Probabilistic Graphical Models

Prof. Joschka Boedecker and Hao Zhu

Department of Computer Science
University of Freiburg

universität freiburg

6. Markov decision problems

- Markov decision processes
  - Episodes and returns
  - Value functions
  - Optimal value functions and policies

- Dynamic programming
  - Policy iteration
  - Value iteration

Outline

- Markov decision processes
  - Episodes and returns
  - Value functions
  - Optimal value functions and policies
- Dynamic programming
  - Policy iteration
  - Value iteration

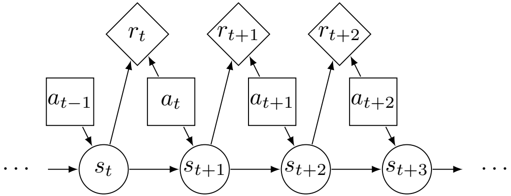
Markov decision processes (MDPs)

- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r : \mathcal{S} \times \mathcal{A} \to \mathbf{R}$: reward function
- π: policy

- the probability of generating a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)$ of length T:

$$p(\tau) = p(s_0) \prod_{t=0}^{T-1} \pi(a_t \mid s_t) p(s_{t+1} \mid s_t, a_t)$$

  - $p(s' \mid s, a)$: transition function



Episodes

- **continuing task**: the agent can potentially interact with the environment forever
- **episodic task**: the interaction terminates once the system enters a terminal state or absorbing state (the next state is always itself with 0 reward)
  - after entering a terminal state, agent starts a new episode from a new initial state $s_0 \sim p(s_0)$
  - the episode length is in general random
  - **finite horizon problems**: the trajectory length T in an episodic task is fixed and known

Returns

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{T-t-1} r_{T-1}$$
$$= \sum_{k=0}^{T-t-1} \gamma^k r_{t+k} = \sum_{i=t}^{T-1} \gamma^{i-t} r_i$$

- G_t: return at time t, the sum of expected rewards obtained going forwards
- $\gamma \in [0, 1]$: discount factor
  - if $\gamma < 1$ and rewards r_t are bounded $\implies$ G_t is always bounded even if $T \to \infty$
- $G_t = 0$, for $t \geq T$, if episode tasks terminate at T
- recursive expression:

$$G_t = r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \cdots) = r_t + \gamma G_{t+1}$$

Value functions

state-value function

$$V^\pi(s) = \mathbf{E}_\pi[G_0 \mid s_0 = s] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right], \quad \text{for all } s \in \mathcal{S}$$

- the expected return starting in state $s \in \mathcal{S}$ and follow π to choose actions

action-value function

$$Q^\pi(s, a) = \mathbf{E}_\pi[G_0 \mid s_0 = s, a_0 = a] = \mathbf{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a\right], \quad \text{for all } s \in \mathcal{S}$$

- the expected return starting by taking action a in state s, and then follow policy π

Value functions

advantage function

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$$

- the benefit of picking action a in state s then switching to policy π, relative to the baseline return of always following π
- $\mathbf{E}_{\pi(s|a)}[A^{\pi}(s,a)] = 0$, since

$$V^{\pi}(s) = \mathbf{E}_{\pi(a|s)}[Q^{\pi}(s,a)]$$

Value functions

Bellman equations: recursive expression of value functions

$$\begin{aligned} V^\pi(s) &= \mathbf{E}_\pi[G_0 \mid s_0 = s] = \mathbf{E}_\pi[r_0 + \gamma G_1 \mid s_0 = s] \\ &= \sum_{a \in \mathcal{A}} \pi(a \mid s) \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \mathbf{E}_\pi[G_1 \mid s_1 = s'] \right] \\ &= \sum_{a \in \mathcal{A}} \pi(a \mid s) \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^\pi(s') \right] \end{aligned}$$

$$\begin{aligned} Q^\pi(s,a) &= \mathbf{E}_\pi[G_0 \mid s_0 = s, a_0 = a] = \mathbf{E}_\pi[r_0 + \gamma G_1 \mid s_0 = s, a_0 = a] \\ &= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \mathbf{E}_\pi[G_1 \mid s_1 = s'] \\ &= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \sum_{a' \in \mathcal{A}} \pi(a' \mid s') Q^\pi(s',a') \end{aligned}$$

Optimal value functions and policies

optimal policy π^* $\implies$ $V^{\pi^*} \geq V^\pi$ for all $s \in \mathcal{S}$ and all policy π

- V^*, Q^*: optimal value functions
- multiple optimal policies for one MDP have the same value functions

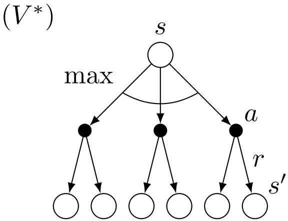
Bellman optimality equations

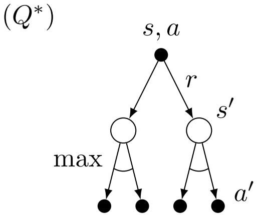
$$\begin{aligned}
V^*(s) &= \max_{a\in\mathcal{A}} Q^*(s,a) = \max_{a\in\mathcal{A}} \mathbf{E}_{\pi^*}[G_0 \mid s_0 = s, a_0 = a] \\
&= \max_{a\in\mathcal{A}} \mathbf{E}_{\pi^*}[r_0 + \gamma G_1 \mid s_0 = s, a_0 = a] \\
&= \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s' \mid s,a) \mathbf{E}_{\pi^*}[G_1 \mid s_1 = s'] \right] \\
&= \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s' \mid s,a) V^*(s') \right]
\end{aligned}$$

Optimal value functions and policies

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \max_{a' \in \mathcal{A}} Q^*(s',a')$$

