Reinforcement Learning

Renyi Qu

2022-12-02

Contents

1 Prerequisites **5**

2 Literature **7**

3 Methods **9**

3.1 math example 9

4 Applications **11**

4.1 Example one 11

4.2 Example two 11

5 Final Words **13**

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.name/tinytex/.

Specification: - State space: $\mathcal{S}$ - Action space: $\mathcal{A}$ - Transition probability (i.e., model): $p(s'|s,a) = P(S_{t+1} = s'|S_t = s, A_t = a) = \sum_{r\in\mathcal{R}} p(s',r|s,a)$ - Reward: $r(s,a,s') = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s'] = \frac{\sum_{r\in\mathcal{R}} r p(s',r|s,a)}{\sum_{r\in\mathcal{R}} p(s',r|s,a)}$ - Discount factor: $\gamma \in [0,1]$

Goal: Find optimal policy $a_t = \pi(s_t)$ to maximize long-term reward:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Policy: - Bellman's Equation ($\forall \pi(a|s)$, including stochastic)

$$V_\pi(s) = E_\pi[G_t|s_t = s] = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V_\pi(s')], \forall s \in \mathcal{S}$$

$$Q_\pi(s,a) = E_\pi[G_t|s_t = s, a_t = a] = \sum_{s',r} p(s',r|s,a)[r + \gamma V_\pi(s')], \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$$

- Bellman's Optimality Equation ($\forall \pi_*(s)$, only deterministic)

$$V_*(s) = \max_{a \in \mathcal{A}(s)} \sum_{s',r} p(s', r|s, a)[r + \gamma V_*(s')], \forall s \in \mathcal{S}$$

$$Q_*(s, a) = \sum_{s',r} p(s', r|s, a)[r + \gamma \max_{a'} Q_*(s')], \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$$

Dynamic Programming: - Policy Evaluation

```python
delta = 0
while delta >= epsilon:
    for s in S:
        v_prev <- V(s)
        V(s)   <- Bellman's Equation
        delta  <- max(delta, abs(v_prev-V(s)))
```

Chapter 2

Literature

Here is a review of existing methods.

Chapter 3

Methods

We describe our methods in this chapter.

Math can be added in body using usual syntax like this

3.1 math example

p is unknown but expected to be around 1/3. Standard error will be approximated

$$SE = \sqrt(\frac{p(1-p)}{n}) \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

You can also use math in footnotes like this[1].

We will approximate standard error to 0.027[2]

[1] where we mention $p = \frac{a}{b}$

[2] p is unknown but expected to be around 1/3. Standard error will be approximated

$$SE = \sqrt(\frac{p(1-p)}{n}) \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

Chapter 4

Applications

Some *significant* applications are demonstrated in this chapter.

4.1 Example one

4.2 Example two

Chapter 5

Final Words

We have finished a nice book.