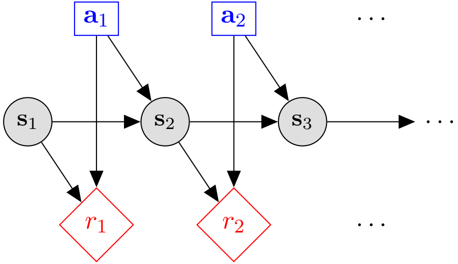
Machine Learning (CS 181): 21. Reinforcement Learning

David C. Parkes and Sasha Rush

Spring 2017

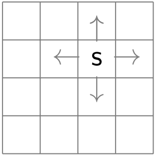
Contents

Markov Decision Process



S	States
A	Actions
$r : S \times A \mapsto \mathbb{R}$	Reward Function
$p(s' \mid s, a)$	Transition Model

Running Illustration: MDP on Gridworld



S	Location of the grid (x_1, x_2)
A	Local movements $\leftarrow, \rightarrow, \uparrow, \downarrow$
$r : S \times A \mapsto \mathbb{R}$	Reward function, e.g. make it to goal
$p(s' \mid s, a)$	Transition model, e.g deterministic or slippages

Policy Evaluation

- Policy function: $\pi : S \rightarrow A$

- Value Function: expected discounted reward

$$V^{\pi}(s) = \underbrace{r(s, \pi(s))}_{\text{reward now}} + \underbrace{\gamma \sum_{s' \in S} p(s' \mid s, \pi(s)) V^{\pi}(s)}_{\text{expected, discounted future reward}} \tag{1}$$

- Q-Function: expected discounted reward of state and action (new)

$$Q^{\pi}(s, a) = \underbrace{r(s, a)}_{\text{reward now}} + \underbrace{\gamma \sum_{s' \in S} p(s' \mid s, a) Q^{\pi}(s', \pi(s'))}_{\text{expected, discounted future reward}} \tag{2}$$

- Can compute Value function from Q-Function

$$V^{\pi}(s) = Q^{\pi}(s, \pi(s'))$$

Policy Evaluation

- Policy function: $\pi : S \to A$

- Value Function: expected discounted reward

$$V^\pi(s) = \underbrace{r(s, \pi(s))}_{\text{reward now}} + \underbrace{\gamma \sum_{s' \in S} p(s' \mid s, \pi(s)) V^\pi(s)}_{\text{expected, discounted future reward}} \quad (1)$$

- Q-Function: expected discounted reward of state and action (new)

$$Q^\pi(s, a) = \underbrace{r(s, a)}_{\text{reward now}} + \underbrace{\gamma \sum_{s' \in S} p(s' \mid s, a) Q^\pi(s', \pi(s'))}_{\text{expected, discounted future reward}} \quad (2)$$

- Can compute Value function from Q-Function

$$V^\pi(s) = Q^\pi(s, \pi(s'))$$

Working with MDPs

An MDP is a general probabilistic framework, and can be utilized in many different scenarios.

- Planning:
  - Full access to the MDP, compute an optimal policy.
  - "How do I act in a known world?"
- Policy Evaluation:
  - Full access to the MDP, compute the 'value' of a fixed policy.
  - "How will this plan perform under uncertainty?"
- Reinforcement Learning (today):
  - Limited access to the MDP.
  - "Can I learn to act in an uncertain world?"

Working with MDPs

An MDP is a general probabilistic framework, and can be utilized in many different scenarios.

- Planning:
  - Full access to the MDP, compute an optimal policy.
  - "How do I act in a known world?"
- Policy Evaluation:
  - Full access to the MDP, compute the 'value' of a fixed policy.
  - "How will this plan perform under uncertainty?"
- Reinforcement Learning (today):
  - Limited access to the MDP.
  - "Can I learn to act in an uncertain world?"

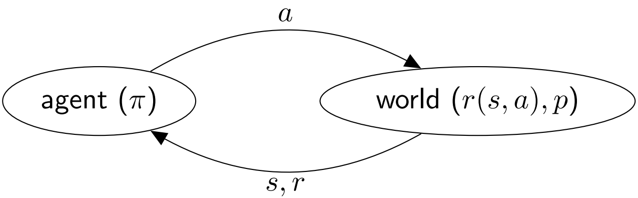
Working with MDPs

An MDP is a general probabilistic framework, and can be utilized in many different scenarios.

- Planning:
  - Full access to the MDP, compute an optimal policy.
  - "How do I act in a known world?"
- Policy Evaluation:
  - Full access to the MDP, compute the 'value' of a fixed policy.
  - "How will this plan perform under uncertainty?"
- Reinforcement Learning (today):
  - Limited access to the MDP.
  - "Can I learn to act in an uncertain world?"

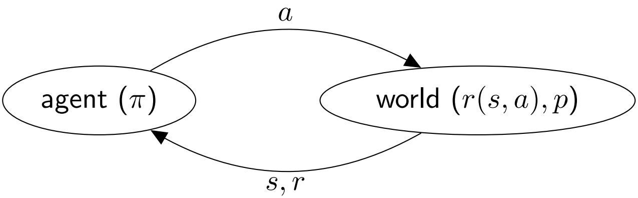
Contents

Reinforcement Learning



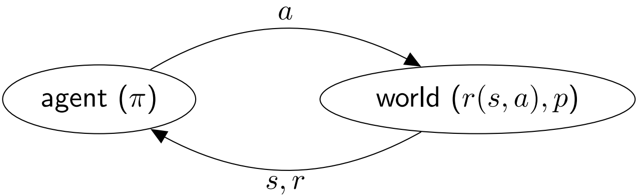
- Agent knows current state s takes actions a, and gets reward r.
- **No access** to reward model $r(s,a)$ or transition model $p(s'|s,a)$, only see outcome reward r and next state s'
- Very challenging problem to learn π while uncertain about model of the world, (contrast with last class).

