Advanced Topics in Reinforcement Learning

Lecture 14: Off-Policy Function Approximation

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Homework due October 21 at 9:30AM (minute class starts)
- Read 9.7 and 16.5 for next week. Deep RL!
- Upcoming dates:
  - Literature survey due: October 30
  - Exam: November 5

Learning Outcomes

After this week, you will be able to:

1. Generalize model-free RL algorithms from the tabular to the function approximation setting.

2. Identify challenges and opportunities with using function approximation in RL.

3. Compare and contrast convergence of different algorithms under either function approximation or off-policy learning.

Function Approximation Review

- Objective with function approximation.
  - $\overline{VE}(\mathbf{w}) \doteq \sum_{s \in \mathcal{S}} \mu(s) \left[v_\pi(s) - \hat{v}(s, \mathbf{w}) \right]^2$
- Semi-gradient TD update.
  - $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha(U_t - \hat{v}(S_t, \mathbf{w}_t)) \nabla \hat{v}(S_t, \mathbf{w}_t)$

Estimate $v_\pi(S_t)$ with $R_{t+1} + \gamma v(S_{t+1}, w_t)$

- Linear Semi-Gradient Update
  - $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha(U_t - \hat{v}(S_t, \mathbf{w}_t))\mathbf{x}(S_t)$

Step-size Selection

- The step-size is an important parameter in any SGD algorithm.
- Book gives rule of thumb:

$$\alpha = (\tau E[x^\top x])^{-1}$$

- Why does this make sense?
- Not often used in practice.

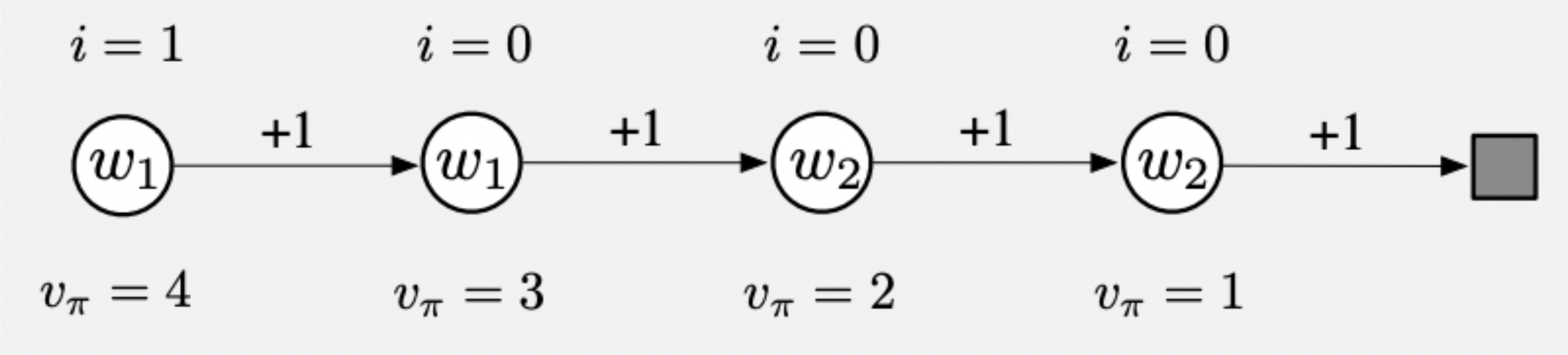
LSTD(0)

- Convergence analysis shows that on-policy, linear TD(0) converges to $\mathbf{w}_{\mathtt{TD}} = \mathbf{A}^{-1}\mathbf{b}$.
  - $A = \mathbf{E}_{\mu}[\mathbf{x}_t(\mathbf{x}_t - \gamma \mathbf{x}_{t+1})^\top]$ and $\mathbf{b} = \mathbf{E}_{\mu}[R_{t+1}\mathbf{x}_t]$.
- LSTD(0) estimates A and b with all data and then directly computes the fixed point.
  - (+) More data efficient than semi-gradient linear TD(0)
  - (-) More computation (after optimizations $O(d^2)$ vs $O(d)$ for TD(0))
- Harder to extend to deep reinforcement learning.

