

Approximate value functions

From tabular to function approximation

Cathy Wu

6.7920 Reinforcement Learning: Foundations and Methods

Readings

1. *DPOC vol2 §2.5.3*
2. Neuro-Dynamic Programming (NDP). §3.1-3.2.
3. [SB 16.5](#)

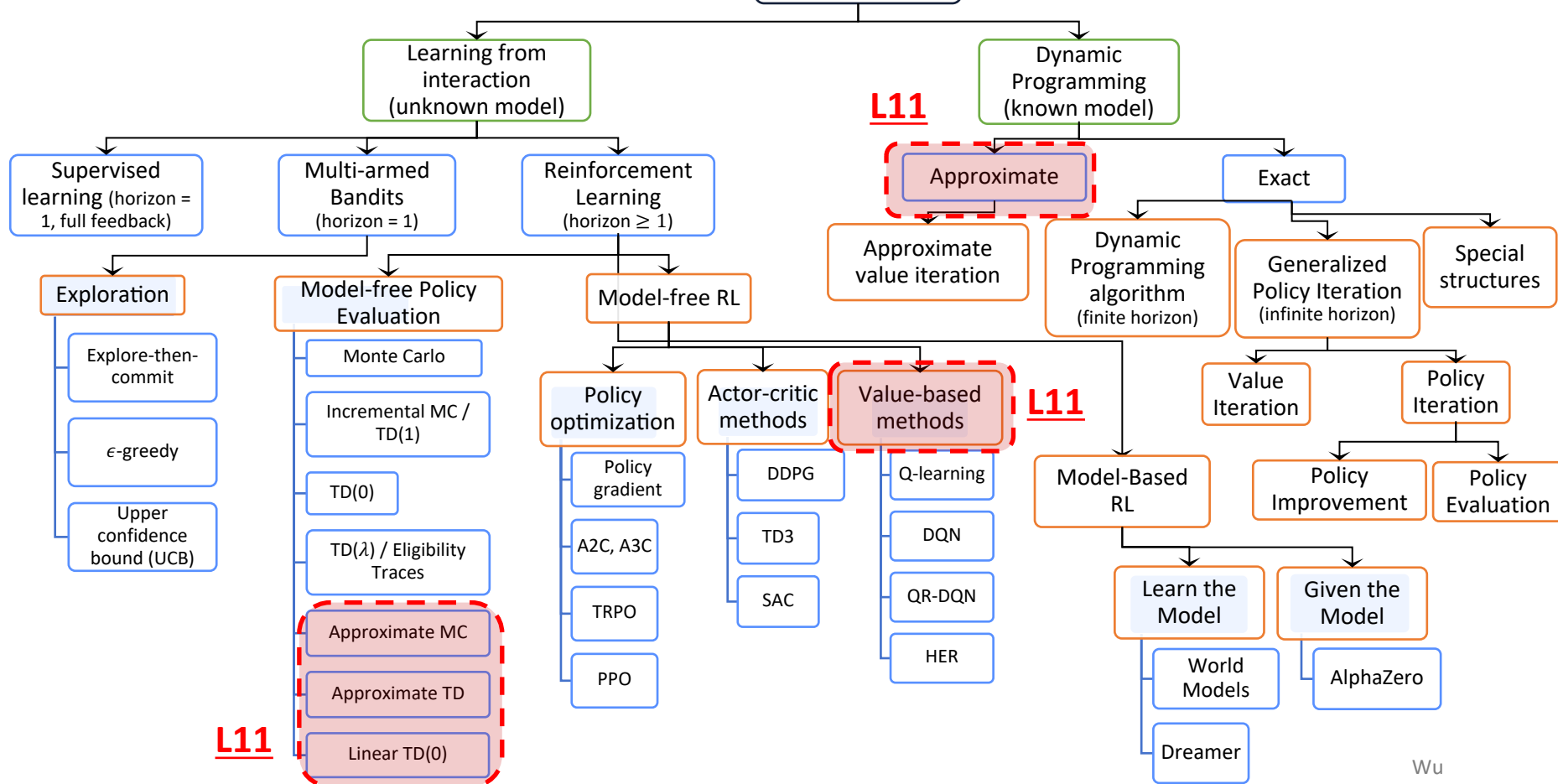
Outline

1. Introduction to value function approximation
2. Approximate value-based algorithms

Taxonomy of RL Methods

Methods for
sequential
decision making

Reinforcement learning (RL) is a framework for
sequential decision making under uncertainty.



Outline

- 1. Introduction to value function approximation**
 - a. Why approximate?
 - b. Linear vs nonlinear function approximation
2. Approximate value-based algorithms

Notice

Reminder: We are typically working in the episodic discounted setting.

Most results smoothly extend to other settings.

NEW: The value functions **cannot** be represented **exactly** (as in the tabular setting).

Recall (L8): Key challenge of huge decision spaces

Arcade Learning Environment (ALE)

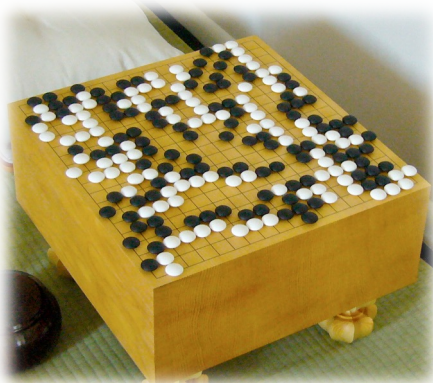


Possible game states: $3^{84 \times 84} \approx 10^{3366}$

For reference:

There are between 10^{78} to 10^{82}
atoms in the observable universe.

Game of Go



t

Possible game states: $3^{19 \times 19} \approx 10^{172}$

Cannot only explore. Cannot only exploit.
Must trade off exploration and exploitation.

Approximating the (state-action) value function

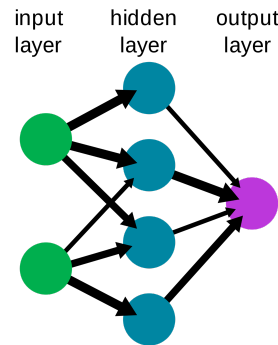
$Q(s, a)$

	2.5	1.4	3.2	5.4
	1.0	3.2	5.1	6.3
	5.2	4.2	5.5	7.2
	8.7	3.4	2.0	8.0
	4.8	2.5	3.5	4.2
s	1.0	3.0	3.3	1.2
	-180	-172	-99.7	-150
	4.2	2.1	3.2	3.7
	2.1	2.0	3.7	3.1
	3.0	1.2	3.2	2.7
	0.1	1.5	0.1	1.0
	↑	↓	←	→
		a		

“Tabular”

$$Q_{\theta}(s, a) = f_{\theta}(s, a)$$

A simple neural network



“Function approximation”

Value Function Approximation

Desiderata

- **Expressiveness**: Rich enough approximation architecture to provide close enough approximation to the function we are trying to approximate.
- **Computational considerations**: Effective algorithms for tuning the parameters of the approximation architecture (“training”).

Example: Optimal replacement problem

State: level of wear of an object (e.g., a car).

Action: $\{(R)\text{eplace}, (K)\text{eep}\}$.

Cost:

- ▶ $c(x, R) = C$
- ▶ $c(x, K) = c(x)$ maintenance plus extra costs.