RL Example: Medical Diagnostics



- States: patient symptoms
- Actions: prescribe drugs, change diet, do nothing, ...
- Reward: $+5$ if health improves, -1 if costly, ...
- Transition model: update of symptoms health based on actions

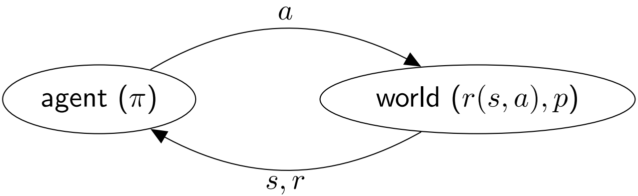
RL Example: Ad Market



- States: current knowledge of user's preferences
- Actions: show particular ad ...
- Reward: +100 if user clicks, -1 if otherwise, ...
- Transition model: user remains on site or leaves

Note: transition model is probabilistic in both planning and RL. The difference is that in planning we **know** the probabilities.

RL Example: Ad Market



- States: current knowledge of user's preferences
- Actions: show particular ad ...
- Reward: $+100$ if user clicks, -1 if otherwise, ...
- Transition model: user remains on site or leaves

Note: transition model is probabilistic in both planning and RL. The difference is that in planning we **know** the probabilities.

Types of RL

- Model-Based RL:
  - Estimate world models $r(s, a; \mathbf{w})$ and $p(s'|s, a; \mathbf{w})$.
  - Utilize planning (value or policy iteration) to develop policy π.
- Model-Free (our focus):
  - Directly learn the policy π from samples of the world.

When might you prefer one over the other?

Types of RL

- Model-Based RL:
  - Estimate world models $r(s, a; \mathbf{w})$ and $p(s'|s, a; \mathbf{w})$.
  - Utilize planning (value or policy iteration) to develop policy π.
- Model-Free (our focus):
  - Directly learn the policy π from samples of the world.

When might you prefer one over the other?

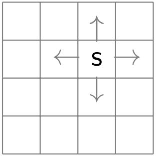
Types of RL

- Model-Based RL:
  - Estimate world models $r(s, a; \mathbf{w})$ and $p(s'|s, a; \mathbf{w})$.
  - Utilize planning (value or policy iteration) to develop policy π.
- Model-Free (our focus):
  - Directly learn the policy π from samples of the world.

When might you prefer one over the other?

Reinforcement Learning Setup

Learning is performed **online**, we learn as we interact with the world.



Contrast with supervised learning:

- No train/test, reward accumulated over interactions.
- Not learning from fixed data, more information acquired as we go.
- Able to influence the training distribution by action decisions.

High-Level Challenges of RL

1. **Exploration/Exploitation**: Trade-off between taking actions with high expected future reward [exploitation], and taking less explored actions to improve estimation [exploration].

2. **Asynchronous Samples**: In previous approaches we had a fixed set of samples, in RL samples come in on the fly based on interaction with the world.

High-Level Challenges of RL

1. **Exploration/Exploitation**: Trade-off between taking actions with high expected future reward [exploitation], and taking less explored actions to improve estimation [exploration].
2. **Asynchronous Samples**: In previous approaches we had a fixed set of samples, in RL samples come in on the fly based on interaction with the world.

Contents

Review: Bellman equations

The planning problem for an MDP is:

$$\pi^* \in \arg\max_{\pi} V^{\pi}(s).$$

(exists a solution that is optimal for every state s).

Definition (Bellman equations)

For an optimal policy π^*, we have

$$V^*(s) = \max_{a \in A} \left[r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V^*(s') \right], \quad \forall s \tag{3}$$

Alternate Form: Bellman equations

Alternate form of the Bellman operator using the Q-Function using:

$$\pi^* \in \arg\max_{\pi} Q^{\pi}(s, a).$$

Definition (Bellman equations)

For an optimal policy π^*, we have

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s' \in S} p(s' \mid s, a) \max_{a' \in A} \left[Q^*(s', a') \right], \quad \forall s, a \tag{4}$$

Model-Free Estimation Strategy

Observe:

- $Q^*(s, a)$ is just a function from $S \times A \mapsto \mathbb{R}$
- If we had Q^* then $\pi^*(s) = \arg\max_a Q^\pi(s, a)$

Strategy:

- Learn the value of a Q-function to estimate Q^*
- Use a parameter table, $\mathbf{w} \in \mathbb{R}^{|S||A|}$:

$$Q(s, a; \mathbf{w}) \triangleq w_{s,a}$$

Model-Free Estimation Strategy

Observe:

- $Q^*(s,a)$ is just a function from $S \times A \mapsto \mathbb{R}$
- If we had Q^* then $\pi^*(s) = \arg\max_a Q^\pi(s,a)$

Strategy:

- Learn the value of a Q-function to estimate Q^*
- Use a parameter table, $\mathbf{w} \in \mathbb{R}^{|S||A|}$:

$$Q(s,a;\mathbf{w}) \triangleq w_{s,a}$$