MDP - POMDP - Dec-POMDP

Alina Vereshchaka

CSE4/510 Reinforcement Learning
Fall 2019

avereshc@buffalo.edu

November 12, 2019

*Some materials are taken from Decision Making under Uncertainty by Mykel J. Kochenderfer

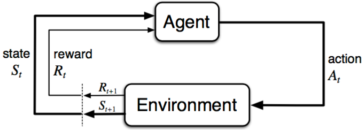
Overview

1. MDP - POMDP
2. Decentralized Partially Observable Markov Decision Process (Dec-POMDP)
3. Multiagent Settings

Table of Contents

Markov Decision Process (MDP)

RL can be formalized as a MDP with $\langle S, A, P, r\gamma \rangle$

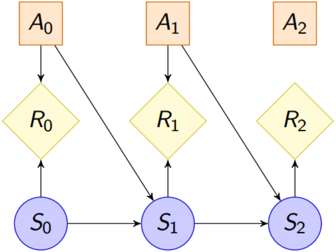


- Markov Property: $P(s_{t+1}|s_1, a_1, \cdots, s_t, a_t) = P(s_{t+1}|s_t, a_t)$
- A policy π is a map from state to action
  - Deterministic policy: $a = \pi(s)$ or $a = \mu(s)$
  - Stochastic policy: $\pi(a|s) = P[a_t = a|s_t = s]$

Definition

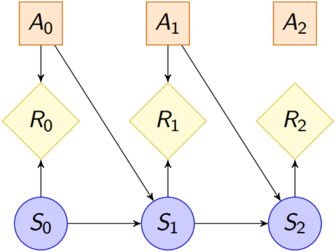
Goal of RL is to find an optimal policy π^* in order to maximize the expected discounted reward: $J(\pi) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t)\right]$

Stochastic Problem



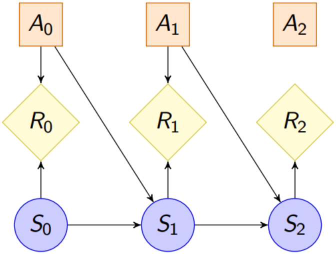
- Agent chooses action A_t at time t based on observing state S_t

Stochastic Problem



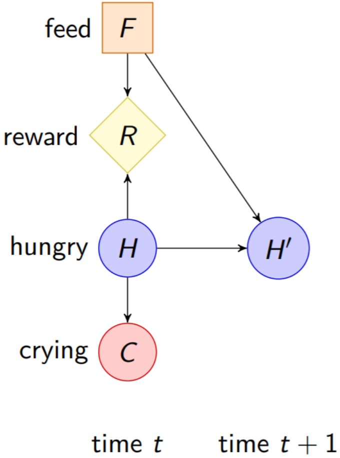
- Agent chooses action A_t at time t based on observing state S_t
- State evolves probabilistically based on current state and action taken by agent (Markov assumption)

Stochastic Problem



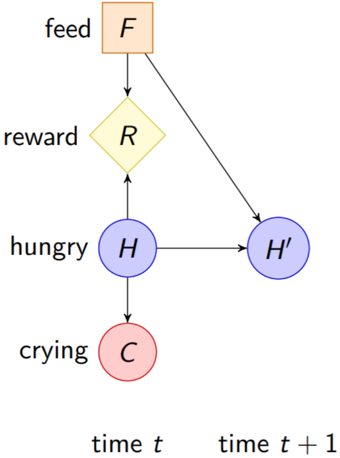
- Agent chooses action A_t at time t based on observing state S_t
- State evolves probabilistically based on current state and action taken by agent (Markov assumption)
- Objective is to maximize sum of rewards R