Dynamics:

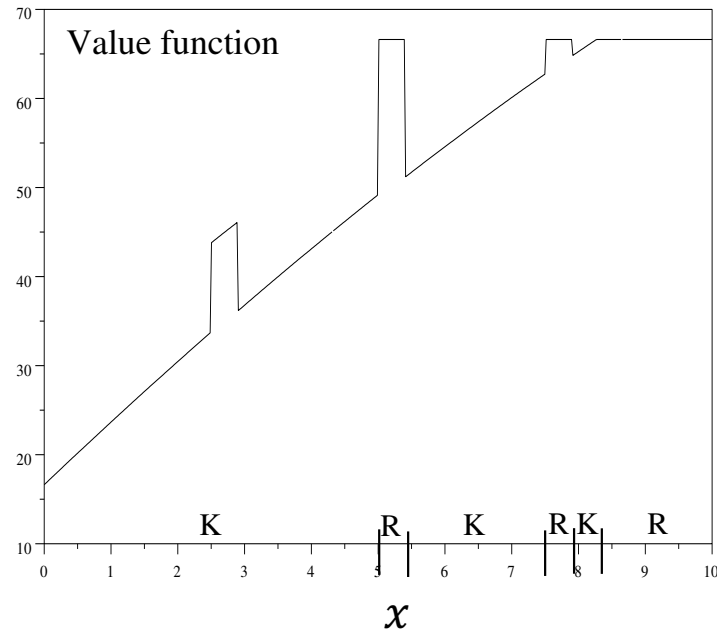
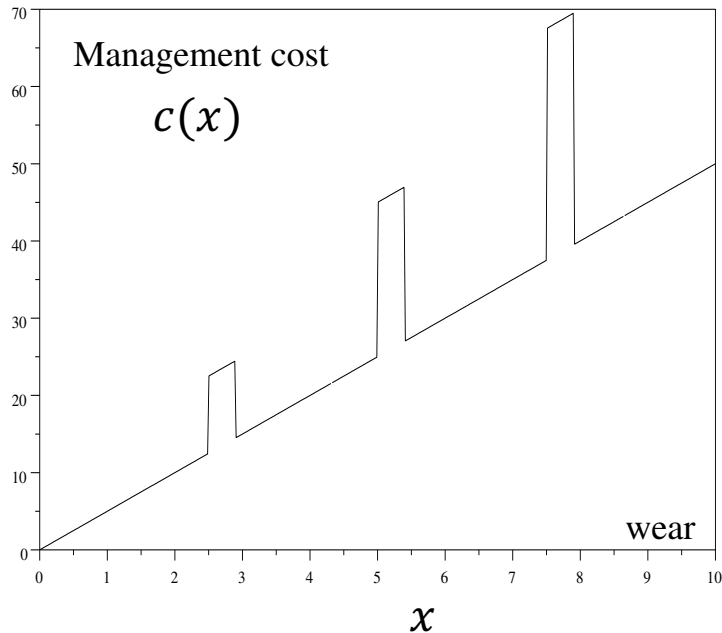
- ▶ $p(\cdot|x, R) \sim \exp(\beta)$ with density $d(y) = \beta \exp^{-\beta y} \mathbb{I}\{y \geq 0\}$,
- ▶ $p(\cdot|x, K) \sim x + \exp(\beta)$ with density $d(y - x)$.

Problem: Minimize the discounted expected cost over an infinite horizon.

