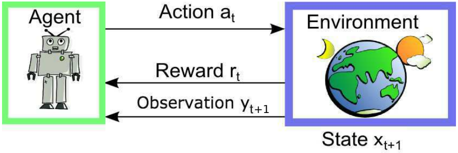
Open Problem: Approximate Planning of POMDPs in the class of Memoryless Policies

Kamyar Azizzadenesheli

U.C. Irvine

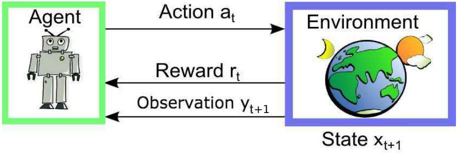
Joint work with Prof. Anima Anandkumar and Dr. Alessandro Lazaric.

Motivation



- Agent-Environment Interaction.

Motivation

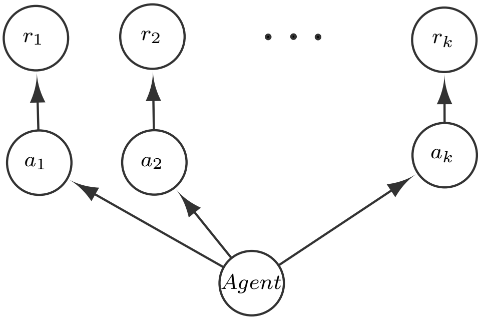


- Agent-Environment Interaction.
- Multi-Armed-Bandit
- Markov Decision Process (MDP)
- Partially Observable Markov Decision Process (POMDP)
- Full information $\rightarrow$ Planning $\rightarrow$ policy π

Planning

Multi Armed Bandit

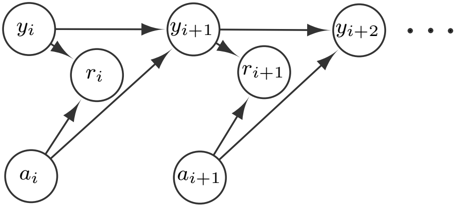
- Single state,
- Action with highest expectation reward.



Planning

Markov Decision Process (MDP)

- Fully Observable Environment: $y = x$.
- Markovian Assumption:
  - $P(y_{t+1}|a_t, y_t) \quad P(r_t|a_t, y_t)$



Planning

Markov Decision Process (MDP)

Discounted Reward $:= \max_{\pi} \sum_{t} \lambda^t r_t \rightarrow$ Bellman Equation $(0 \leq \lambda < 1)$

Planning

Markov Decision Process (MDP)

Discounted Reward $:= \max_{\pi} \sum_{t} \lambda^t r_t \rightarrow$ Bellman Equation $(0 \leq \lambda < 1)$

$$Q(a,x) = \mathbb{E}[r(x,a)] + \lambda \sum_{x'} P(x'|a,x) \max_{a'}\{Q(a',x')\}$$

Planning

Markov Decision Process (MDP)

Discounted Reward $:= \max_{\pi} \sum_{t} \lambda^{t} r_{t} \rightarrow$ Bellman Equation $(0 \leq \lambda < 1)$

$$Q(a,x) = \mathbb{E}[r(x,a)] + \lambda \sum_{x'} P(x'|a,x) \max_{a'}\{Q(a',x')\}$$

Long Term Average Reward $:= \max_{\pi} \sum_{t} r_{t} \rightarrow$ Poisson Equation

Planning

Markov Decision Process (MDP)

Discounted Reward $:= \max_{\pi} \sum_{t} \lambda^t r_t \rightarrow$ Bellman Equation $(0 \leq \lambda < 1)$

$$Q(a,x) = \mathbb{E}[r(x,a)] + \lambda \sum_{x'} P(x'|a,x) \max_{a'}\{Q(a',x')\}$$

Long Term Average Reward $:= \max_{\pi} \sum_{t} r_t \rightarrow$ Poisson Equation

$$1)\pi$$

$$2)R_\pi(x),\ \ P_\pi(x'|x),\ \ \eta_\pi$$

$$3)(I - P_\pi)V + \eta e = R_\pi \ \ \rightarrow \ \ V \ \ := \ \ (\textit{Performance Potential})$$

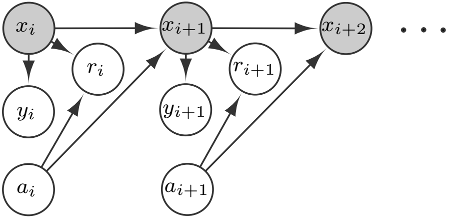
$$4)P_{\pi'}V + R_{\pi'} \succeq P_\pi V + R_\pi$$

$$5)\pi \leftarrow \pi'$$

Planning

Partially Observable Markov Decision Process (POMDP)

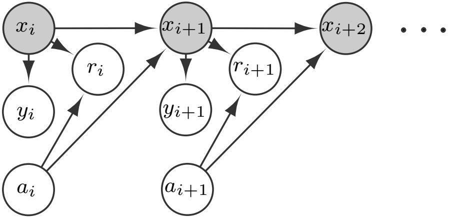
- Partially Observability,
- Transition Probability $P(x_t|a_t, x_t)$,
- Observation Distribution $P(y_t|x_t)$.



