Advanced Topics in Reinforcement Learning

Lecture 15: Off-Policy Function Approximation II

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Homework 3 due 1 minute ago; homework 4 released ~~tonight~~ Soon.
  - Due Nov 17. We won't cover relevant material until 2 weeks from now.
- Begin reading deep RL readings: Section 9.7 and 16.5 of course textbook.
- Midterm survey
  - At 65% right now. Please complete by Friday evening!

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.

Yohei’s Presentation

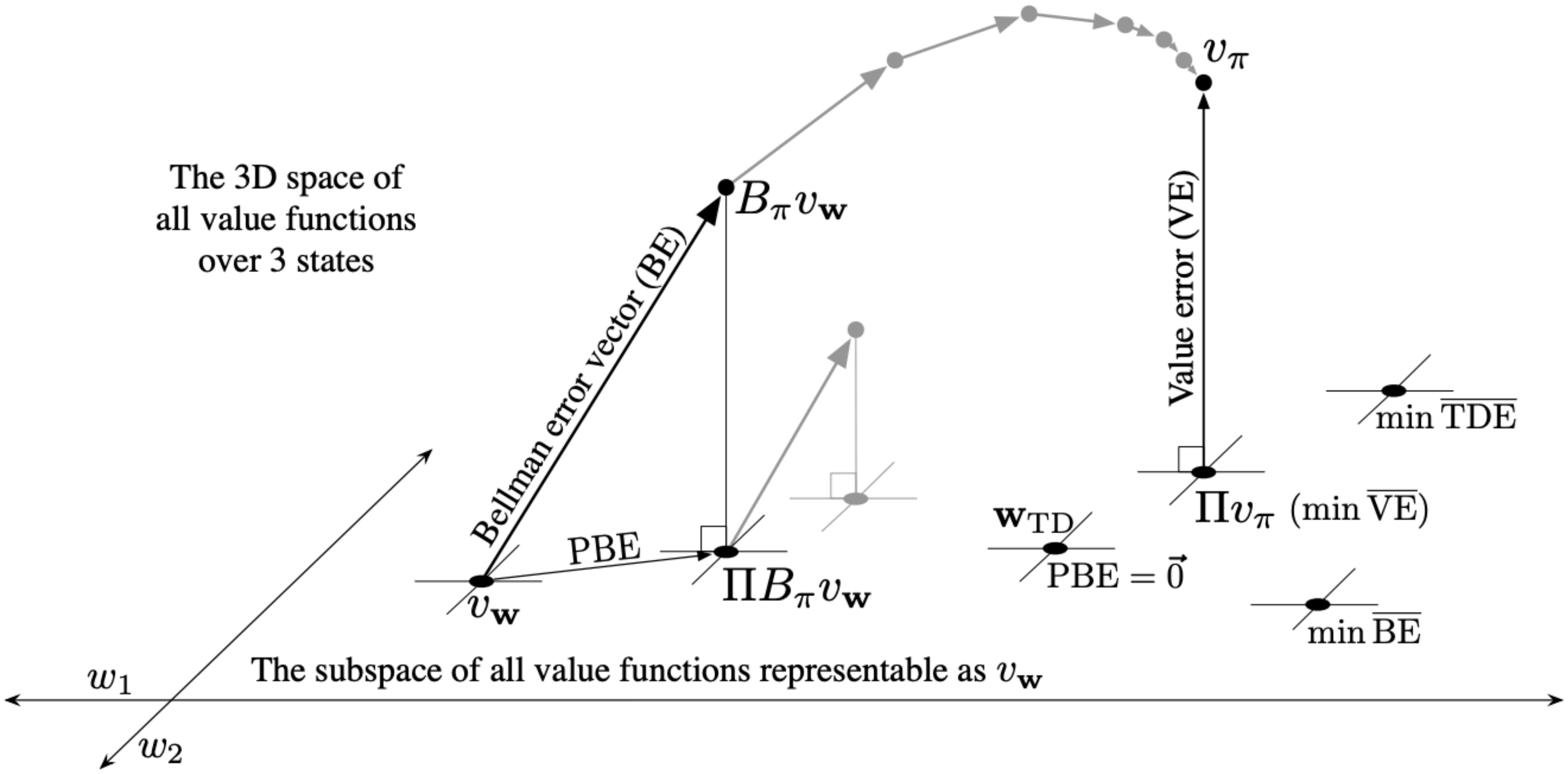
- Slides

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.