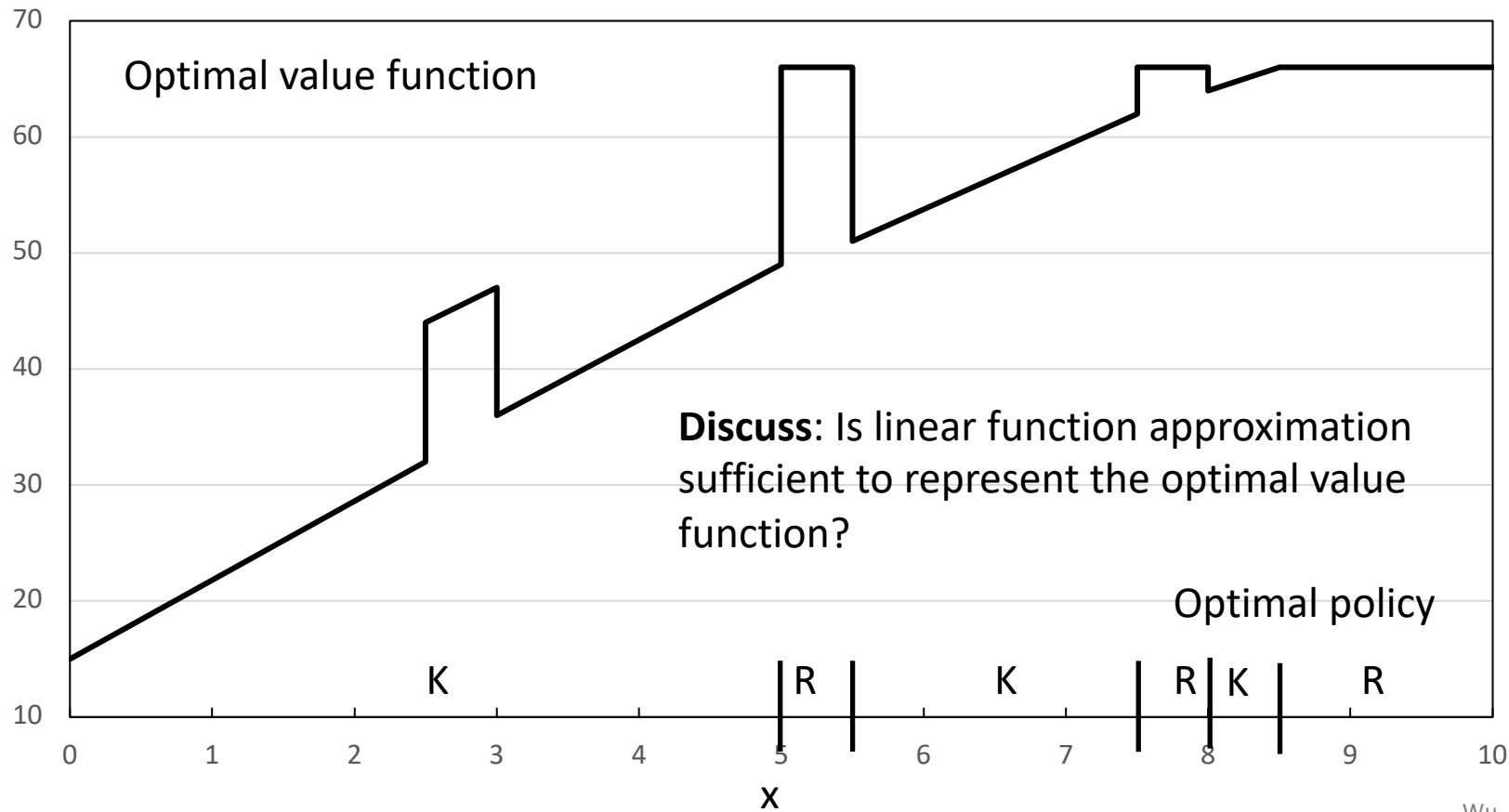
Optimal replacement problem

Optimal value function

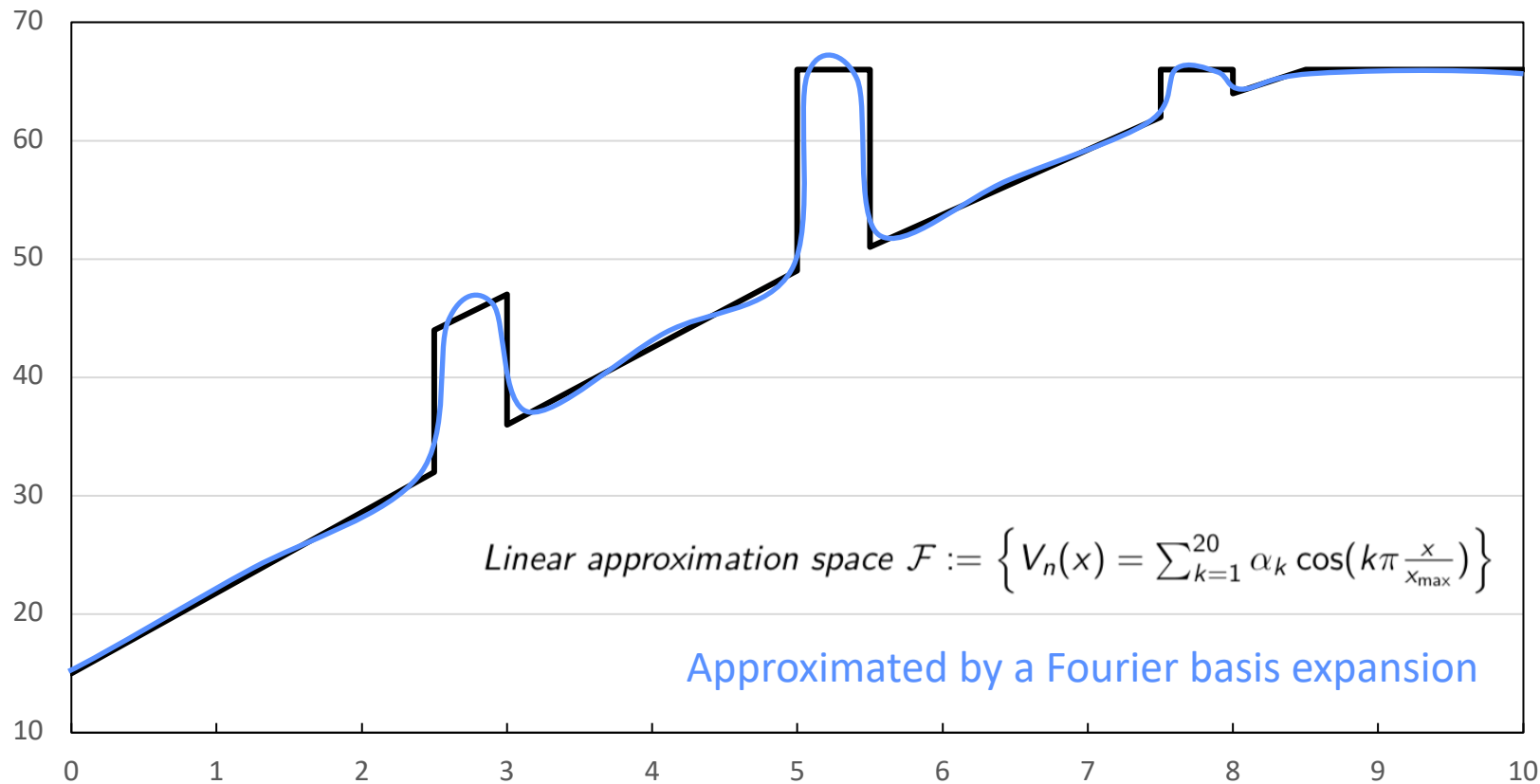
$$V^*(x) = \min \left\{ c(x) + \gamma \int_0^\infty d(y-x) V^*(y) dy, C + \gamma \int_0^\infty d(y) V^*(y) dy \right\}$$



From Exact to Approximate RL



From Exact to Approximate RL



Preview: approximate value iteration

Collect N sample on a uniform grid.

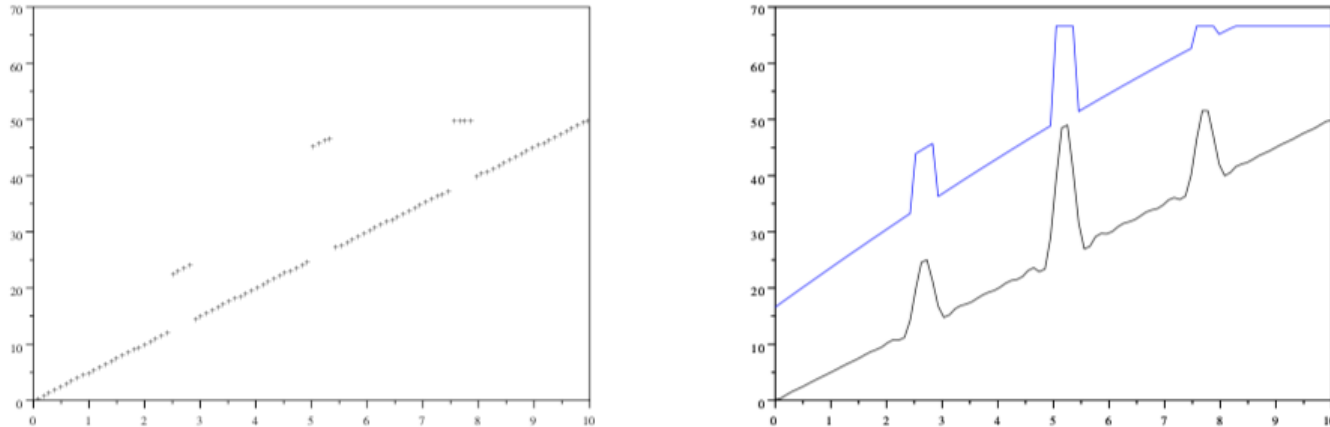


Figure: Left: the *target* values computed as $\{\mathcal{T}V_0(x_n)\}_{1 \leq n \leq N}$. Right: the approximation $V_1 \in \mathcal{F}$ of the target function $\mathcal{T}V_0$.

Preview: approximate value iteration

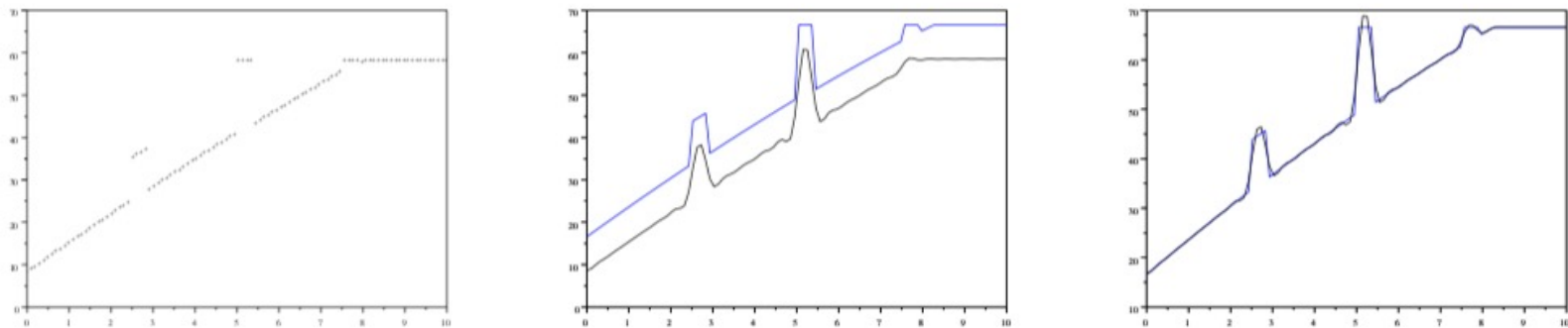


Figure: Left: the *target* values computed as $\{\mathcal{T}V_1(x_n)\}_{1 \leq n \leq N}$. Center: the approximation $V_2 \in \mathcal{F}$ of $\mathcal{T}V_1$. Right: the approximation $V_n \in \mathcal{F}$ after n iterations.

