REINFORCEMENT LEARNING: THEORY AND PRACTICE

Ch. 11: Off-policy Methods with Approximation

Profs. Amy Zhang and Peter Stone

Previously

Chapter 9 Policy Evaluation with Function Approximation
Chapter 10 On-policy Control with Approximation

Off-policy semi-gradient methods

Recap from Chapter 7, per-step importance sampling ratio:

$$\rho_t \doteq \rho_{t:t} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$

Off-policy semi-gradient methods

Recap from Chapter 7, per-step importance sampling ratio:

$$\rho_t \doteq \rho_{t:t} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$

Off-policy TD(0): $\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\rho_t\delta_t\nabla\hat{v}(S_t,\mathbf{w}_t)$

$\delta_t \doteq R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}_t) - \hat{v}(S_t,\mathbf{w}_t)$, or — Episodic and discounted

$\delta_t \doteq R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1},\mathbf{w}_t) - \hat{v}(S_t,\mathbf{w}_t)$. — Continuing and undiscounted

Off-policy semi-gradient methods

Recap from Chapter 7, per-step importance sampling ratio:

$$\rho_t \doteq \rho_{t:t} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$

Semi-gradient Expected Sarsa (for action values): No importance sampling!

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

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

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \sum \pi(a|S_{t+1})\hat{q}(S_{t+1}, a, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t).$$ Continuing and undiscounted

N-step Off-policy semi-gradient methods

Recap from Chapter 7, per-step importance sampling ratio:

$$\rho_t \doteq \rho_{t:t} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$

N-step semi-gradient Expected Sarsa (for action values): Importance sampling is back…

$$\mathbf{w}_{t+n} \doteq \mathbf{w}_{t+n-1} + \alpha\rho_{t+1}\cdots\rho_{t+n-1}\left[G_{t:t+n} - \hat{q}(S_t, A_t, \mathbf{w}_{t+n-1})\right]\nabla\hat{q}(S_t, A_t, \mathbf{w}_{t+n-1})$$

with

$$G_{t:t+n} \doteq R_{t+1} + \cdots + \gamma^{n-1}R_{t+n} + \gamma^n\hat{q}(S_{t+n}, A_{t+n}, \mathbf{w}_{t+n-1}), \text{ or}$$

Episodic and discounted

$$G_{t:t+n} \doteq R_{t+1} - \bar{R}_t + \cdots + R_{t+n} - \bar{R}_{t+n-1} + \hat{q}(S_{t+n}, A_{t+n}, \mathbf{w}_{t+n-1}),$$

Continuing and undiscounted

Reading Responses

[Karen Chen]
If function approximation is significantly more difficult for off-policy learning, what are cases where it's the most effective approach? Is the default to use on-policy learning with approximation?

Reading Responses

[Karen Chen]
If function approximation is significantly more difficult for off-policy learning, what are cases where it's the most effective approach? Is the default to use on-policy learning with approximation?
Off-policy learning is still useful because we can re-use data that we can't with on-policy methods.

Trade-off: is it cheaper to get on-policy samples or deal with the instability of off-policy learning?

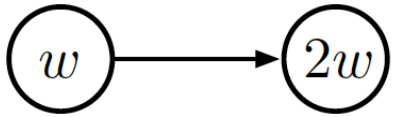
The deadly triad

Divergence is possible when all 3 parts of the deadly triad are present:

- Function approximation
- Bootstrapping
- Off-Policy training

Scott's slide

Off-policy semigradient methods



Stability of semigradient methods depends on on-policy distribution of updates. Why?

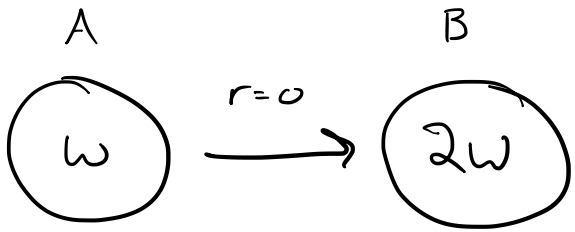
Imagine only updating one state S over and over again (i.e. off-policy):

- In tabular case, updating one state's value leaves all others changed
- With function approx + MC, multiple state values are updated, but V(S) is estimated independently of them via rewards only
- With function approx + TD (semigradient), multiple values are updated, which are then used to help estimate V(S) via bootstrapping, which are then updated again, which are then used to help estimates V(S)…

On-policy distribution forces state values to be "grounded" to something real

Scott's slide

Examples of Off-policy Divergence



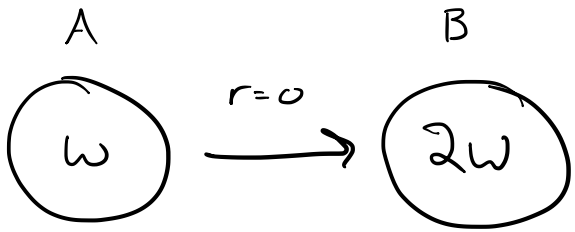
Init: $w_0 = 10$ Thus: $\hat{v}(A) = 10$, $\hat{v}(B) = 20$

