Finite Markov Decision Processes

Summary of Chapter 3

Reference book: Reinforcement Learning, an introduction - 2nd Edition

Book club February 26, 2021

RL fundaments

- Reinforcement learning is about **learning from interaction** how to behave in order **to achieve a goal**;

RL fundaments

- Reinforcement learning is about **learning from interaction** how to behave in order **to achieve a goal**;
- The RL **agent** and its **environment** interact over a sequence of discrete time steps;

RL fundaments

- Reinforcement learning is about **learning from interaction** how to behave in order **to achieve a goal**;
- The RL **agent** and its **environment** interact over a sequence of discrete time steps;
- The agent's goal is to **maximize the amount of reward** it receives over time.

RL elements

How do agent and environment interact define the task:

- the **actions** are the choices made by the agent;

RL elements

How do agent and environment interact define the task:

- the **actions** are the choices made by the agent;
- the **states** are the basis for making the choices;

RL elements

How do agent and environment interact define the task:

- the **actions** are the choices made by the agent;
- the **states** are the basis for making the choices;
- the **rewards** are the basis for evaluating the choices;

RL elements

How do agent and environment interact define the task:

- the **actions** are the choices made by the agent;
- the **states** are the basis for making the choices;
- the **rewards** are the basis for evaluating the choices;
- A **policy** is a stochastic rule by which the agent selects actions as a function of states.

Finite MDPs

A common framework in which much of the RL has been formulated over the years are the **finite Markov decision processes**, but methods can be applied more generally. The basic assumptions are:

Finite MDPs

A common framework in which much of the RL has been formulated over the years are the **finite Markov decision processes**, but methods can be applied more generally. The basic assumptions are:

- **finite** state, action and reward **sets**;

Finite MDPs

A common framework in which much of the RL has been formulated over the years are the **finite Markov decision processes**, but methods can be applied more generally. The basic assumptions are:

- **finite** state, action and reward **sets**;
- the process has **no memory** of past states it was in: the transition probability function p is conditioned only to the last state the system was in;

Returns and episodes

- The **return** is the function of future rewards that the agent seeks to maximize (in expected value). It has several formulation as one could choose to **discount delayed rewards** on not.

Returns and episodes

- The **return** is the function of future rewards that the agent seeks to maximize (in expected value). It has several formulation as one could choose to **discount delayed rewards** on not.
- Tasks could be:
  - **episodic** if the agent–environment interaction breaks naturally into episodes;
  - **continuing** if the agent–environment interaction continues without limits;

Value functions

- Policy's **value functions** are:

Value functions

- Policy's **value functions** are:
  - $v_\pi(s)$, which assigns to each state s the expected return from that state given that the agent uses the policy π;

Value functions

- Policy's **value functions** are:
  - $v_\pi(s)$, which assigns to each state s the expected return from that state given that the agent uses the policy π;
  - $q_\pi(s, a)$, which assigns to each state-action pair (s, a) the expected return from that state-action pair given that the agent uses the policy π;

Value functions

- Policy's **value functions** are:
  - $v_\pi(s)$, which assigns to each state s the expected return from that state given that the agent uses the policy π;
  - $q_\pi(s, a)$, which assigns to each state-action pair (s, a) the expected return from that state-action pair given that the agent uses the policy π;

- The **optimal value functions** assign to each state ($v^*(s)$), or state–action pair ($q^*(s, a)$), the largest expected return achievable by any policy.

Optimality equations

- A policy whose value functions are optimal is an **optimal policy**. There can be many optimal policies, whereas the optimal value function is unique for each MDP;

Optimality equations

- A policy whose value functions are optimal is an **optimal policy**. There can be many optimal policies, whereas the optimal value function is unique for each MDP;
- Any policy that is **greedy** with respect to the optimal value functions must be an optimal policy.

Optimality equations

- A policy whose value functions are optimal is an **optimal policy**. There can be many optimal policies, whereas the optimal value function is unique for each MDP;
- Any policy that is **greedy** with respect to the optimal value functions must be an optimal policy.
- The **Bellman optimality equations** are special consistency conditions that the optimal value functions must satisfy and that can, in principle, be solved for the optimal value functions, from which an optimal policy can be determined.

Constraints

It is usually not possible to simply compute an optimal policy by solving the Bellman optimality equation:

- The agent could have a partial or **incomplete knowledge** of the environment, that is for a MDP a model of the dynamics given by the transition probability function p;

Constraints

It is usually not possible to simply compute an optimal policy by solving the Bellman optimality equation:

- The agent could have a partial or **incomplete knowledge** of the environment, that is for a MDP a model of the dynamics given by the transition probability function p;
- The **memory available** is also an important constraint: the state space could be so large that **approximations** are needed