Outline

1. Introduction to value function approximation
2. **Approximate value-based algorithms**
 - a. **Policy evaluation**
 - Approximate Monte Carlo
 - Approximate TD(0)
 - Convergence results: LSTD & Linear TD
 - Divergence counterexample
 - b. Policy learning (next time)

Recall: earlier progression

- Policy evaluation
 - Monte Carlo
 - TD(1), i.e. Incremental Monte Carlo
 - TD(0)
 - TD(λ)
- Policy learning
 - Q-learning
 - SARSA

Consider: How to incorporate function approximation into these?

Policy Evaluation

Fixed policy π

For $i = 1, \dots, n$

1. Set $t = 0$
2. Set initial state $s_0 \sim \mathcal{D}$ [initial state distribution]
3. **While** ($s_{t,i}$ not terminal) [execute one trajectory]
 1. Take action $a_{t,i} = \pi(s_{t,i})$
 2. Observe **next state** $s_{t+1,i}$ and **reward** $r_{t,i} = r(s_{t,i}, a_{t,i})$
 3. Set $t = t + 1$

EndWhile

EndFor

Return: Estimate of the value function $\hat{V}^\pi(\cdot)$

A Gentle Start: Supervised Learning

Linear space to approximate value functions

$$\mathcal{F} = \left\{ V_{\theta}(s) = \sum_{j=1}^d \theta_j \varphi_j(s), \theta \in \mathbb{R}^d \right\}$$

With features

$$\varphi_j: S \rightarrow [0, L] \quad \phi(s) = [\varphi_1(s) \dots \varphi_d(s)]^T$$

Approximate Monte-Carlo as Supervised Learning

- Distribution over initial states $\mathcal{D}$
- Function approximation $V_\theta: S \rightarrow \mathbb{R}, \theta \in \mathbb{R}^d$ [e.g. linear, deepNet]
- Build training set of n samples

$$s_i \sim \mathcal{D} \quad R_i = \sum_{t=0}^H r_{t,i} = V^\pi(s_i) + \epsilon_i \quad (\mathbb{E}[\epsilon_i] = 0)$$

Approximate Monte-Carlo as Supervised Learning

- Training (batch)

$$\hat{\theta}_n = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n L(s_i, R_i; \theta) = \frac{1}{2n} \sum_{i=1}^n (V_{\theta}(s_i) - R_i)^2$$

- Testing (aka generalization error)

$$L(\hat{\theta}_n) = \mathbb{E}_{\mathcal{D}} \left[\frac{1}{2} \left(V^{\pi}(s) - V_{\hat{\theta}_n}(s) \right)^2 \right]$$

Approximate Monte-Carlo as Supervised Learning

Proposition (qualitative)

Let n be the number of samples used to build the Monte-Carlo training set. Let also $r(s, a) \in [0, r_{\max}]$ and trajectories to be as long as $H = \frac{1}{1-\gamma}$, then approximate Monte-Carlo has a generalization error:

$$L(\hat{\theta}_n) \leq \min_{\theta} L(\theta) + O\left(\frac{1}{1-\gamma} \sqrt{\frac{d}{n}}\right)$$

👉 Tends to the best possible approximation as n tends to infinity.

👎 Variance may be big.

■ Proof: Apply Hoeffding's inequality

d = number of features
 θ = parameterizes V_{θ}

Approximate Monte-Carlo as Supervised Learning

- Monte-Carlo with online training after each sample (s_i, R_i) with learning rate α_i

$$\begin{aligned}\hat{\theta}_{i+1} &= \hat{\theta}_i - \alpha_i \nabla_{\theta} L(s_i, R_i; \theta_i) \\ &= \hat{\theta}_i - \alpha_i (V_{\theta_i}(s_i) - R_i) \nabla_{\theta} V_{\theta_i}(s_i)\end{aligned}$$

Outline

1. Introduction to value function approximation
2. **Approximate value-based algorithms**
 - a. **Policy evaluation**
 - Approximate Monte Carlo
 - **Approximate TD(0)**
 - Convergence results: LSTD & Linear TD
 - Divergence counterexample
 - b. Policy learning

Approximate TD(0) as Pseudo-Gradient Descent

- Run π to generate a single trajectory $(s_0, r_0, s_1, r_1, s_2, r_2, \dots, s_n, r_n)$

- TD loss using bootstrapped target

$$\tilde{L}(s_t, \tilde{R}_t; \theta) = \frac{1}{2} (V_\theta(s_t) - \tilde{R}_t)^2 = \frac{1}{2} (V_\theta(s_t) - r_t - \gamma V_{\theta_t}(s_{t+1}))^2$$

- TD online update with learning rate α_t

$$\begin{aligned} \hat{\theta}_{t+1} &= \hat{\theta}_t - \alpha_t \nabla_\theta \tilde{L}(s_t, \tilde{R}_t; \hat{\theta}_t) \\ &= \hat{\theta}_t - \alpha_t (V_{\hat{\theta}_t}(s_t) - r_t - \gamma V_{\hat{\theta}_t}(s_{t+1})) \nabla_\theta V_\theta(s_t; \hat{\theta}_t) \end{aligned}$$

👉 Not really a gradient method...

- Discuss:** Why not?

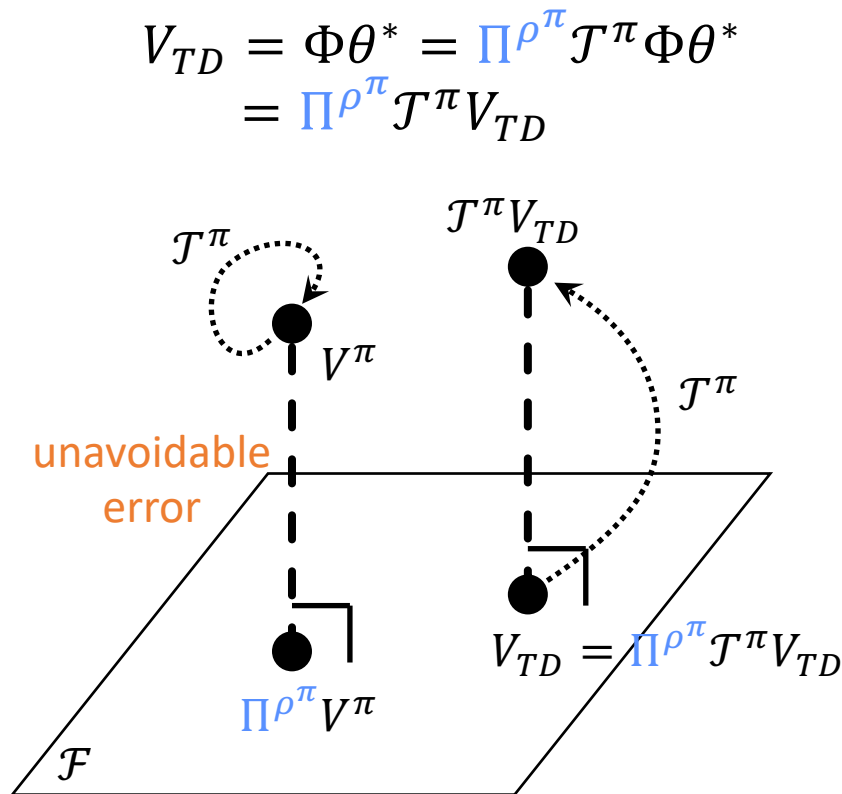
Outline

1. Introduction to value function approximation
2. **Approximate value-based algorithms**
 - a. **Policy evaluation**
 - Approximate Monte Carlo
 - Approximate TD(0)
 - **Convergence results: LSTD & Linear TD**
 - Divergence counterexample
 - b. Policy learning