Geometric Interpretation of Value Functions



Possible Learning Objectives

- Minimum value error
  - $\overline{VE}(\mathbf{w}) = \sum_s \mu(s)(v_\pi(s) - \hat{v}(s, \mathbf{w}))^2 = ||v_\mathbf{w} - v_\pi||^2_\mu$

Only truly SGD with $v_\pi(s) \approx G_t$ for $S_t = s$.

$\delta_t = R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t)$

- Minimum TD-Error
  - $\overline{TDE}(\mathbf{w}) = \sum_s \mu(s)\mathbf{E}_\pi[\delta_t^2 \,|\, S_t = s, A_t \sim \pi]$

Full-gradient TD learning (Naive residual gradient)

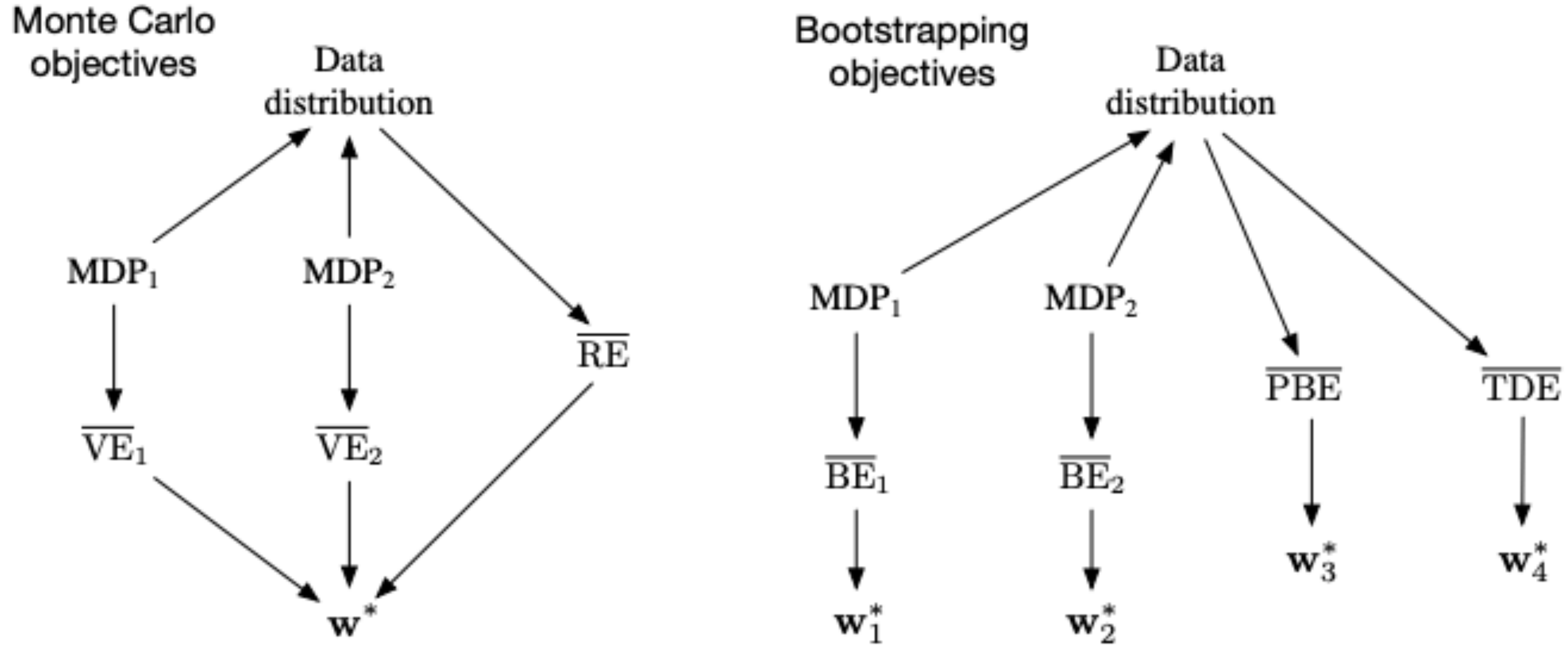
- Minimum Bellman error:
  - $\overline{BE}(\mathbf{w}) = ||\delta_\mathbf{w}||^2_\mu$
  - $\delta_\mathbf{w} = \mathbf{E}_\pi[\delta_t \,|\, S_t = s, A_t \sim \pi]$