Crying Baby Problem



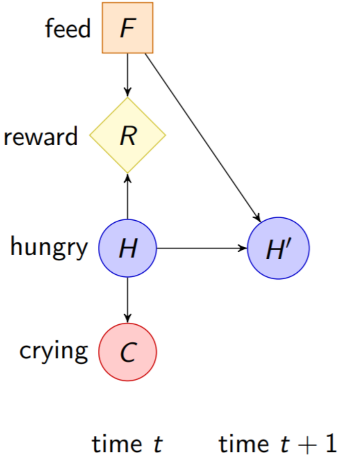
- Need to decide whether to feed baby given whether baby is crying

Crying Baby Problem

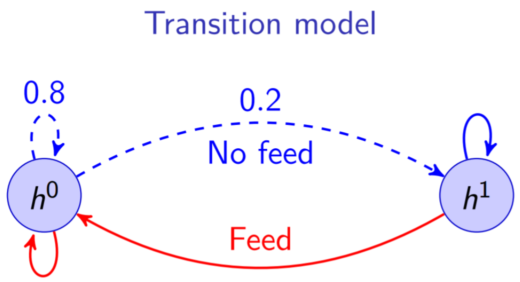


- Need to decide whether to feed baby given whether baby is crying
- Crying is a noisy indication that the baby is hungry

Crying Baby Problem



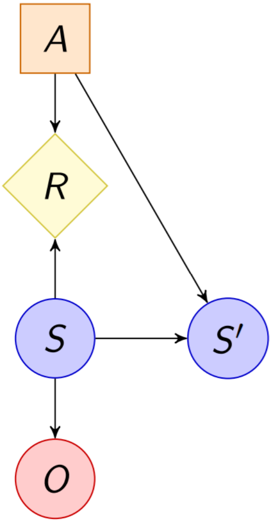
- Need to decide whether to feed baby given whether baby is crying
- Crying is a noisy indication that the baby is hungry



$P(c^1|h^0) = 0.2$ (cry when not hungry)

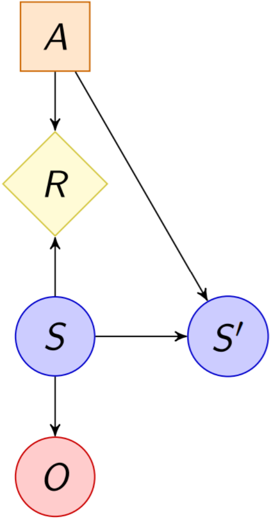
$P(c^1|h^1) = 0.8$ (cry when hungry)

Partially Observable Markov Decision Process (POMDP)



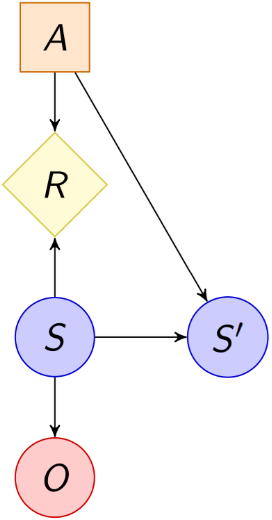
- POMDP = MDP + sensor model

Partially Observable Markov Decision Process (POMDP)



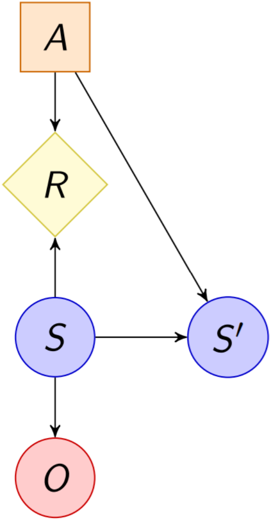
- POMDP = MDP + sensor model
- Sensor model: $O(o|s)$ or sometimes $O(o|s, a)$

Partially Observable Markov Decision Process (POMDP)