Least Squares TD, i.e. LSTD

- **Projection perspective** (assume: stationary distribution ρ^π)
- Compact notation: $V_\theta = \Phi\theta$, where
 - $\Phi = [\phi(s_1)^T; \phi(s_2)^T; \dots \phi(s_S)^T] \in \mathbb{R}^{S \times d}$
 - $\phi(s) = [\varphi_1(s) \dots \varphi_d(s)]^T \in \mathbb{R}^d$
- Interested in fixed point solution of

$$\Phi\theta = \Pi^{\rho^\pi} \mathcal{T}^\pi(\Phi\theta)$$
- With linear approximation, projection is linear.
- Π^{ρ^π} : **weighted** least squares projection, i.e., minimizes $\|V - \Phi\theta\|_{\rho^\pi}^2$)



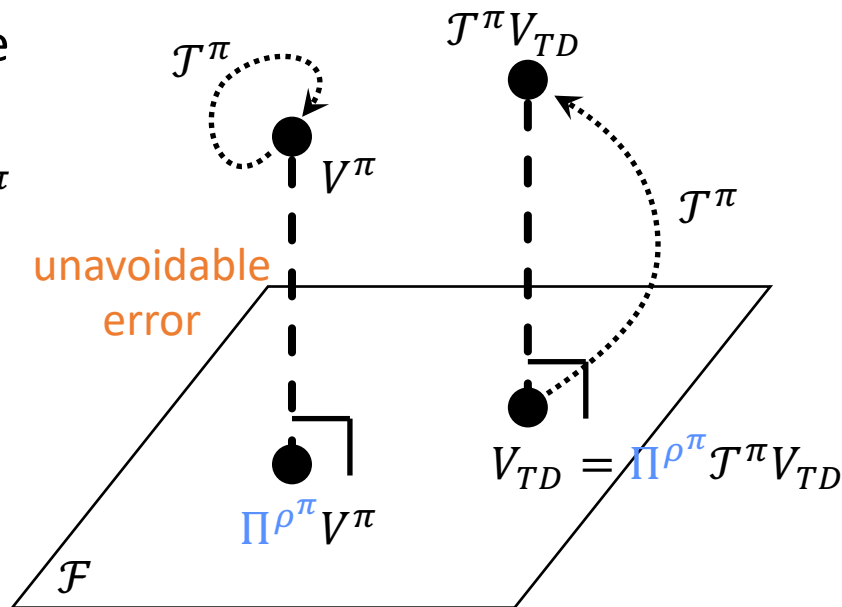
Least Squares TD, i.e. LSTD

- Recall: $\mathcal{T}^\pi V = r + \gamma P^\pi V$
- By definition of projection (assume linearly independent features Φ), the unique solution θ^* satisfies:

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^K} \|\Phi\theta - (r + \gamma P^\pi \Phi\theta)\|_{\rho^\pi}^2$$

- Setting gradient to 0, we obtain:
 $\Phi^T D_{\rho^\pi} (\Phi\theta^* - (r + \gamma P^\pi \Phi\theta^*)) = 0$
 With D_{ρ^π} a diagonal matrix with entries ρ^π .

$$\begin{aligned} V_{TD} &= \Phi\theta^* = \Pi^{\rho^\pi} \mathcal{T}^\pi \Phi\theta^* \\ &= \Pi^{\rho^\pi} \mathcal{T}^\pi V_{TD} \end{aligned}$$



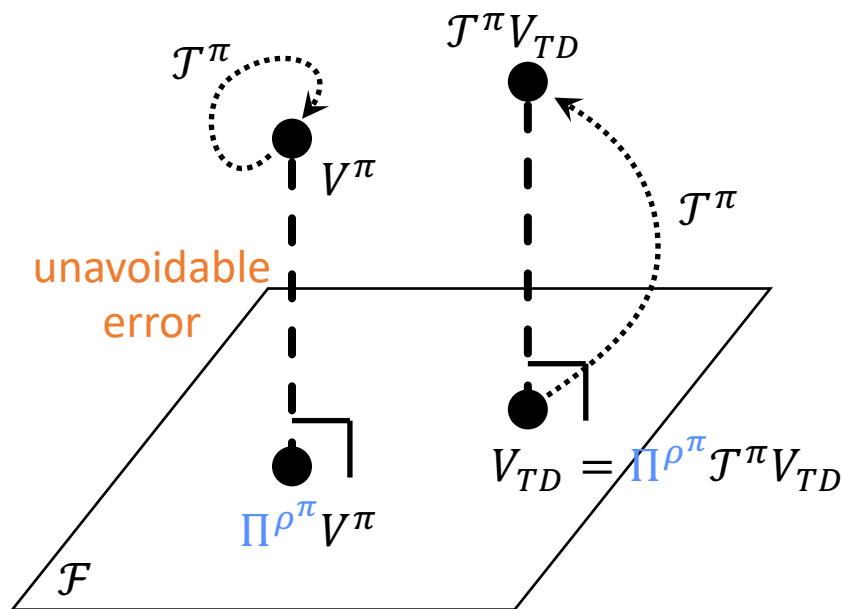
Least Squares TD, i.e. LSTD

Solving for θ^*

$$\Phi^T D_{\rho^\pi} (\Phi \theta^* - (r + \gamma P^\pi \Phi \theta^*)) = 0$$

- **Direct solution:** $C\theta = d$, where $C = \Phi^T D_{\rho^\pi} (I - \gamma P^\pi) \Phi$, $d = \Phi^T D_{\rho^\pi} r$
- **Iterative method** (projected VI (PVI), analogous to VI):
 - $\Phi \theta_{t+1} = \Pi^{\rho^\pi} \mathcal{T}(\Phi \theta_t)$
 - Can write PVI explicitly:
 $\theta_{t+1} = \arg \min_{\theta \in \mathbb{R}^K} \|\Phi \theta - (r + \gamma P^\pi \Phi \theta_t)\|_{\rho^\pi}^2$
- **Incremental variants** (like TD(0), TD(λ))

$$\begin{aligned} V_{TD} &= \Phi \theta^* = \Pi^{\rho^\pi} \mathcal{T}^\pi \Phi \theta^* \\ &= \Pi^{\rho^\pi} \mathcal{T}^\pi V_{TD} \end{aligned}$$



Linear TD (c.f. NDP, Assumption 6.1 and Prop 6.5)

Theorem (Bradtke and Barto, 1996)

We assume that:

1. **[Stationary distribution]** There exists a distribution ρ^π over $\mathcal{S}$ such that $\lim_{t \rightarrow \infty} P^\pi(s_t = s' | s_0 = s) = \rho^\pi(s') > 0, \forall s, s' \in \mathcal{S}$; denote $\rho^\pi = (\rho^\pi(s_1), \rho^\pi(s_2), \dots, \rho^\pi(s_{\mathcal{S}}))$ and let Π^{ρ^π} be the projection with respect to the **weighted Euclidean norm** $\|\cdot\|_{\rho^\pi}$.
2. **[Features are full rank]** The features $(\phi_i)_{1 \leq k \leq K \leq |\mathcal{S}|}$ are linearly independent.

Then, the mappings $\mathcal{T}^{\lambda, \pi}$ and $\Pi^{\rho^\pi} \mathcal{T}^{\lambda, \pi}$ are contractions of factor $\alpha_\lambda = \frac{\gamma(1-\lambda)}{1-\gamma\lambda}$ w.r.t $\|\cdot\|_{\rho^\pi}$.

Furthermore, the linear TD estimate converges to θ^* , and $\Phi\theta^*$ is the fixed point of the **projected Bellman operator**:

$$\Phi\theta^* = \Pi_{\rho^\pi} \mathcal{T}^{\lambda, \pi} \Phi\theta^*$$

And it has error:

$$L_{\rho^\pi}(\theta^*) \leq \frac{1}{\sqrt{1 - \alpha_\lambda^2}} \min_{\theta} L_{\rho^\pi}(\theta)$$

Where L_{ρ^π} is the expected loss w.r.t. the stationary distribution ρ^π .