Interest and Emphasis

- So far, assumed we are updating states equally (same learning rate) but according to the on-policy state distribution, μ.
- We may wish to emphasize some states more.
- State interest, I_t, represents how much we care about accurate estimation in state S_t.
- Emphasis is a learned multiplier on the learning rate.
  - $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha M_t[R_t - \hat{v}(S_{t+1}, \mathbf{w}) - \hat{v}(S_t, \mathbf{w})] \nabla \hat{v}(S_t, \mathbf{w})$
  - $M_t \leftarrow I_t + \gamma M_{t-1}$

Interest and Emphasis



- Interest is (1, 0, 1, 0)
- Semi-gradient 2-step TD converges to weight vector (3.5, 1.5)
- Emphatic 2-step TD converges to weight vector (4, 2)

On-Policy Control

- As usual, for control we will estimate action-values, $\hat{q}(s, a, \mathbf{w})$.
- For linear function approximation, features are now a function of (s,a) pairs, $\mathbf{x}(s, a)$.
- Function approximation often inherently means that making $\hat{q}(s, a, \mathbf{w})$ more accurate at one state will make it less accurate at another state.
- Now making π greedy w.r.t. $\hat{q}(s, a, \mathbf{w})$ is no longer guaranteed to improve π — no more policy improvement theorem.

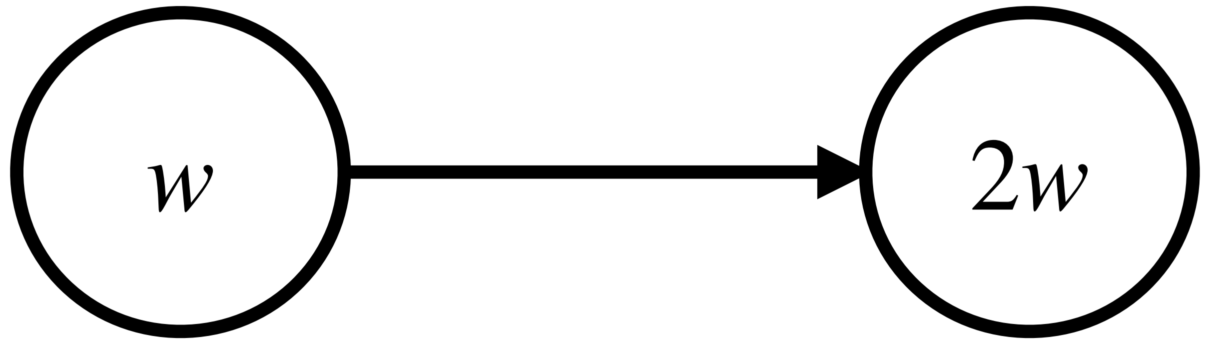
Off-Policy Prediction with Linear Function Approximation

- Last time, U_t, was produced by following π. Now we will follow a different behavior policy b.
- Recall from chapter 5, that we can correct for this by importance sampling.
  - N-step return: $G_{t:t+n} := R_{t+1} + \ldots + \gamma^{n-1} R_{t+n-1} + \gamma^n \hat{v}(S_{t+n}, \mathbf{w}_{t+n-1})$
  - For off-policy, replace $G_{t:t+n}$ with $\rho_{t:t+n} G_{t:t+n}$.

