CNRS - Toulouse INP - UT3 - UT Capitole - UT2

Institut de Recherche en Informatique de Toulouse

Lecture 5: Temporal-Difference Learning

N8EN18B - Contrôle et Apprentissage

Guilherme IECKER RICARDO

IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3

Contents

1 On-Policy Control: SARSA

2 Off-Policy Control: q-Learning

3 Deep Reinforcement Learning

4 Final Exam

Recap – Monte-Carlo Learning

Access the Python Notebook:
https://guilhermeir.github.io/teaching/rl/mc.ipyng

Recap – TD Learning: Prediction

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\mathit{terminal}) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

Recap – Bellman Equations

- Bellman Expectation Equation

$$
\begin{aligned}
v_\pi(s) &= \sum_{a \in \mathcal{A}} \pi(a|\, s) \cdot \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \cdot v_\pi(s') \right) \\
q_\pi(s, a) &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|\, s') \cdot q_\pi(s', a')
\end{aligned} \tag{1}
$$

- Bellman Optimality Equation

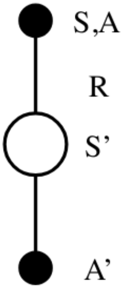
$$
\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}} \left\{ \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \cdot v_*(s') \right\} \\
q_*(s, a) &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \cdot \max_{a' \in \mathcal{A}} \{ q_*(s', a') \}
\end{aligned} \tag{2}
$$

On-Policy Control: SARSA

MC vs. TD Control

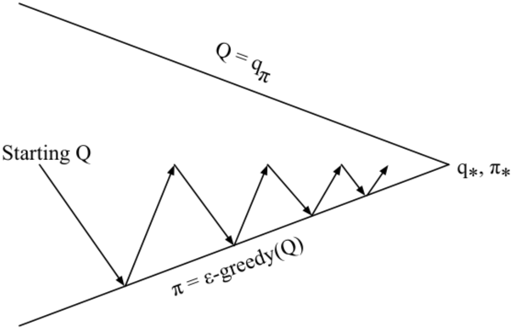
- Temporal-Difference (TD) Learning has several advantages over Monte-Carlo (MC) Learning
  - Lower variance
  - Online
  - Incomplete sequences
- Natural idea: use TD instead of MC in our control loop (value iteration), i.e.,
  - Apply TD to $Q(S, A)$
  - Use ϵ-Greedy policy improvement
  - Update every time-step

Updating Action-Value Functions with SARSA



$$Q(S,A) \leftarrow Q(S,A) + \alpha(R + \gamma Q(S',A') - Q(S,A))$$

On-Policy Control with SARSA



Every time-step:
Policy evaluation: SARSA, $Q \approx q_\pi$
Policy improvement: ϵ-Greedy Policy Improvement

SARSA Algorithm for On-Policy Control

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Off-Policy Control: q-Learning

Off-Policy Learning

- Evaluate target policy $\pi(a|s)$ to compute $v_\pi(s)$ or $q_\pi(s,a)$
- While following behavior policy $\mu(a|s)$

$$\{S_1, A_1, R_2, \ldots, S_T\} \sim \mu$$

- Why is this important?
  - Learn from observing humans or other agents
  - Re-use experience generated from old policies $\pi_1, \pi_2, \ldots, \pi_{t-1}$
  - Learn about optimal policy while following exploratory policy
  - Learn about multiple policies while following one policy

Q-Learning

- We now consider off-policy learning of action-values $Q(s, a)$
- Next action is chosen using behavior policy $A_{t+1} \sim \mu(\cdot|S_t)$
- But we consider alternative successor action $A' \sim \pi(\cdot|S_t)$
- And update $Q(S_t, A_t)$ towards value of alternative action

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A') - Q(S_t, A_t))$$

Off-Policy Control with Q-Learning

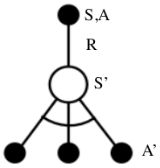
- We now allow both behavior and target policies to **improve**
- The target policy π is **greedy** w.r.t. $Q(s, a)$

$$S(S_{t+1} = \text{arg}\max_{a'} Q(S_{t+1,a'} \tag{3}$$

- The behavior policy μ is, e.g., **ϵ-greedy** w.r.t. $Q(s, a)$
- The Q-Learning target then simplifies:

$$\begin{aligned} & R_{t+1} + \gamma Q(S_{t+1}, A') \\ =& R_{t+1} + \gamma Q(S_{t+1}, \text{arg}\max_{a'} Q(S_{t+1,a'})) \\ =& R_{t+1} + \max_{a'} \gamma Q(S_{t+1}, a') \end{aligned} \tag{4}$$

Q-Learning Control Algorithm



$$Q(S,A) \leftarrow Q(S,A) + \alpha(R + \gamma \max_{a'} Q(S',a') - Q(S,A))$$

Theorem

Q-Learning control converges to the optimal action-value function, $Q(s,a) \rightarrow q_*(s,a)$

Q-Learning Algorithm for Off-Policy Control

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
        $S \leftarrow S'$
    until S is terminal

Deep Reinforcement Learning

Deep Reinforcement Learning

Additional set of slides

Final Exam

How to study?

- Class slides (available here)
- Reading (text-book available here)
  - Chapter 1: all sections
  - Chapter 3: all sections
  - Chapter 4: all sections except 4.5
  - Chapter 5: sections 5.1 - 5.4
  - Chapter 6: sections 6.1 - 6.5
- Studying examples and solving problems from text-book (included in the sections above)
  - Here you have the code for all examples in the book
  - Here you have all solutions for the book's questions
- This list of exercises.

Rules

- 1-hour long final exam
- There are 20 points + 5 bonus points, i.e., your **actual grade** $\in [0, 25]$
- Your **final grade** is $\min(20, \textbf{actual grade})$
- Only 1 page (1 side of a sheet of paper) of personal notes is allowed
- No devices, books, etc.
- Zero cheating tolerance!