Reinforcement Learning

—

Exercise 4: Model-free Prediction

Nico Meyer

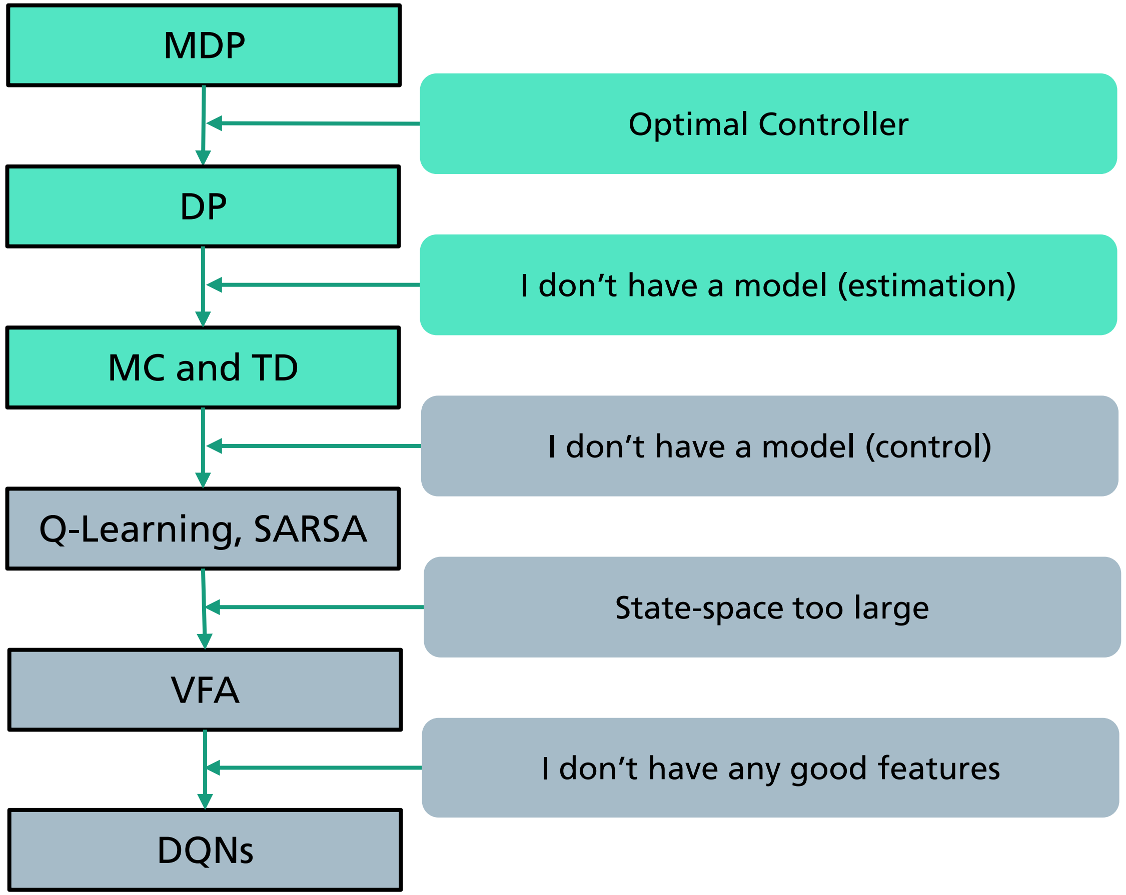
Fraunhofer IIS

Fraunhofer-Institut für Integrierte Schaltungen IIS

Overview
Exercise Content

Week	*Date*	*Topic*	*Material*	*Who?*
1	22.04.		*no exercises*	
2	29.04.	MDPs (slides)	ex1.pdf	Nico
3	06.05.	T.B.D.		
4	13.05.	Dynamic Programming (slides)	ex2.pdf, ex2_skeleton.zip	Alex
5	20.05.	OpenAI Gym, PyTorch-Intro (slides) TD-Learning (slides)		Nico
6	27.05.	TD-Control (slides)		Nico
7	03.06.	**Intermediate exam**		
8	*10.06.*		*no exercises*	
9	17.06.	DQN (slides)		Nico
10	24.06.	VPG (slides)		Alex
11	01.07.	A2C (slides)		Nico
12	08.07.	Multi-armed Bandits (slides)		Alex
13	15.07.	RND/ICM (slides)		Alex
14	22.07.	MCTS (slides)		Alex

Overview
Overall Picture



Recap

Model-free Prediction

Recap
Monte Carlo and TD Methods

- So far: We know our MDP model $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$.
  - Planning by using dynamic programming
  - Solve a known MDP

- What if we don’t know the model, i.e., $\mathcal{P}$ or $\mathcal{R}$ or both?