👉 Linear TD converges.

👉 $\gamma = 0 \rightarrow$ accurate. **Discuss: Why?**

👉 Error is related to the best possible error.

👉 $\lambda = 0 \rightarrow$ inaccurate. **Discuss: Why?**

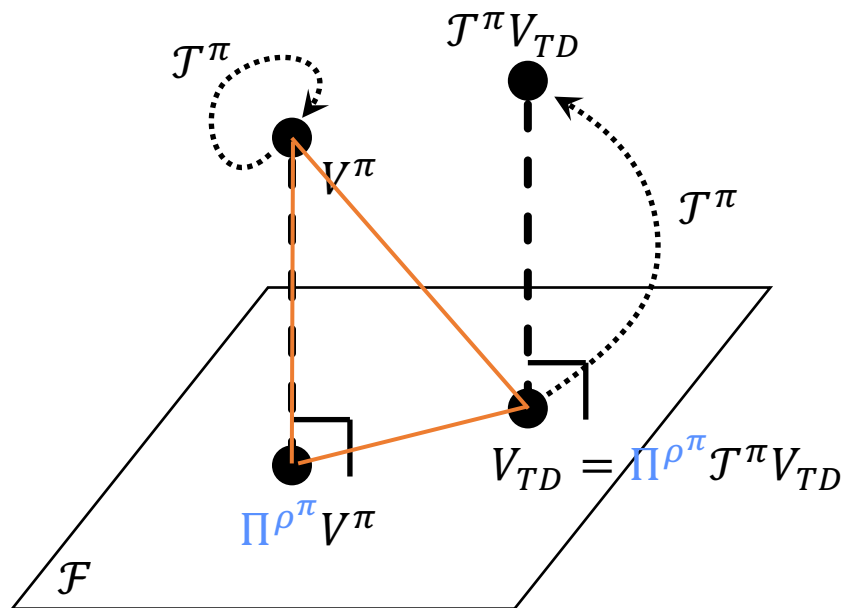
Proof (sketch): Linear TD ($\lambda = 0 \rightarrow \alpha_0 = \gamma$)

1. Show that $\Pi^{\rho^\pi} \mathcal{J}^\pi$ is a contraction in L_{2,ρ^π} with a unique fixed point V_{TD} .

- $\mathcal{J}^\pi$ is a contraction
- Π^{ρ^π} is a non-expansion

2. Bound the error using the Pythagorean theorem.

$$\begin{aligned} V_{TD} &= \Phi \theta^* = \Pi^{\rho^\pi} \mathcal{J}^\pi \Phi \theta^* \\ &= \Pi^{\rho^\pi} \mathcal{J}^\pi V_{TD} \end{aligned}$$



Approximate TD

Approximate TD **may not converge** (i.e. it might diverge) if:

- Linear approximation but states s_i are obtained by following a different policy (**off-policy learning**)
- **Non-linear approximation** and states s_i are obtained by following π

Outline

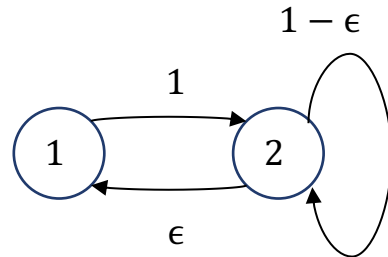
1. Introduction to value function approximation
2. **Approximate value-based algorithms**
 - a. **Policy evaluation**
 - Approximate Monte Carlo
 - Approximate TD(0)
 - Convergence results: LSTD & Linear TD
 - **Divergence counterexample**
 - b. Policy learning

Counterexample: Divergence of Off-policy Linear TD(0)

- Consider the Markov chain induced by π , with two states $S = \{1, 2\}$. Reward is always 0.
- Consider a $\rho' \neq \rho^\pi$, that weights each state $\{1, 2\}$ with equal probability.
- Linear function approximation with parameter $w \in \mathbb{R}$
 - $V_w(s) = w \cdot s$
 - Note: $\phi(s) := s$, that is, $\phi(1) = 1, \phi(2) = 2$

- Recall linear TD update:

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \alpha_t (\phi_t^T \theta_t - r_t - \gamma \phi_{t+1}^T \theta_t) \phi_t$$



Compare

- TD(0) with states sampled from ρ^π (**converges**)
- TD(0) with states sampled from ρ' (**diverges** for $\gamma > 3/4$, ϵ small enough, if $w_0 \neq 0$)

$$\begin{aligned} \mathbb{E}[w_{t+1}] &= \mathbb{E}[w_t] - \alpha_t 4(1 - \gamma) \mathbb{E}[w_t] + O(\epsilon) \mathbb{E}[w_t] \\ \mathbb{E}[w_{t+1}] &= \mathbb{E}[w_t] - \alpha_t \frac{1}{2} (1 - 2\gamma) \mathbb{E}[w_t] - \alpha_t \left((1 - \gamma) + \frac{\gamma\epsilon}{2} \right) \mathbb{E}[w_t] \\ &= \mathbb{E}[w_t] - \alpha_t \frac{1}{2} (3 - 4\gamma) \mathbb{E}[w_t] + O(\epsilon) \mathbb{E}[w_t] \end{aligned}$$

Numerical example: Baird's counterexample

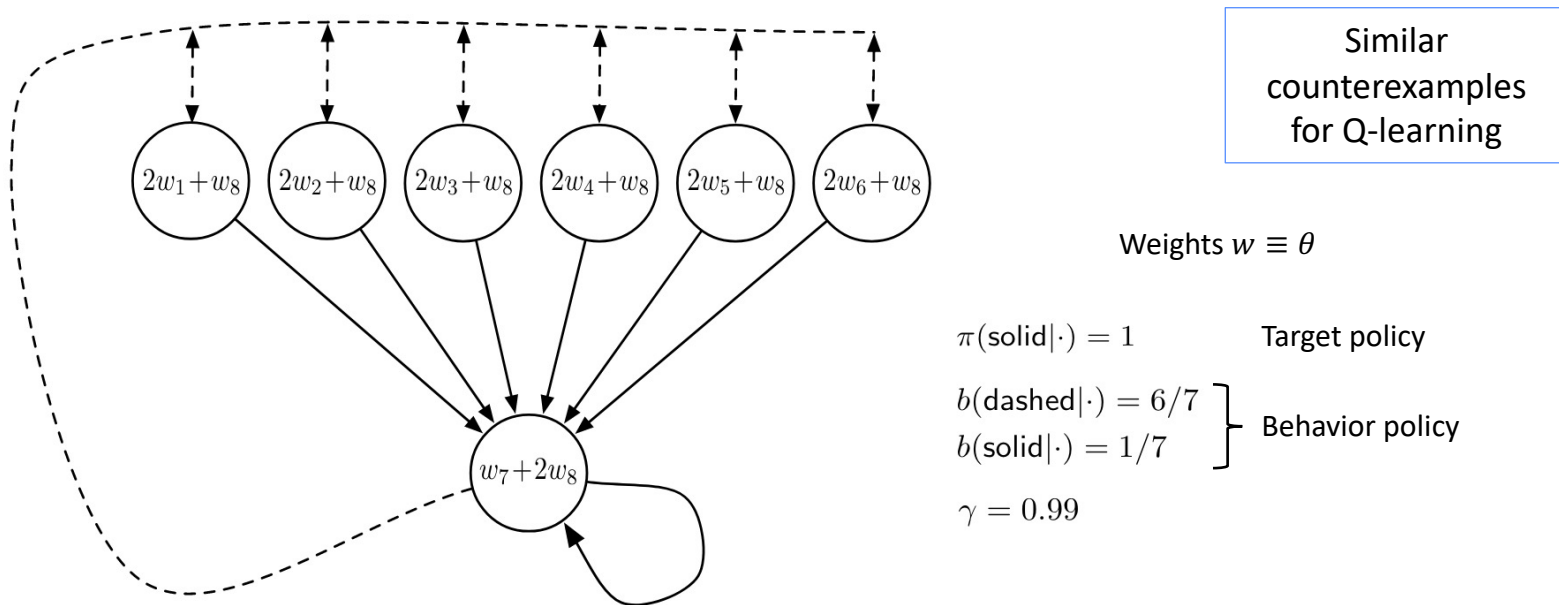


Figure 11.1: Baird's counterexample. The approximate state-value function for this Markov process is of the form shown by the linear expressions inside each state. The **solid** action usually results in the seventh state, and the **dashed** action usually results in one of the other six states, each with equal probability. The reward is always zero.