$$\rho_{t:t+n} := \prod_{i=t}^{t+n} \frac{\pi(A_i \mid S_i}{b(A_i \mid S_i)}$$

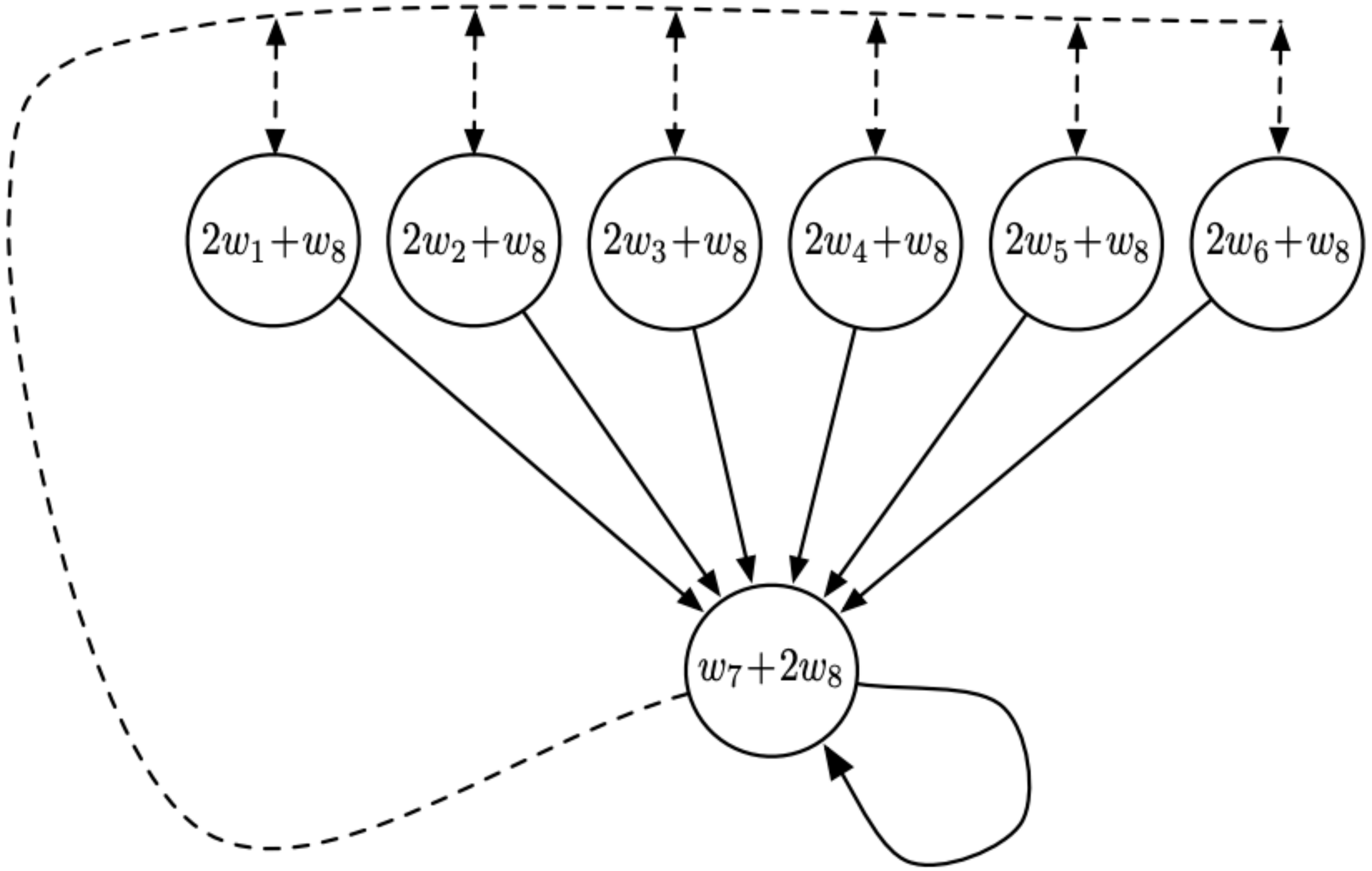
- Consider $U_t \leftarrow \rho_{t:t+n} G_{t:t+n}$. Does this update minimize our $\overline{VE}$ objective?
  - No — does not adjust for state weighting.

Divergence Example #1



- Initialize $w = 10$, $\gamma = 0.99$, $\alpha = 0.1$, and the transition gives zero reward.
- What happens with semi-gradient TD after you've seen this transition?
  - w increases to try and match bootstrapping target of $2\gamma w$.
- How can we fix divergence here?
  - First extend example to full MDP, then remove off-policy, bootstrapping, or function approximation.