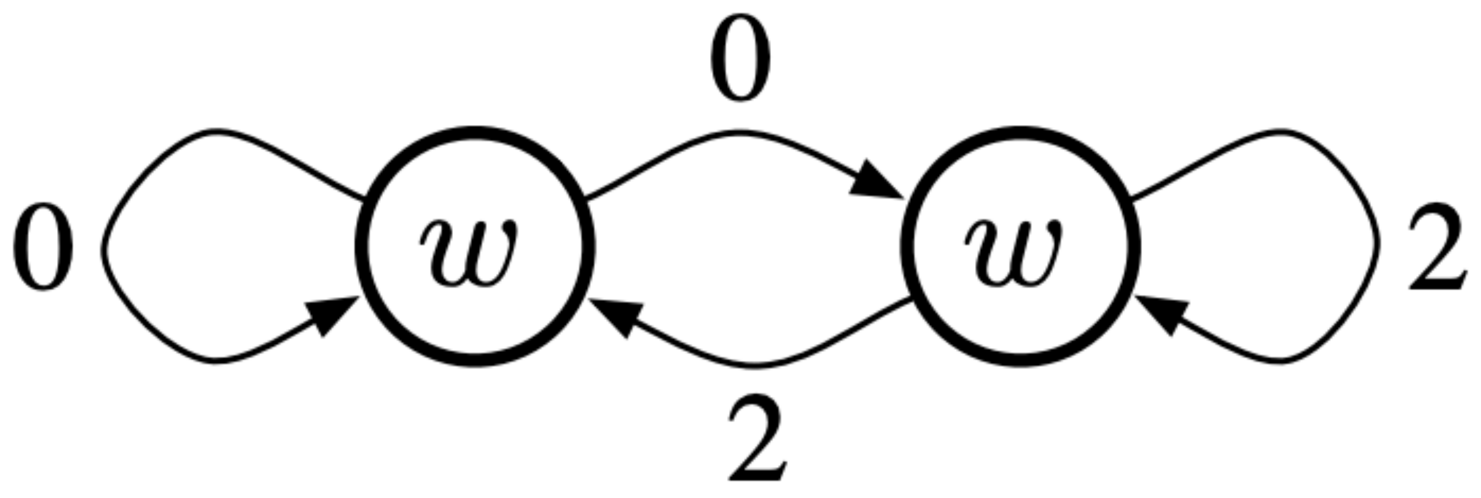
Residual Gradient Algorithm

Bellman Error

- The Bellman error in a state is the expected TD error in that state.
- The Bellman error objective is the per-state Bellman error weighted by μ.
  - $\overline{BE}(\mathbf{w}) = ||\delta_{\mathbf{w}}||_{\mu}^{2}$
  - $\delta_{\mathbf{w}} = \mathbb{E}_{\pi}[R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t)]$
- In the tabular setting, $\delta_{\mathbf{w}} = 0 \implies v_{\mathbf{w}} = v_{\pi}$. What can we say about linear function approximation?
  - May not be possible to obtain zero error.
- $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \alpha \nabla E_{\pi}[\delta_t]^2$ with $\nabla E_{\pi}[\delta_t]^2 = E_b[\rho_t \delta_t][\nabla \hat{v}(S_t, \mathbf{w}_t) - \gamma E_b[\rho_t \hat{v}(S_{t+1}, \mathbf{w}_t)]]$