Numerical example: Baird's counterexample

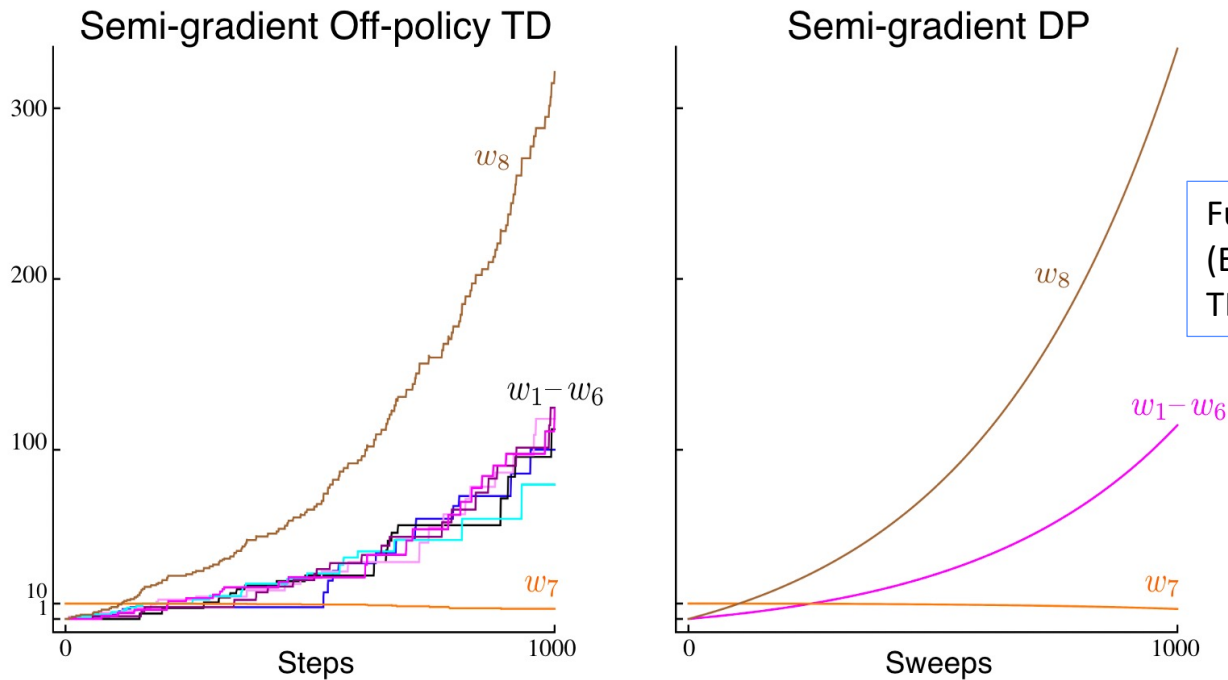


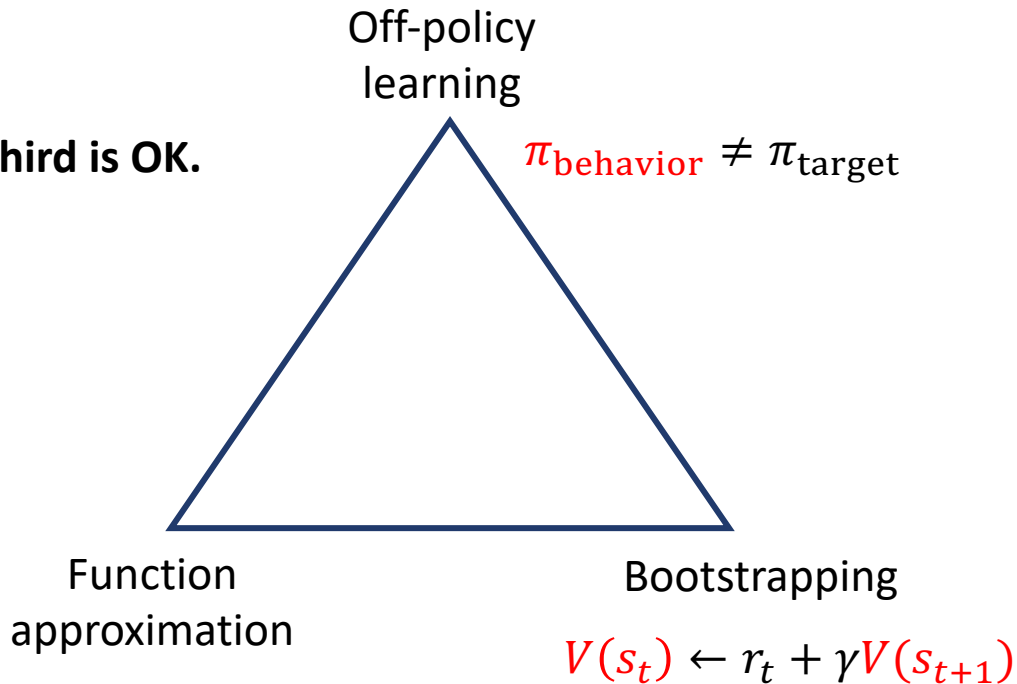
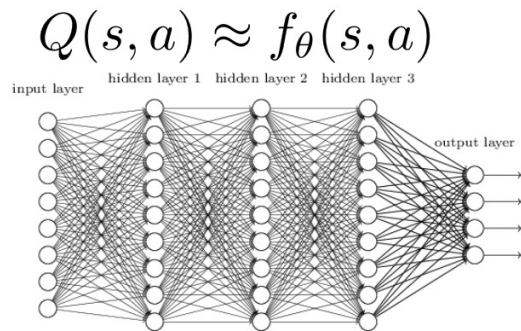
Figure 11.2: Demonstration of instability on Baird's counterexample. Shown are the evolution of the components of the parameter vector $\mathbf{w}$ of the two semi-gradient algorithms. The step size was $\alpha = 0.01$, and the initial weights were $\mathbf{w} = (1, 1, 1, 1, 1, 1, 10, 1)^\top$.

Weights $\mathbf{w} \equiv \boldsymbol{\theta}$

The deadly triad

The risk of divergence arises whenever we combine:

Claim: Any two without the third is OK.



Possible remedies (and further readings)

- More careful algorithm designs [Sutton et al., 2009; S&B §11.7-11.8]
 - “Fast gradient-descent methods for temporal-difference learning with linear function approximation.” ICML.
 - Gradient TD (GTD)
 - TD with gradient correction (TDC)
 - Emphatic TD [Sutton et al., 2016], etc.
 - “An emphatic approach to the problem of off-policy temporal-difference learning.” JMLR.
 - **Convergence** guarantees for off-policy and “mildly” non-linear approximators
- Reducing variance [S&B §11.9]
- Using a target network [Mnih et al., 2015, Zhang et al., 2021]
 - “Human-level control through deep reinforcement learning.” Nature.
 - “Breaking the deadly triad with a target network.” ICML.
 - Target network Q_{target} : periodically synced by the value network
 - Value network Q : updated via gradient methods
 - Key ingredients in (double) deep Q-learning (DQN).

