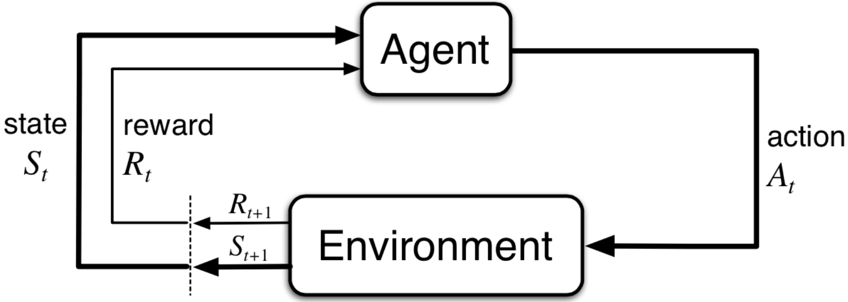
Outline

1. Background on POMDPs
2. Grocery shopping as planning in a POMDP
3. Demo!
4. What worked
5. What failed

A *partially observable Markov decision process* (POMDP) is a tuple (S, A, Ω, R, T, O)

- S: state space
- A: action space
- Ω: observation space
- $R : S \times A \to \mathbb{R}$ reward function
- T: transition operator. $T(s' \mid s, a)$ is probability of next state s' given state s and action a
- O: observable operator. $O(o \mid s)$ is probability of observing o given at state s

Background



A POMDP induces an equivalent representation as a *belief MDP* with tuple (B, A, τ, R)

- B: set of belief states over the POMDP states
- A: action space of original POMDP
- τ: belief state transition operator[1]

$$\tau(b, a, b') = \sum_{o \in \Omega} P(b' \mid b, a, o) P(o \mid a, b)$$

- $r : B \times A \to \mathbb{R}$ belief state reward function

$$r(b, a) = \sum_{s \in S} b(s) R(s, a)$$

[1] Given $b(s)$, after taking action a and observing o (and reaching state s'), update belief states

$$b'(s') = \frac{P(o \mid b, a, s')}{P(o \mid b, a)} = \frac{O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)}{\sum_{s' \in S} O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)} \tag{1}$$

A POMDP induces an equivalent representation as a *belief MDP* with tuple (B, A, τ, R)

- B: set of belief states over the POMDP states
- A: action space of original POMDP
- τ: belief state transition operator[2]

$$\tau(b, a, b') = \sum_{o \in \Omega} P(b' \mid b, a, o) P(o \mid a, b)$$

- $r : B \times A \rightarrow \mathbb{R}$ belief state reward function

$$r(b, a) = \sum_{s \in S} b(s) R(s, a)$$

[2] Given $b(s)$, after taking action a and observing o (and reaching state s'), update belief states

$$b'(s') = \frac{P(o \mid b, a, s')}{P(o \mid b, a)} = \frac{O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)}{\sum_{s' \in S} O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)} \tag{2}$$

Background

Implemented MDP solvers:

- Q-learning
- SARSA
- R-MAX
- Thompson sampling

There are a lot!

- Function approximations with adaptive basis functions
- BOSS
- Spectral methods
- Skill chaining
- . . .

Implemented MDP solvers:

- Q-learning (Watkins, 1989)
- SARSA (Rummery and Niranjan, 1994)
- R-MAX (Brafman and Tennenholtz, 2002)
- Thompson sampling (Strens, 2000)

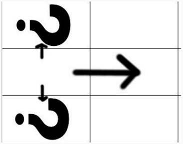
There are a lot more!

- Function approximations with adaptive basis functions (Mnih et al., 2013)
- BOSS (Asmuth et al., 2009)
- Spectral methods (Boots et al., 2009)
- Skill chaining (Konidaris and Barto, 2009)
- ...

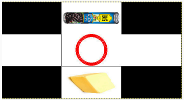
Grocery shopping

Setup: Grid World POMDP

Uncertain movement



Can only see around current cell (partially observable)



World is not fully known beforehand

- Model of how items in the same aisle correlate
- Unknown arrangement of aisles
- Unknown arrangement of items within aisles

Grocery shopping

GUI interface: *pygame*

Every second:

- Agent provides next action based on current belief state
- Simulator executes action (errors may happen)
- Belief state is updated based on transition probabilities
- Belief state is updated based on observation
- Belief about the world is updated based on belief state, and observation

Challenges:

- Markov assumption is not completely accurate
- Bias towards increasing probability of most likely states

demo

We encode a **Max Probability** MDP

- Motivated from greedy policies
- Choose the most likely state from belief states as one's position in an MDP
- Solve the MDP!

Our working solver

Value iteration:

$$
\begin{aligned}
v_{k+1}(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s'} p(s' \mid s, a)[r(s, a, s') + \gamma v_k(s')]
\end{aligned}
$$

Failed tasks

- Continuous state space in belief MDP: Value iteration
- Thompson sampling
- TD(λ) methods: Q-Learning, SARSA, Monte Carlo Tree Search

Most simplified task (GridWorld)

Play with it!

github.com/dustinvtran/bayesrl