Learnability of the Bellman Error





0,2,2,2,2,0,0,2,2,2,2,2,2,2,2,0,0,0

Minimal Projected Bellman Error

- Projected Bellman Error: Apply Bellman operator to $v_{\mathbf{w}}$, then project into representable space of value functions.

  Policy evaluation update from chapter 4

  - $\overline{PBE}(\mathbf{w}) = ||\Pi B_\pi v_{\mathbf{w}} - v_{\mathbf{w}}||_\mu^2$ or equivalently $||\Pi(B_\pi v_{\mathbf{w}} - v_{\mathbf{w}})||_\mu^2$.

- The *projected Bellman Error* is uniquely determined by the data distribution.

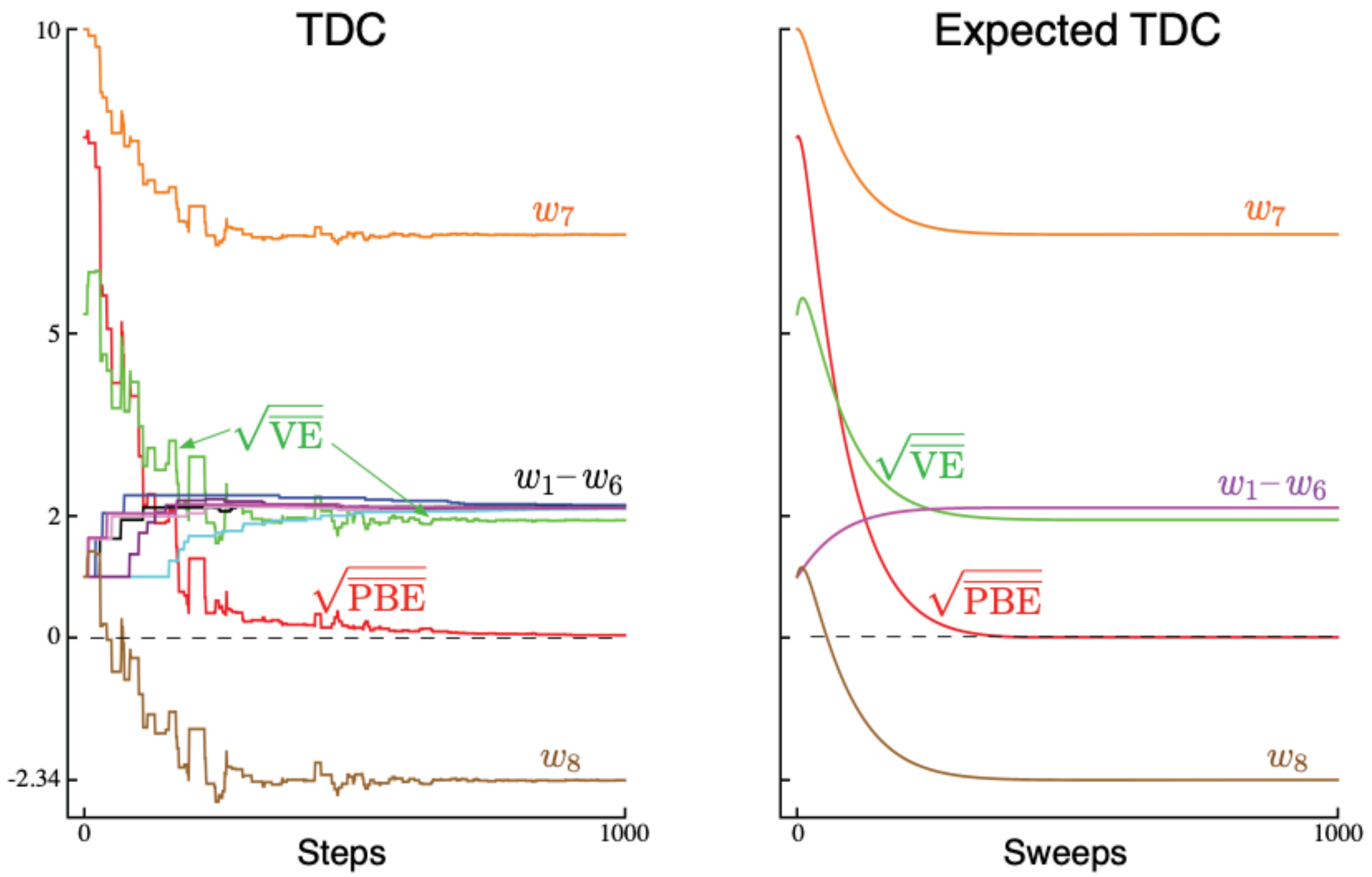
  - Learnable!