Summary

- **Value function approximation**: features vs function class
- Unlike its tabular counterparts, approximate value function methods may diverge, even for policy evaluation
 - **Approximate TD**, pseudo-gradient methods, and its linear version
- **Approximate Q-value iteration** shares convergence properties with **approximate value iteration** and lends itself to regression.
- **Fitted Q iteration** avoids enumeration of the state space by fitting the Q function to bootstrap targets.
 - Basis for a value-based deep RL methods, including **Deep Q Networks (DQN)**.
- **DQN**: **replay buffer**, **target networks**, **over-estimation** and other advances

References

1. *DPOC vol2 §2.5.3; [SB 16.5](#)*
2. Alessandro Lazaric. INRIA Lille. Reinforcement Learning. 2017, Lecture 4.
3. Neuro-Dynamic Programming (NDP). §3.1-3.2, Ch 6.

Reference: Detailed proof for Linear TD (Prop 6.5)

And theorem for the incremental version

Proof: Linear TD ($\lambda = 0 \rightarrow \alpha_0 = \gamma$)

First, want to show that $\Pi^{\rho^\pi} \mathcal{T}^\pi$ is a contraction in L_{2,ρ_π} with a unique fixed point.

- The transition matrix induced by π does not increase the (weighted) norm:

$$\begin{aligned} \|P^\pi V\|_{\rho^\pi}^2 &= \sum_s \rho^\pi(s) (\sum_{s'} p(s'|s, \pi(s)) V(s'))^2 \\ &\leq \sum_s \rho^\pi(s) \sum_{s'} p(s'|s, \pi(s)) V(s')^2 \\ &= \sum_{s'} \rho^\pi(s') V(s')^2 = \|V\|_{\rho^\pi}^2 \end{aligned}$$

- The second inequality follows the Jensen's inequality.
 - The third equality holds because ρ^π is a stationary distribution.
- Then it immediately follows that $\mathcal{T}^\pi$ is a contraction in L_{2,ρ_π} , i.e.,

$$\|\mathcal{T}^\pi V_1 - \mathcal{T}^\pi V_2\|_{\rho_\pi} = \gamma \|P^\pi(V_1 - V_2)\|_{\rho_\pi} \leq \gamma \|V_1 - V_2\|_{\rho_\pi}$$
- It can be shown that Π^{ρ^π} is a non-expansion.
- $\rightarrow$ unique fixed point $V_{TD} = \Pi^{\rho^\pi} \mathcal{T}^\pi V_{TD}$.

Proof: Linear TD ($\lambda = 0 \rightarrow \alpha_0 = \gamma$)

Now, let's consider the error:

- By **Pythagorean theorem**, we have:

$$\begin{aligned} & \|V^\pi - V_{TD}\|_{\rho^\pi}^2 \\ &= \|V^\pi - \Pi^{\rho^\pi} V^\pi\|_{\rho^\pi}^2 + \|\Pi^{\rho^\pi} V^\pi - V_{TD}\|_{\rho^\pi}^2 \end{aligned}$$

- But:

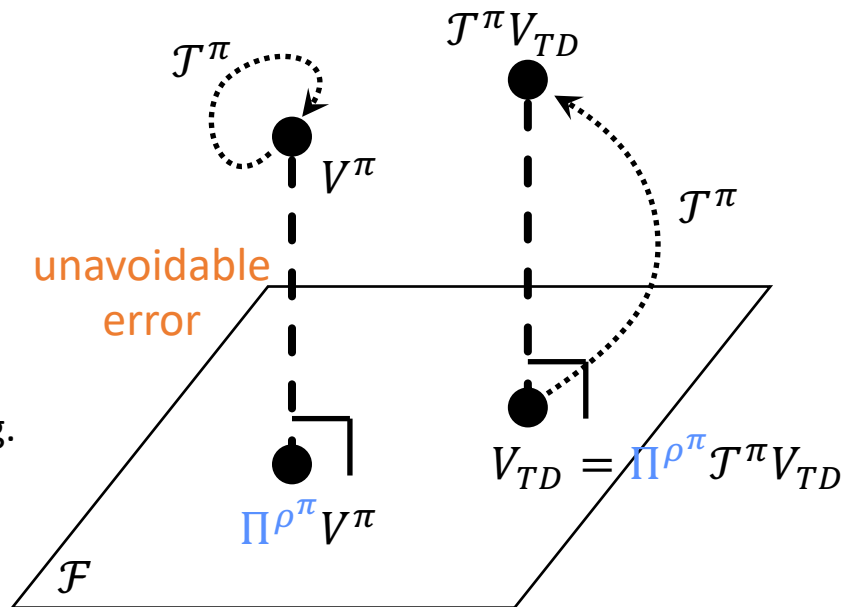
$$\begin{aligned} \|\Pi^{\rho^\pi} V^\pi - V_{TD}\|_{\rho^\pi}^2 &= \|\Pi^{\rho^\pi} V^\pi - \Pi^{\rho^\pi} \mathcal{T}^\pi V_{TD}\|_{\rho^\pi}^2 \\ &\leq \|\mathcal{T}^\pi V^\pi - \mathcal{T}^\pi V_{TD}\|_{\rho^\pi}^2 \leq \gamma^2 \|V^\pi - V_{TD}\|_{\rho^\pi}^2 \end{aligned}$$

- Thus

$$\begin{aligned} & \|V^\pi - V_{TD}\|_{\rho^\pi}^2 \\ &\leq \|V^\pi - \Pi^{\rho^\pi} V^\pi\|_{\rho^\pi}^2 + \gamma^2 \|V^\pi - V_{TD}\|_{\rho^\pi}^2 \end{aligned}$$

- Which corresponds to the result after reordering. ■

$$\begin{aligned} V_{TD} &= \Phi \theta^* = \Pi^{\rho^\pi} \mathcal{T}^\pi \Phi \theta^* \\ &= \Pi^{\rho^\pi} \mathcal{T}^\pi V_{TD} \end{aligned}$$



Incremental Linear TD

Theorem (Tsitsiklis and Van Roy, 1996; c.f. NDP, Prop 6.5)

Let the learning rate η_t satisfy $\sum_{t \geq 0} \eta_t = \infty$, and $\sum_{t \geq 0} \eta_t^2 < \infty$.

We assume that there exists a distribution ρ over $\mathcal{S}$ such that $\forall s, s' \in \mathcal{S}$, $\lim_{t \rightarrow \infty} P(s_t = s' | s_0 = s) = \rho(s') > 0$ and that the features $(\phi_i)_{1 \leq k \leq K \leq |\mathcal{S}|}$ are linearly independent. Let the updates be given by:

$$\theta_{t+1} = \theta_t + \eta_t \delta_t \sum_{k=0}^t (\gamma \lambda)^{t-k} \phi(s_k)$$

Then there exists a fixed θ^* such that $\lim_{t \rightarrow \infty} \theta_t = \theta^*$. Furthermore, we obtain:

$$\|V_{\theta^*} - V^\pi\|_{2,\rho} \leq \frac{1 - \lambda\gamma}{1 - \gamma} \inf_{\theta} \|V_\theta - V^\pi\|_{2,\rho}$$

👉 See NDP §6.3.3 for detailed analysis. Need to cope with updates that worsen value function. Noise is NOT conditionally mean zero. **Leverage stochastic approximation results for Markovian noise (NDP §4.4).**