Divergence Example #2: Baird's Counter-example



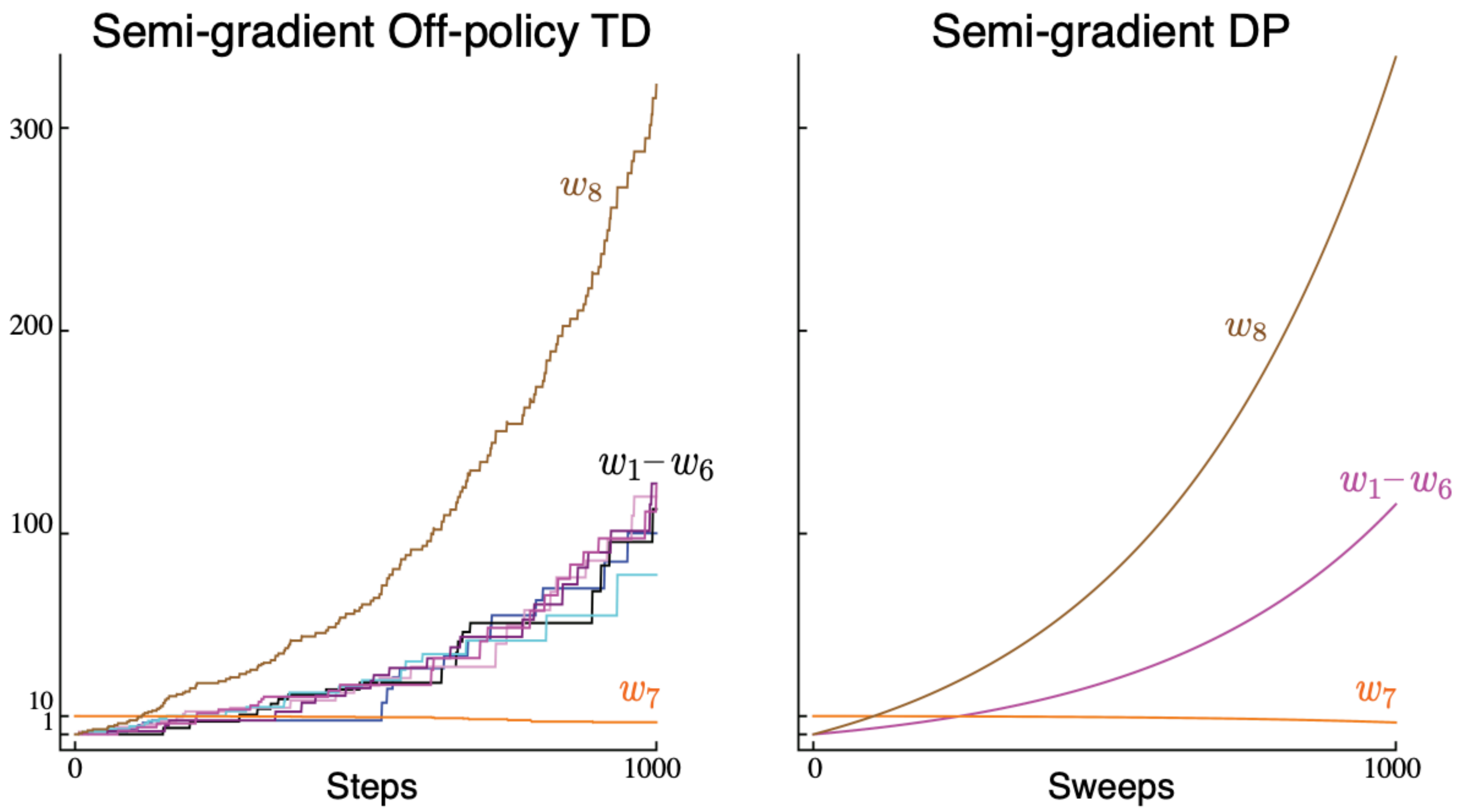
$\pi(\text{solid}|\cdot) = 1$

$b(\text{dashed}|\cdot) = 6/7$

$b(\text{solid}|\cdot) = 1/7$

$\gamma = 0.99$

Divergence Example #2: Baird's Counter-example



Off-Policy Divergence

- In general, we lack convergence or even stability results for the simplest and most practical off-policy, semi-gradient methods.
- Includes Q-learning which is one of the most widely used algorithms in RL.
    - Maybe OK if behavior and target policy are close?
    - State distributions will then be close.

The Deadly Triad

1. Function Approximation: changing the value estimate at one state affects the value estimate at other states.

2. Bootstrapping: using existing estimated values as part of the learning target instead of only using actual returns.

3. Off-Policy Learning: using a distribution of transitions (s, a, s', r) other than that of the target policy.

Do we need the deadly triad?

- Why use function approximation?
  - Too many states to represent explicitly; need generalization.
- Why bootstrap?
  - Memory and computation requirements; learning in non-episodic tasks; faster learning.
- Why use off-policy learning?
  - Separate exploration and exploitation; general purpose learning agents must learn about multiple reward signals and target policies at the same time.

The Deadly Triad in Deep RL

- In practice, each component of the deadly triad is not binary.
- Bootstrapping: can use n-step returns or target networks to decrease amount of bootstrapping.
- Function approximation: larger neural networks decrease over-generalization.
- Off-Policy learning: controlling distribution of samples from the replay buffer modulates how off-policy updates are.

“Deep Reinforcement Learning and the Deadly Triad.” Van Hasselt et al. 2018.

Hanwen’s Presentation

- Emphatic Temporal-Difference (ETD) Learning
- Rupam Mahmood, Huizhen Yu, Martha White, and Richard Sutton.
- Slides

Summary

- Off-policy semi-gradient methods often lack stability and convergence results due to the deadly triad.
- Deadly Triad: off-policy, function approximation, and bootstrapping.

Action Items

- Complete homework.
- Begin literature review.
- Begin reading Chapter 9.7 and 16.5.