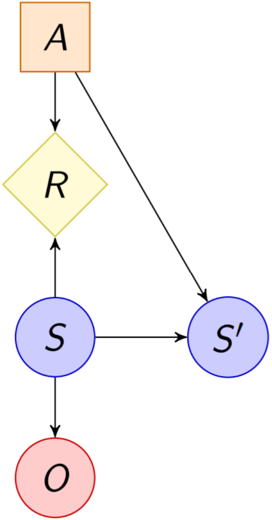
- POMDP = MDP + sensor model
- Sensor model: $O(o|s)$ or sometimes $O(o|s, a)$
- Decisions can only be based on history of observations $o_1, o_2, \cdots, o_t$

Partially Observable Markov Decision Process (POMDP)



- POMDP = MDP + sensor model
- Sensor model: $O(o|s)$ or sometimes $O(o|s, a)$
- Decisions can only be based on history of observations $o_1, o_2, \cdots, o_t$
- Instead of keeping track of arbitrarily long histories, we keep track of the belief state

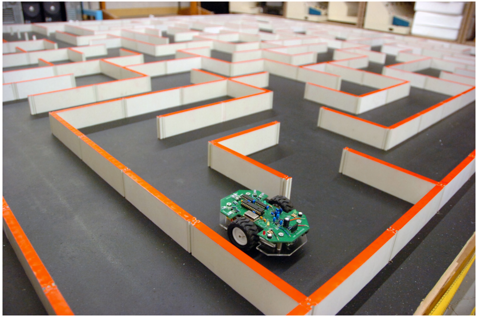
Partially Observable Markov Decision Process (POMDP)



- POMDP = MDP + sensor model
- Sensor model: $O(o|s)$ or sometimes $O(o|s, a)$
- Decisions can only be based on history of observations $o_1, o_2, \cdots, o_t$
- Instead of keeping track of arbitrarily long histories, we keep track of the belief state
- A belief state is a distribution over states; in belief state b, probability $b(s)$ is assigned to being in s

Partially Observable Markov Decision Process (POMDP)

- Agent observes the entire environment $\rightarrow$ **MDP**
- Agent only observes a part of environment $\rightarrow$ **POMDP**
- POMDP is popular is the real-world applications



(a) Robot Navigation in Maze

(b) Self-Driving Car

Partially Observable Markov Decision Process (POMDP)

A POMDP is a tuple $(S, A, T, R, \Omega, O, \gamma)$, where

- S is a set of states
- A_i is a set of actions
- T is a set of conditional transition probabilities between states
- $R : S \times A \to \mathbb{R}$ is the reward function
- Ω_i is a set of observations
- O is a set of conditional observation probabilities $O(s', a, o) = P(o \mid s', a)$
- $\gamma \in [0, 1)$ is the discount factor

Computing Belief States

- Begin with some initial belief state b prior to any observations
- Compute new belief state b' based on current belief state b, action a, and observation o

$$b'(s') = P(s'|o, a, b)$$

Computing Belief States

- Begin with some initial belief state b prior to any observations
- Compute new belief state b' based on current belief state b, action a, and observation o

$$
\begin{aligned}
b'(s') &= P(s'|o, a, b) \\
&= P(o|s', a, b)P(s'|a, b)
\end{aligned}
$$

Computing Belief States

- Begin with some initial belief state b prior to any observations
- Compute new belief state b' based on current belief state b, action a, and observation o

$$
\begin{aligned}
b'(s') &= P(s'|o, a, b) \\
&= P(o|s', a, b)P(s'|a, b) \\
&= O(o|s', a)P(s'|a, b)
\end{aligned}
$$

Computing Belief States

- Begin with some initial belief state b prior to any observations
- Compute new belief state b' based on current belief state b, action a, and observation o

$$\begin{aligned}
b'(s') &= P(s'|o, a, b) \\
&= P(o|s', a, b)P(s'|a, b) \\
&= O(o|s', a)P(s'|a, b) \\
&= O(o|s', a)\sum_{s} P(s'|a, b, s)P(s|a, b)
\end{aligned}$$