Planning

Partially Observable Markov Decision Process (POMDP)

- Partially Observability,
- Transition Probability $P(x_t|a_t, x_t)$,
- Observation Distribution $P(y_t|x_t)$.

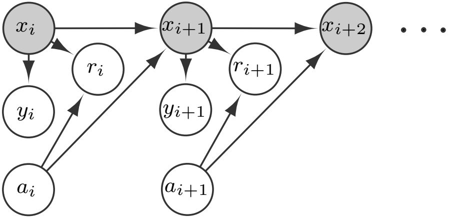


Efficient Learning Algorithm by Tensor Methods,

Planning

Partially Observable Markov Decision Process (POMDP)

- Partially Observability,
- Transition Probability $P(x_t|a_t, x_t)$,
- Observation Distribution $P(y_t|x_t)$.



Efficient Learning Algorithm by Tensor Methods,

Learning part is solved, remaining part is the planning.

Planning

Partially Observable Markov Decision Process (POMDP)

- Distribution over states $[b(x_1), \ldots, b(x_k)]$,
- Apply action a, and observe y,

$$b'(x') = \frac{P(y|x') \sum_x P(x'|x,a) b(x)}{P(y|b,a))},$$

Planning

Partially Observable Markov Decision Process (POMDP)

- Distribution over states $[b(x_1), \ldots, b(x_k)]$,
- Apply action a, and observe y,

$$b'(x') = \frac{P(y|x') \sum_x P(x'|x,a) b(x)}{P(y|b,a))},$$

Bellman Equation: $(0 \leq \lambda < 1)$

$$Q(a,b) = \mathbb{E}[r(b,a)] + \lambda \sum_{b'} P(b'|a,b) \max_{a'} \{Q(a',b')\}.$$

Planning

Partially Observable Markov Decision Process (POMDP)

- Distribution over states $[b(x_1), \ldots, b(x_k)]$,
- Apply action a, and observe y,

$$b'(x') = \frac{P(y|x') \sum_x P(x'|x,a) b(x)}{P(y|b,a))},$$

Bellman Equation: $(0 \leq \lambda < 1)$

$$Q(a,b) = \mathbb{E}[r(b,a)] + \lambda \sum_{b'} P(b'|a,b) \max_{a'} \{Q(a',b')\}.$$

Belief point grows $\mathcal{O}((YA)^T)$

PSpace-Complete $\rightarrow$ Point-Based VI

Planning

Partially Observable Markov Decision Process (POMDP)

- Memory Less Policy, Limited Memory Policy
  $\pi(a_t|y_t, y_{t-1}, \ldots, y_{t-n+1})$

order-n policy

Planning

Partially Observable Markov Decision Process (POMDP)

- Memory Less Policy, Limited Memory Policy
  $\pi(a_t|y_t, y_{t-1}, \ldots, y_{t-n+1})$

  order-n policy

- In some POMDP settings optimal policy is order-n policy,

Planning

Partially Observable Markov Decision Process (POMDP)

- Memory Less Policy, Limited Memory Policy
  $\pi(a_t|y_t, y_{t-1}, \ldots, y_{t-n+1})$

  order-n policy

- In some POMDP settings optimal policy is order-n policy,

- Advantages: Memory Efficient, no need to solve PSpace-Complete problem,

Planning

Partially Observable Markov Decision Process (POMDP)

- Memory Less Policy, Limited Memory Policy
  $\pi(a_t|y_t, y_{t-1}, \ldots, y_{t-n+1})$

  order-n policy

- In some POMDP settings optimal policy is order-n policy,

- Advantages: Memory Efficient, no need to solve PSpace-Complete problem,

- Optimal Deterministic memoryless policy, [Yanjie Li], [John Loch].

Planning

Partially Observable Markov Decision Process (POMDP)

Optimal memoryless policy in general is stochastic,

Q-function is not contractive mapping,

Optimization Problem $\Rightarrow$

$$\boxed{\eta = \max_{\pi} \sum_t r_t \;=\; \max_{\pi} \sum_x P_{\pi}(x) R_{\pi}(x)}$$

Where $R_{\pi}(x) = \sum_a \sum_y P(y|x)\pi(a|y)r(a,x)$.

Planning

Partially Observable Markov Decision Process (POMDP)

Optimal memoryless policy in general is stochastic,

Q-function is not contractive mapping,

Optimization Problem $\Rightarrow$

$$\boxed{\eta = \max_{\pi} \sum_{t} r_t \; = \; \max_{\pi} \sum_{x} P_{\pi}(x) R_{\pi}(x)}$$

Where $R_{\pi}(x) = \sum_{a} \sum_{y} P(y|x)\pi(a|y)r(a,x)$.

Solution??

Thank You!