Chapter 11

Off-policy methods with approximation

Keys to off-policy methods

- Two policies:
  - the *target policy* π whose value function we are learning
  - the *behavior policy* b, which is used to select actions
- Off-policy is <u>much harder</u> with Function Approximation
  - even for linear FA
  - even for prediction (two fixed policies π and b)
  - even for dynamic programming

Challenges of off-policy learning

- Challenges can be divided into TWO parts:
  - the target of the learning update
    - which we know how to solve from Chapters 5 & 6
    - using importance sampling in target
  - the distribution of the updates
    - we are no longer updating according to the on-policy distribution (serious issue in function approximation case)
    - ***we are not sure how to solve***

per-step importance sampling ratio

$$\rho_t \doteq \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$

Off-policy semi-gradient methods

- Value prediction (e.g., TD(0))

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha \rho_t \delta_t \nabla \hat{v}(S_t, \mathbf{w}_t),$$

$$\delta_t \doteq R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t), \text{ or} \qquad \text{(episodic)}$$

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t). \qquad \text{(continuing)}$$

- Control (e.g., semi-gradient expected Sarsa)

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha \delta_t \nabla \hat{q}(S_t, A_t, \mathbf{w}_t)$$

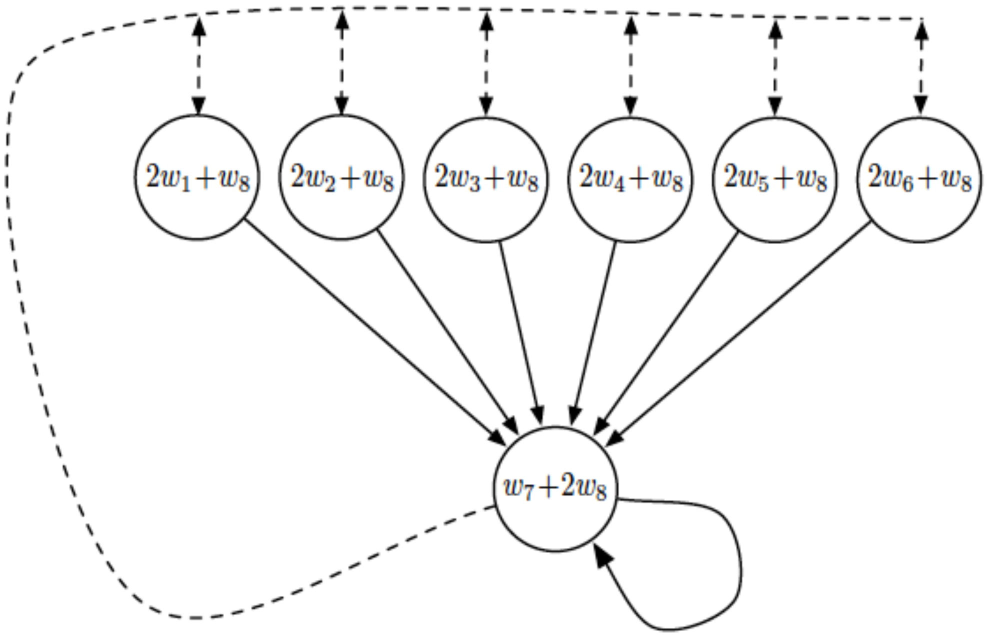
$$\delta_t \doteq R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) \hat{q}(S_{t+1}, a, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t), \text{ or} \qquad \text{(episodic)}$$

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \sum_a \pi(a|S_{t+1}) \hat{q}(S_{t+1}, a, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t). \qquad \text{(continuing)}$$

- Extension to n-step cases is straightforward

- Off-policy methods are with poor stability

Baird's counterexample



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

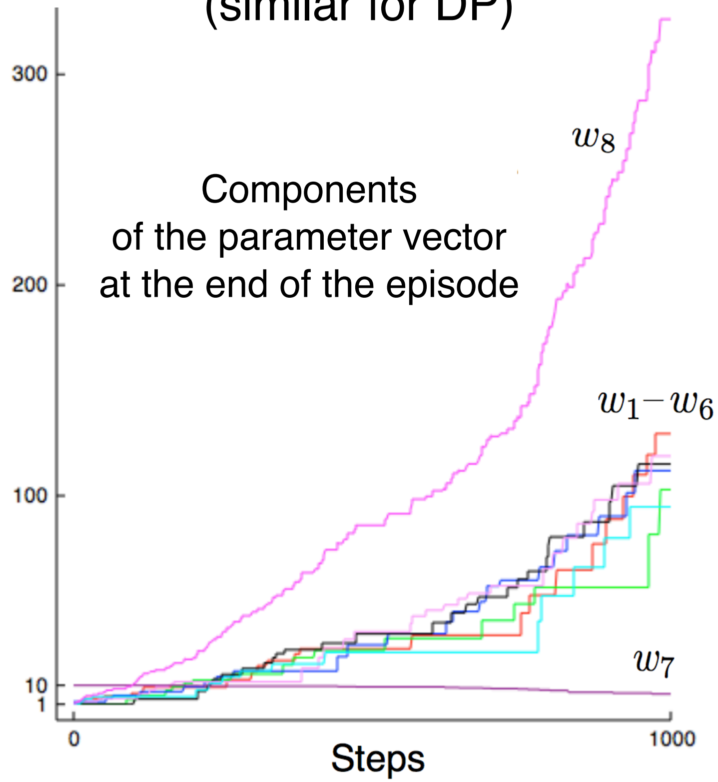
$\mu(\text{dashed}|\cdot) = 6/7$

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

$\gamma = 0.99$

reward = 0

Semi-gradient off-policy TD(0)
(similar for DP)



initial weights $w = (1, 1, 1, 1, 1, 1, 10, 1)^\top$

What causes the instability?