- We distinguish between 2 problems for unknown MDPs:

  - **Model-free Prediction:** Evaluate the future, given the policy π.
    (estimate the value function)

  - **Model-free Control:** Optimize the future by finding the best policy π.
    (optimize the value function)

Recap

Monte Carlo Policy Evaluation

- MC Policy Evaluation
  - MC methods learn from episodes of experience under policy π:

$$s_t, a_t, r_t, s_{t+1}, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T \sim \pi$$

  - To evaluate a state $s \in \mathcal{S}$ we keep track of the rewards received from that state onwards.

- First-Visit Monte-Carlo Policy Evaluation:
  - First time-step t that state s is visited in an episode
    - Increment counter $N(s) \leftarrow N(s) + 1$,
    - Increment total return $S(s) \leftarrow S(s) + G_t$,
    - Value is estimated by mean return: $V(s) = S(s)/N(s)$
  - Our estimation $V(s)$ will come close to $V^\pi(s)$ as $N(s) \rightarrow \infty$.
    (considering the law of large numbers)

Recap

Temporal Difference Policy Evaluation

- Temporal-Difference Learning
  - Breaks up episodes and makes use of the intermediate returns
  - Learns from incomplete episodes (bootstrapping)
  - **We update a guess towards a guess**

$$V^{\pi}(s) = \underbrace{r(s,\pi(s))} + \underbrace{\gamma \sum_{s' \in S} \boxed{\mathcal{P}(s'|s,\pi(s))}\, V^{\pi}(s')}$$

We don't know the transition model

$$\boxed{(s,a,r,s')}$$

But we have real transitions available

$$\boxed{V^{\pi}(s) = r + \gamma V^{\pi}(s')}$$

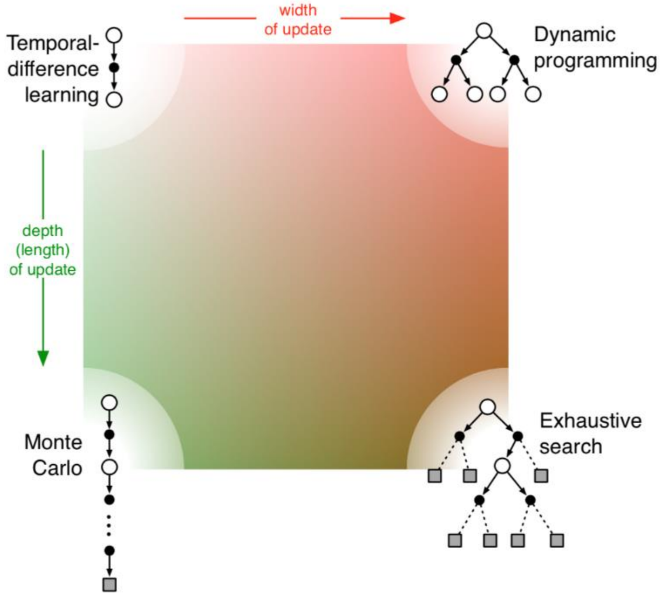
Let's assume that the reality is the transition we observed