Computing Belief States

- Begin with some initial belief state b prior to any observations
- Compute new belief state b' based on current belief state b, action a, and observation o

$$
\begin{aligned}
b'(s') &= P(s'|o, a, b) \\
&= P(o|s', a, b)P(s'|a, b) \\
&= O(o|s', a)P(s'|a, b) \\
&= O(o|s', a)\sum_{s} P(s'|a, b, s)P(s|a, b) \\
&= O(o|s', a)\sum_{s} T(s'|s, a){\color{red}b(s)}
\end{aligned}
$$

Computing Belief States

- Begin with some initial belief state b prior to any observations
- Compute new belief state b' based on current belief state b, action a, and observation o

$$
\begin{aligned}
b'(s') &= P(s'|o, a, b) \\
&= P(o|s', a, b)P(s'|a, b) \\
&= O(o|s', a)P(s'|a, b) \\
&= O(o|s', a)\sum_{s} P(s'|a, b, s)P(s|a, b) \\
&= O(o|s', a)\sum_{s} T(s'|s, a)b(s)
\end{aligned}
$$

- Kalman filter: exact update of the belief state for linear dynamical systems

Computing Belief States

- Begin with some initial belief state b prior to any observations
- Compute new belief state b' based on current belief state b, action a, and observation o

$$\begin{aligned} b'(s') &= P(s'|o,a,b) \\ &= P(o|s',a,b)P(s'|a,b) \\ &= O(o|s',a)P(s'|a,b) \\ &= O(o|s',a)\sum_s P(s'|a,b,s)P(s|a,b) \\ &= O(o|s',a)\sum_s T(s'|s,a)b(s) \end{aligned}$$

- Kalman filter: exact update of the belief state for linear dynamical systems
- Particle filter: approximate update for general systems

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

No feed, no cry

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

No feed, no cry

$b = (0.9759, 0.0241)$

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

No feed, no cry

$b = (0.9759, 0.0241)$

No feed, no cry

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

No feed, no cry

$b = (0.9759, 0.0241)$

No feed, no cry

$b = (0.9701, 0.0299)$

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

No feed, no cry

$b = (0.9759, 0.0241)$

No feed, no cry

$b = (0.9701, 0.0299)$

No feed, cry

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

No feed, no cry

$b = (0.9759, 0.0241)$

No feed, no cry

$b = (0.9701, 0.0299)$

No feed, cry

$b = (0.4624, 0.5376)$

Crying Baby Example

$b = (h^0, h^1) = (0.5, 0.5)$

No feed, cry

$b = (0.0928, 0.9072)$

Feed, no cry

$b = (1, 0)$

No feed, no cry

$b = (0.9759, 0.0241)$

No feed, no cry

$b = (0.9701, 0.0299)$

No feed, cry

$b = (0.4624, 0.5376)$

POMDP Execution

1. Initialize belief state b
2. Execute $a = \pi(b)$
3. Observe o
4. Update b based on b, a, and o
5. Go to 2

Markov Models

	No Agents	Single Agent	Multiple Agents
State Known	Markov Chain	Markov Decision Process (MDP)	Markov Game (a.k.a. Stochastic Game)
State Observed Indirectly	Hidden Markov Model (HMM)	Partially-Observable Markov Decision Process (POMDP)	Partially-Observable Stochastic Game (POSG)

Table of Contents

Dec-POMDP

- The decentralized partially observable Markov decision process (Dec-POMDP) is a model for coordination and decision-making among multiple agents.

Dec-POMDP

- The decentralized partially observable Markov decision process (Dec-POMDP) is a model for coordination and decision-making among multiple agents.
- It is a probabilistic model that can consider uncertainty in outcomes, sensors and communication (i.e., costly, delayed, noisy or nonexistent communication)

Dec-POMDP

- The decentralized partially observable Markov decision process (Dec-POMDP) is a model for coordination and decision-making among multiple agents.
- It is a probabilistic model that can consider uncertainty in outcomes, sensors and communication (i.e., costly, delayed, noisy or nonexistent communication)
- It is a generalization of a Markov decision process (MDP) and a partially observable Markov decision process (POMDP) to consider multiple decentralized agents.

Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

A Dec-POMDP is a tuple $(S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma)$, where

- S is a set of states
- A_i is a set of actions for agent i, with $A = \times_i A_i$ is the set of joint actions
- T is a set of conditional transition probabilities between states, $T(s, a, s') = P(s' \mid s, a)$
- $R : S \times A \to \mathbb{R}$ is the reward function
- Ω_i is a set of observations for agent i, with $\Omega = \times_i \Omega_i$ is the set of joint observations,
- O is a set of conditional observation probabilities $O(s', a, o) = P(o \mid s', a)$
- $\gamma \in (0, 1]$ is the discount factor

Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

- Agents must consider the choices of all others in addition to the state and action uncertainty present in POMDPs
- This makes DEC-POMDPs much harder to solve
- No common state estimate (centralized belief state)
  - Each agent depends on the others
  - This requires a belief over the possible policies of the other agents

What problems Dec-POMDPs are good for?

- Sequential (not "one shot" or greedy)
- Cooperative (not single agent or competitive)
- Decentralized (not centralized execution or free, instantaneous communication)
- Decision-theoretic (probabilities and values)

MDP, POMDP and Dec-POMDP

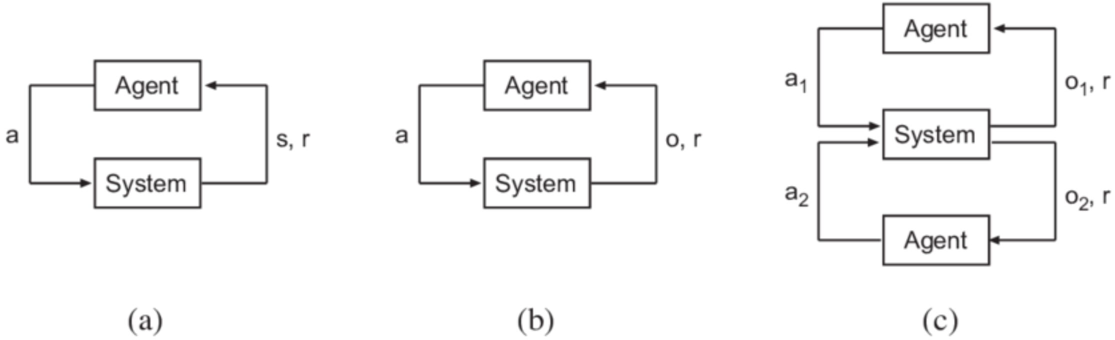
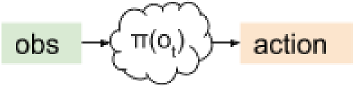


Figure: (a) Markov decision process (MDP) (b) Partially observable Markov decision process (POMDP) (c) Decentralized partially observable Markov decision process with two agents (Dec-POMDP)

Table of Contents

Single Agent

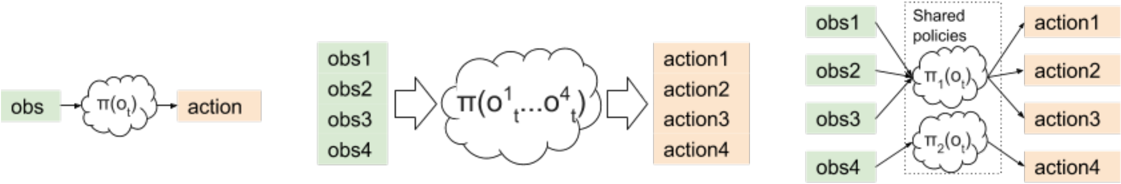


(a) Single-agent

Single 'Super' Agent

(a) Single-agent (b) Multiple logical entities, single "super-agent"

Multiagent



(a) Single-agent (b) Multiple logical entities, single "super-agent" (c) Multi-agent