- the discrepancy between the right- and left-hand sides are called Bellman error
- the Bellman optimality equations has a unique solution π^* for finite MDPs





Optimal value functions and policies

- given optimal value functions V^* and Q^*, the optimal policy π^* can be obtained according to

$$\pi^*(s) = \operatorname*{argmax}_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^*(s') \right]$$

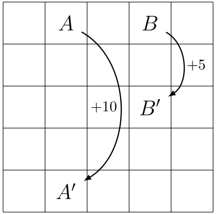
or

$$\pi^*(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^*(s,a)$$

Example: gridworld

- $\mathcal{A} = \{$*up*, *down*, *left*, *right*$\}$
- from state A, all four actions yield a reward of $+10$ and take the agent to A'
- from state B, all actions yield a reward of $+5$ and take the agent to B'
- actions taking the agent off the grid leave its location unchanged with a reward of -1, and all other actions result in a reward of 0

Gridworld



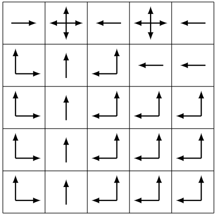
V^{rand}

3.8	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	−0.4
−1.0	−0.4	−0.4	−0.6	−1.2
−1.9	−1.3	−1.2	−1.4	−2.0

V^*

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

π^*



Outline

- Markov decision processes
  - Episodes and returns
  - Value functions
  - Optimal value functions and policies
- Dynamic programming
  - Policy iteration
  - Value iteration

Policy iteration

policy evaluation: given some policy π, evaluate

$$V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a \mid s) \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s' \mid s,a) V^\pi(s') \right], \quad \text{for all } s \in \mathcal{S}$$

- for finite MDPs: solving a system of $\mathbf{card}(\mathcal{S})$ linear equations with $\mathbf{card}(\mathcal{S})$ unknowns

iterative policy evaluation: approximate V^π with the sequence $V^{(0)}, V^{(1)}, V^{(2)}, \ldots$, where

$$V^{(i+1)}(s) = \sum_{a\in\mathcal{A}} \pi(a \mid s) \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s' \mid s,a) V^{(i)}(s') \right]$$

- $V^{(0)}, V^{(1)}, \ldots, V^{(i)}, \ldots$ converges to V^π as $i \to \infty$

Policy iteration

- **'two array' implementation**: use two arrays, one for the old values $V^{(i)}$, and one for the new values $V^{(i+1)}$, then the new values can be computed one by one from the old values without the old values being changed
- **'in place' implementation**: use one array of V, and with each new value immediately overwriting the old one

---

given the policy π to be evaluated.
initialize $V(s)$ for all $s \in \mathcal{S}$ arbitrarily, if s is not terminal, otherwise 0.
repeat
  for $s \in \mathcal{S}$ **do**
    $V(s) := \sum_{a \in \mathcal{A}} \pi(a \mid s) \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V(s') \right].$
  end for
until stop criterion reached.

- the in-place version usually converges faster than the two-array version, which is influenced by the order of states for update

Policy iteration

policy improvement: given the value function V^π for some policy π, find a new policy

$$\pi'(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^\pi(s,a) = \operatorname*{argmax}_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^\pi(s') \right], \quad \text{for all } s \in \mathcal{S}$$

- π' is as good as, or better than the old policy π
- if π' is as good as π, then $\pi' = \pi = \pi^*$
  - **proof**: suppose π' is as good as, but not better than π, i.e., $V^\pi = V^{\pi'}$, we have

$$V^{\pi'}(s) = \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^{\pi'}(s') \right], \quad \text{for all } s \in \mathcal{S}$$

Policy iteration

policy iteration: alternating between policy evaluation and policy improvement

$$\pi^{(0)} \xrightarrow{\text{E}} V^{\pi^{(0)}} \xrightarrow{\text{I}} \pi^{(1)} \xrightarrow{\text{E}} V^{\pi^{(1)}} \xrightarrow{\text{I}} \pi^{(2)} \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi^{*} \xrightarrow{\text{E}} V^{*}$$

- $\xrightarrow{\text{E}}$: policy evaluation
- $\xrightarrow{\text{I}}$: policy improvement

Policy iteration

```
1.  Initialization.
```
initialize $V(s) \in \mathbf{R}$ and $\pi(s) \in \mathcal{A}$ for all $s \in \mathcal{S}$.
repeat
```
    2.  Policy evaluation.
```
  repeat
    for $s \in \mathcal{S}$ **do**
      $V(s) := \sum_{a \in \mathcal{A}} \pi(a \mid s) \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V(s') \right].$
    end for
  until stop criterion reached.
```
    3.  Policy improvement.
```
  for $s \in \mathcal{S}$ **do**
    $\pi(s) := \mathrm{argmax}_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V(s') \right].$
  end for
until policy is stable.

Value iteration (VI)

solving Bellman optimality equations with iterative methods:

- approximate V^* with the sequence $V^{(0)}, V^{(1)}, V^{(2)}, \ldots$, where

$$V^{(i+1)}(s) = \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^{(i)}(s') \right]$$

  - $V^{(0)}, V^{(1)}, \ldots, V^{(i)}, \ldots$ converges to V^* as $i \to \infty$

- can be considered as policy improvement + (1-sweep) truncated policy evaluation

Value iteration (VI)

---

initialize $V(s)$ for all $s \in \mathcal{S}$ arbitrarily, if s is not terminal, otherwise 0.
repeat
  for $s \in \mathcal{S}$ **do**
    $V(s) := \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V(s') \right]$.
  end for
until stop criterion reached.
output a deterministic policy $\pi := \operatorname{argmax}_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V(s') \right]$.