- Since PBE is learnable, we can use $\overline{PBE}(\mathbf{w})$ as an objective for SGD.

Gradient-TD

- SGD with: $\nabla \overline{PBE}(\mathbf{w}) = 2\mathbb{E}[\rho_t(\gamma x_{t+1} - x_t)x_t^\top]\mathbb{E}[x_t x_t^\top]^{-1}\mathbb{E}[\rho_t \delta_t x_t]$
- Define $\mathbf{v} \approx \mathbb{E}[x_t x_t^\top]^{-1}\mathbb{E}[\rho_t \delta_t x_t]$.
  - In matrix form, this is a solution to a linear regression with features x_t and target $\rho_t \delta_t$.
  - Instead of instantly solving for $\mathbf{v}$, we will estimate with SGD:
- When $\mathbf{v}$ is learned, we substitute it in for the last two terms in the gradient.
  - $\nabla \overline{PBE}(\mathbf{w}) \approx \rho_t(x_t - \gamma x_{t+1})x_t^\top v_t$

Gradient-TD

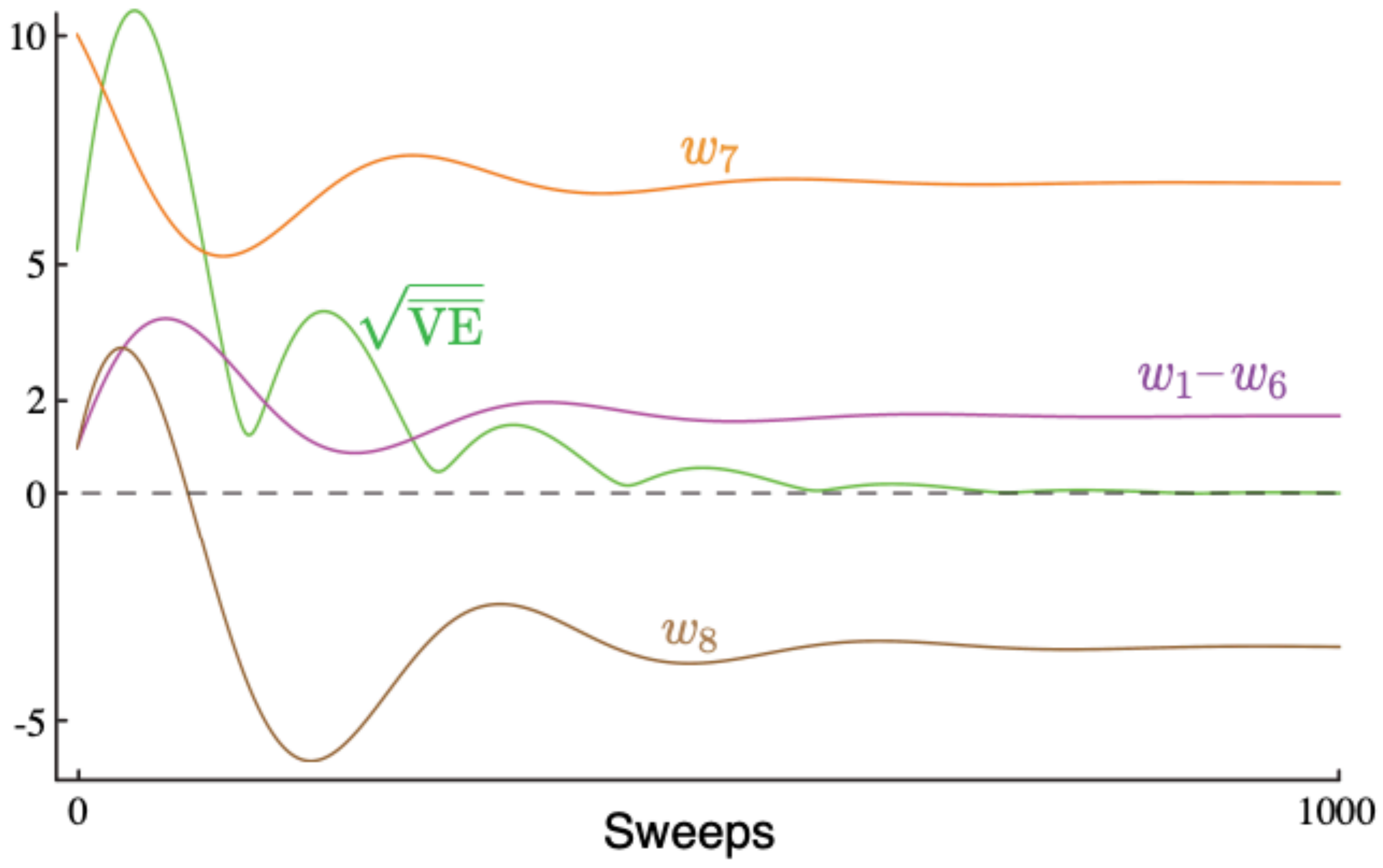


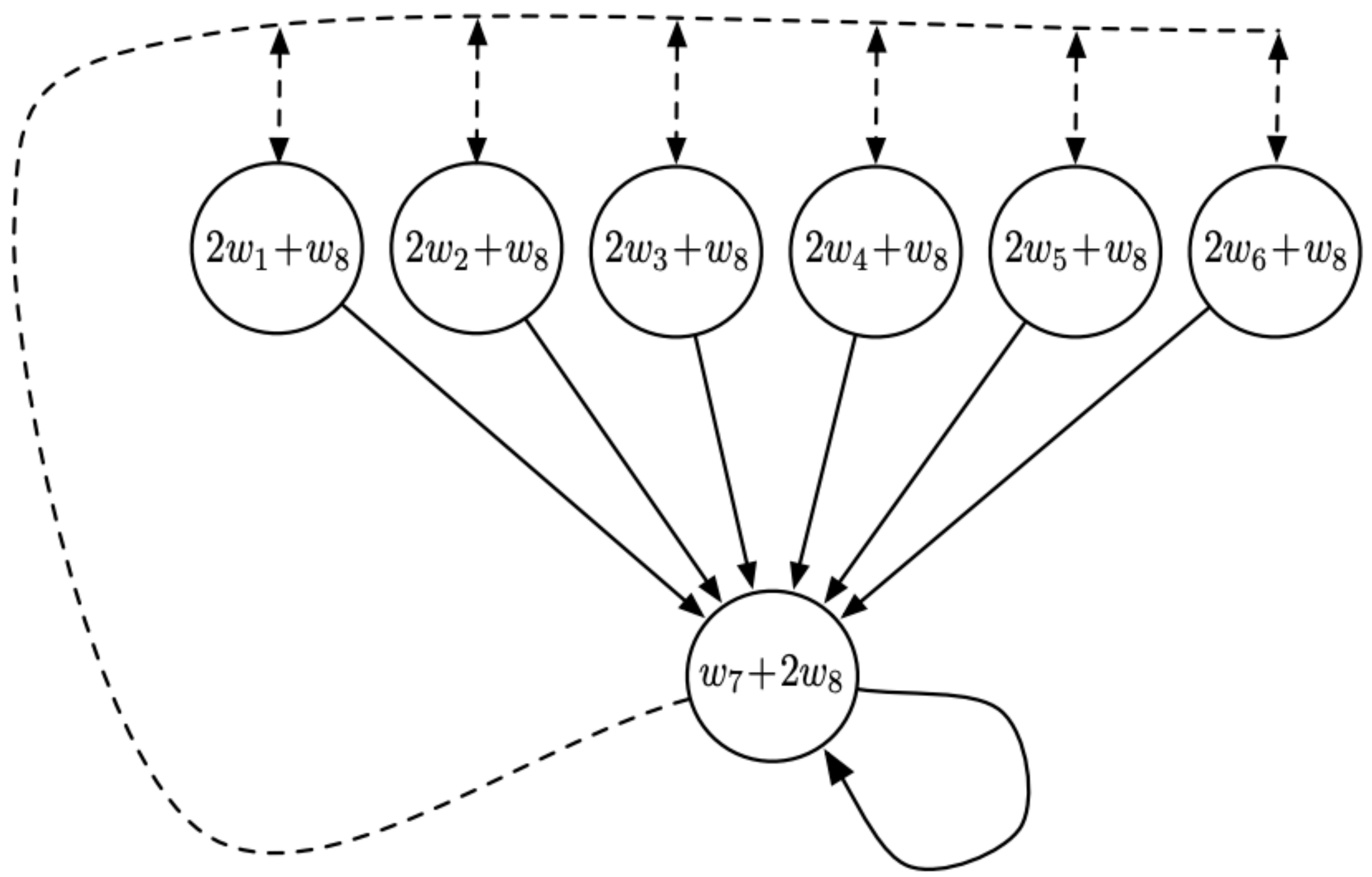
TDC
Expected TDC
10
5
2
0
-2.34
0
1000
Steps
Sweeps
w_7
$w_1 - w_6$