$$\boxed{V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s))}$$

→ and update our old estimate "a bit" in this direction

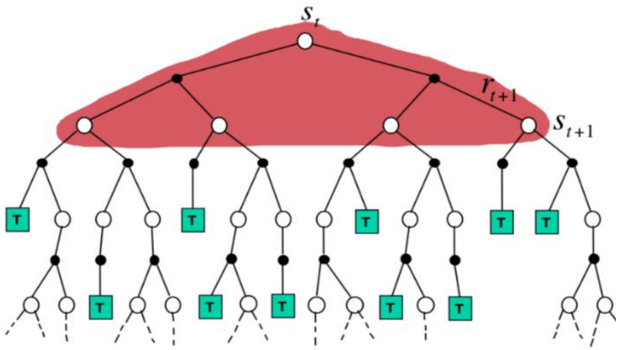
Recap

DP vs. MC vs. TD



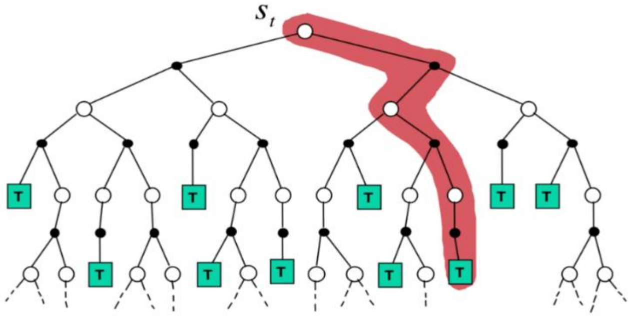
<u>DP Backup</u>

$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1})\right]$$



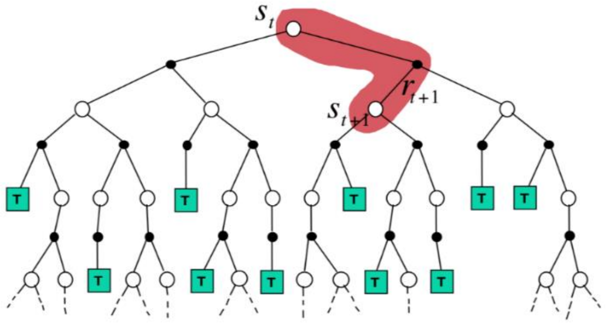
<u>MC Backup</u>

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t - V(S_t)\right)$$



<u>TD Backup</u>

$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$$

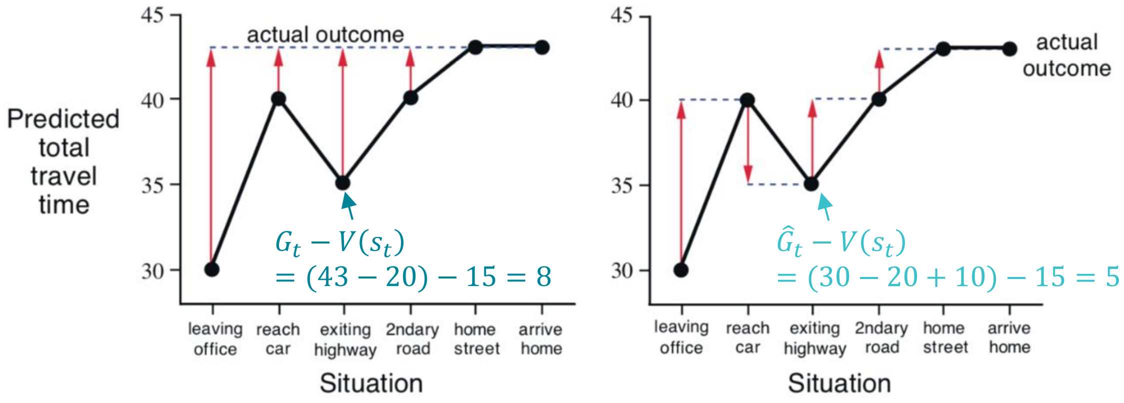


Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Recap

TD and MC Algorithms

MC ($\alpha = 1$)



TD ($\alpha = 1$)

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

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 $V(terminal) = 0$

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

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
    $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
    $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
    Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
            Append G to $Returns(S_t)$
            $V(S_t) \leftarrow \text{average}(Returns(S_t))$

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Recap

Advantages and Disadvantages of MC and TD

- Which one should I use? Does it make any difference?
  - Bias/Variance Trade-Off
  - MC has high variance, but zero bias
    - good convergence (even with FA)
    - insensitive to initialization (no bootstrapping), simple to understand
    - only works for episodic problems (must wait until end of episode for update)
    - more efficient in non-Markov environments
  - TD has low variance, but some bias
    - TD(0) converges to $\pi_v(s)$ (be careful with FA: bias is a risk)
    - sensitive to initialization (because of the bootstrapping)
    - update after each step
    - exploits Markov property and is more efficient in Markov environment
    - **usually more efficient in practice**

Exercise Sheet 4

Model-free Prediction

Fraunhofer IIS

Fraunhofer-Institut für Integrierte Schaltungen IIS

Thank you for your attention!