LG Advanced Data Scientists Program
Deep Learning

[9: Reinforcement Learning (Part 2)]

Prof. Sungroh Yoon

Electrical & Computer Engineering | Seoul National University

(last compiled at 15:36:00 on 2020/02/27)

Outline

References

- books/papers:
  - Reinforcement Learning (2nd edition)[1] ▸ Link
  - Artificial Intelligence: A Modern Approach[2]
  - A brief survey of deep reinforcement learning[3]
- online resources:
  - Silver UCL class ▸ Link & ICML tutorial ▸ Link
  - Schulman MLSS tutorial ▸ Link
  - Abbeel & Schulman NIPS tutorial ▸ Link
  - UC Berkeley CS188 (AI) ▸ Link & CS294 (DRL) ▸ Link
  - Stanford CS234 (RL) ▸ Link

---

[1] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press

[2] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson Education Limited

[3] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*

Outline

Value-based methods

- estimate optimal value function $\Rightarrow$ derive optimal policy π^* therefrom
- learning = changing values of states we visit
  - for more accurate value estimation (*e.g.* winning probabilities)
- to do this: we "________" the value of
  - s' : state after each move to
  - s : state before the move

$$
\begin{array}{ccc}
s & \xrightarrow{\text{state: move}} & s' \\
V(s) & \xleftarrow[\text{value: backup}]{} & V(s')
\end{array}
$$

i.e. current value of earlier state s:

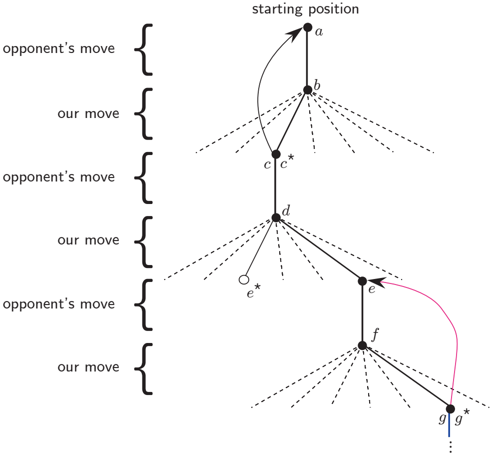
- ▷ adjusted to be closer to value of later state s'

- learning involves a lot of backup operations

Example: a sequence of moves in a two-player game

- solid lines:
  - moves taken during a game
- dashed lines:
  - moves considered but not taken
  - discarded by "exploitation"
- exploratory moves

  e.g. our second move
  - taken even if another sibling move (leading to e^*) was better
  - "exploration"
- curved arrows
  - backups ⇒ ______



(source: [Sutton and Barto, 2018][4])

[4] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press

Backup operations

- transfer value information *back*
  - to a state from its successor states

    or
  - to a state-action pair from its successor state-action pairs
- that is, "backup" refers to "______" of values
- backups are at the heart of RL methods

Three ways to do backup

1. **full** backup by *dynamic programming* (DP)

$$V(s) \leftarrow \mathbb{E}\left[r + \gamma V(s')\right]$$

2. **sample** backup by *Monte Carlo* (MC) learning

$$V(s) \leftarrow V(s) + \alpha\left[R - V(s)\right]$$

3. **sample** backup by *temporal-difference* (TD) learning

$$V(s) \leftarrow V(s) + \alpha\left[r + \gamma V(s') - V(s)\right]$$

- R : sample return (actual return from a trajectory)
- α : step-size parameter
  - a small positive fraction that influences ____________

more on way #3:

- use a simple rule to update $V(s)$

$$\begin{aligned} & V(s) \leftarrow V(s) + \alpha \left[r + \gamma V(s') - V(s) \right] \\ \Longleftrightarrow\ & V(s) \leftarrow \underbrace{(1-\alpha)}_{\substack{\text{weight on}\\ \text{old value}}} V(s) + \underbrace{\alpha}_{\substack{\text{weight on}\\ \text{new value}}} \left[r + \gamma V(s') \right] \end{aligned} \tag{1}$$

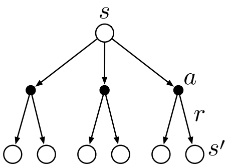
- update rule (1): an example of *temporal-difference* (TD) learning
  - changes are based on $\underbrace{r + \gamma V(s') - V(s)}_{\uparrow}$

difference between estimates at two different times

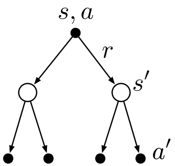
Backup diagram

- depict relationships that form the basis of ______ operations

  e.g. for dynamic programming to compute $V(s)$ and $Q(s, a)$:



$V(s) \leftarrow \mathbb{E}[r + \gamma V(s')]$



$Q(s, a) \leftarrow \mathbb{E}[r + \gamma Q(s', a')]$

- notations
  - open circle: a state
  - solid circle: a state-action pair

Taxonomy of value-based methods

two kinds of defining characteristics:

- if we bootstrap
  - we update estimates based on other ________ (not true target)
- if we sample
  - we do not compute but just sample an expectation

	sample backup	**full backup**
bootstrap (shallow backup)	temporal-difference (TD) learning	dynamic programming (DP)
no bootstrap (deep backup)	Monte Carlo (MC) learning	exhaustive search

example: sample-backup methods

- **Monte-Carlo** (MC) learning
  - go all the way to ___ of a trajectory and
  - estimate the value just by looking at sample return

  $\Rightarrow$ no bootstrapping

- **temporal-difference** (TD) learning[5]
  - just look one step ahead and
  - estimate the value after one step using one-step lookahead value estimate

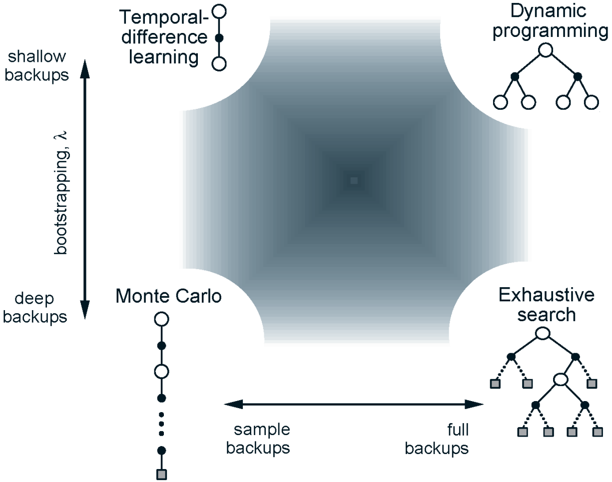
  $\Rightarrow$ bootstrapping

- TD(λ): generalize/unify[6]
  - use arbitrary # of lookaheads

[5] more precisely, one-step TD or TD(0)

[6] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press

Unified view of RL



(source: [Sutton and Barto, 2018][7])

[7] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press

Outline

Summary

- value-based reinforcement learning methods
  - estimate optimal value function $V^*(s)$ or $Q^*(s, a)$
  - $\Rightarrow$ then find optimal policy π^* therefrom
  - key operation: backup (= update of $V(s)$ using $V(s'))$
  - defining characteristic #1: sample vs full backup
  - defining characteristic #2: shallow (=bootstrap) vs deep backup

- tabular methods: represent value function by lookup table
  - dynamic programming: full + shallow backup
    - value iteration and policy iteration
  - temporal-difference (TD) learning: sample + shallow backup
    - Q-learning (off-policy) and SARSA (on-policy)
  - Monte Carlo (MC) learning: sample + deep backup

- value function approximation by deep neural net
  - deep Q-network (DQN): experience replay with fixed Q-learning target