- It has nothing to do with learning or sampling
  - Even dynamic programming suffers from divergence with FA
- It has nothing to do with exploration, greedification, or control
  - Even prediction alone can diverge
- It has nothing to do with local minima or complex non-linear approximators
  - Even simple linear approximators can produce instability

The deadly triad

- The risk of divergence arises whenever we combine all the three things:

  1. Function approximation
     - significantly generalizing to large state space
  2. Bootstrapping
     - updating targets that include existing estimates, e.g. DP and TD
  3. Off-policy learning
     - training on a distribution of transitions other than that produced by the target policy, e.g. Q-learning and DP

Any 2 is ok,
3 is dangerous!

How to survive the deadly triad?

- Give up one of the three?
  - NO FA: we need scalability to large problem;
  - NO bootstrapping: critical to the computational and data efficiency

    - compare the time and memory requirement with Monte Carlo

    - on the other hand, bootstrapping introduces bias, which harms the asymptotic performance of approximate methods
  - NO off-policy: essential to learning multiple policies in parallel

To survive, we need to look into each of them and see how to modify it.

A Stable Case: Emphatic-TD Methods

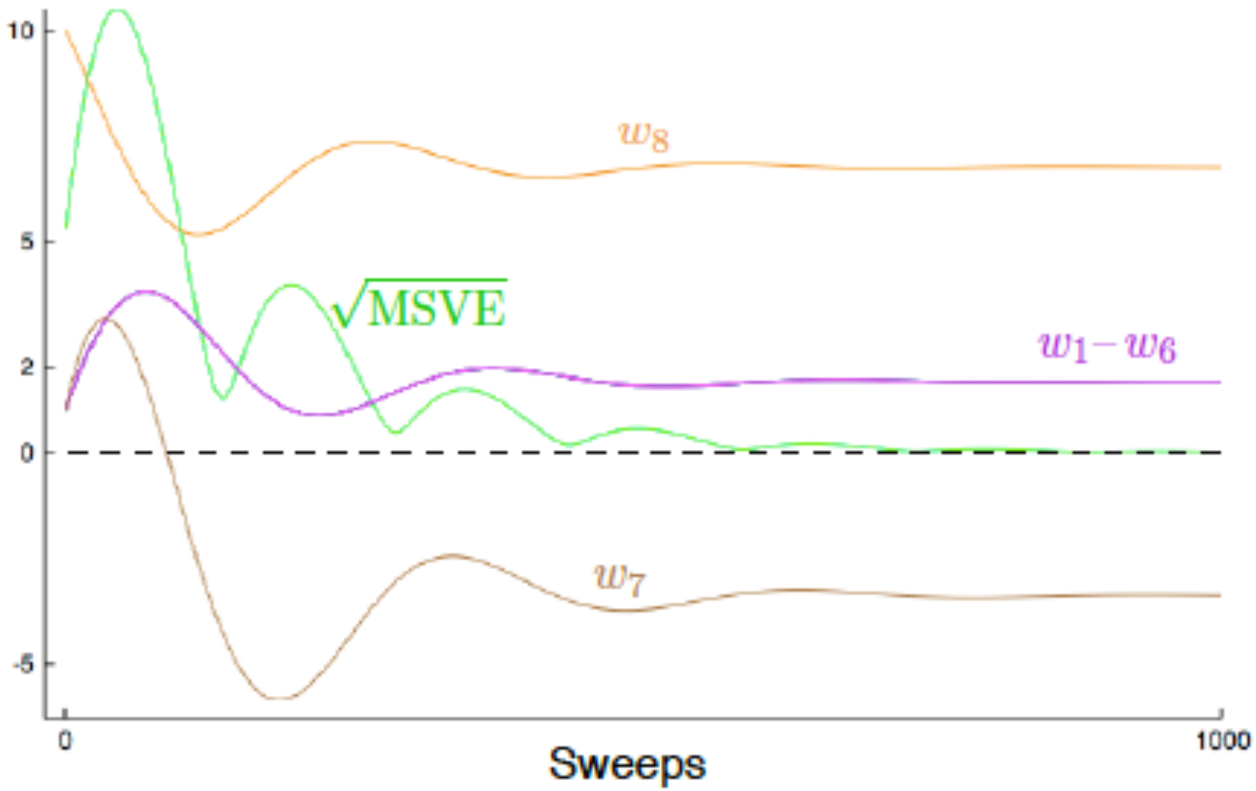
- State weightings are powerful
  - They are the difference between convergence and divergence in on-policy and off-policy TD learning
  - We can change the weighting by *emphasizing some steps* more than others in learning
- Some time steps are more important intrinsically → Interest I_t
  - e.g. early time steps in an episode
  - We want to control the the importance at each individual steps (intrinsically)
- Bootstrapping interacts with state importance → resultant Emphasis M_t
  - if the state is important, then it becomes important to accurately value the later states even if they are not important on their own.
  - If the state is not important, corresponding Emphasis contribution is zero

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

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha M_t \rho_t \delta_t \nabla \hat{v}\left(S_t, \boldsymbol{w}_t\right)$$

$$M_t = \gamma \rho_{t-1} M_{t-1} + I_t$$

Baird's Example with Emphatic-TD



Summary

- Function approximation with off-policy learning is possible
- But direct extensions from on-policy methods are not stable
- A promising approach with Emphatic TD
  - Converging
  - But with large variance
- Many on-going research efforts on this; a quite open field