$\sqrt{\overline{VE}}$
$\sqrt{\overline{PBE}}$
w_8

Emphatic TD

- Keep $\overline{VE}(\mathbf{w})$ as our objective.
- Naively applying semi-gradient TD-learning will update states according to their visitation probability (i.e., the on-policy state distribution of the behavior policy).
- We can artificially change the importance of states by emphasizing some states more than others.
- 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.
  - $M_t \leftarrow I_t + \gamma \rho_{t-1} M_{t-1}$
  - $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \alpha M_t \rho_t [R_t - \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t)] \nabla \hat{v}(S_t, \mathbf{w}_t)$

Emphatic TD





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

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

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

$\gamma = 0.99$

Variance in Off-Policy Learning

- In many cases, off-policy learning is inherently of higher variance than on-policy learning.
  - Though not all cases!
- What to do in practice:
  - Keep behavior and target policy close.
  - Clip importance weights: $\bar{\rho}_t \leftarrow \min(\frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}, 1)$.
  - Weighted importance sampling.
  - Learn state density ratios: $\frac{d_\pi(S_t)}{d_b(S_t)}$.

Summary

- Deadly Triad: off-policy, function approximation, and bootstrapping.
- Two paths forward:
    - Reconsider our prediction objective with function approximation.
        - Leads to Gradient-TD methods.
    - Re-weight state updates.
        - Emphatic TD methods.
- Not clear what the “right” algorithm is yet!

Action Items

- Homework 4.
- Literature review due next week.
- Begin deep RL readings.
- Midterm survey (by tomorrow evening).