Assume $\alpha = 0.5$, $\gamma = 0.9$

observe Transition from A to B

$$w_{t+1} = w_t + \alpha \rho \left[R_t + \gamma \hat{v}(B) - \hat{v}(A) \right] \nabla \hat{v}(A)$$

Scott's slide



Init: $w_0 = 10$ Thus: $\hat{v}(A) = 10$, $\hat{v}(B) = 20$

Assume $\alpha = 0.5$, $\gamma = 0.9$

observe Transition from A to B

$$w_{t+1} = w_t + \alpha \rho \left[R_t + \gamma \hat{v}(B) - \hat{v}(A) \right] \nabla \hat{v}(A)$$

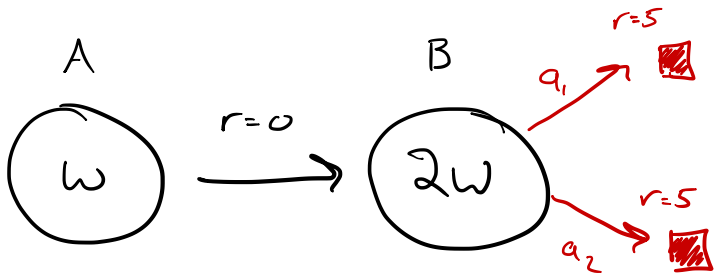
$$= 10 + (0.5)(1)\left[0 + .9(20) - 10 \right] \cdot 1$$

$$= 10 + 4$$

$$= 14$$

Thus: $\hat{v}(A) = 14$, $\hat{v}(B) = 28$

Scott's slide



- off policy ignores transition from B and diverges!
- on-policy uses transitions from B, which lowers $\hat{v}(B)$ and $\hat{v}(A)$ and converges

Init: $w_0 = 10$ Thus: $\hat{v}(A) = 10$, $\hat{v}(B) = 20$

Assume $\alpha = 0.5$, $\gamma = 0.9$

observe Transition from A to B

$$w_{t+1} = w_t + \alpha \rho \left[R_t + \gamma \hat{v}(B) - \hat{v}(A) \right] \nabla \hat{v}(A)$$

$$= 10 + (0.5)(1)\left[0 + .9(20) - 10 \right] \cdot 1$$

$$= 10 + 4$$

$$= 14$$

Thus: $\hat{v}(A) = 14$, $\hat{v}(B) = 28$

Scott's slide

Reading Responses

[Troy Dutton]
I'm confused why bootstrapping is part of the deadly triad. Why could a problem converge with monte carlo, but not with some bootstrapping method? Is it because the estimate Ut is dependent upon the weights? I thought that just meant we could only guarantee convergence to a local minima.

Reading Responses

[Troy Dutton]
I'm confused why bootstrapping is part of the deadly triad. Why could a problem converge with monte carlo, but not with some bootstrapping method? Is it because the estimate Ut is dependent upon the weights? I thought that just meant we could only guarantee convergence to a local minima.
Monte Carlo means we're updating only with real rewards, not value estimates, which bounds our error. With bootstrapping we get runaway growth.

Baird's Counterexample

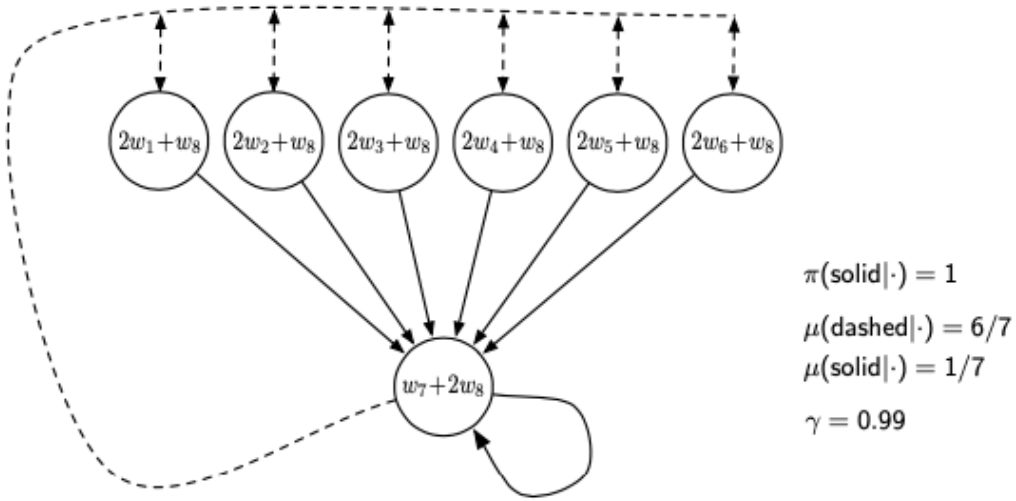
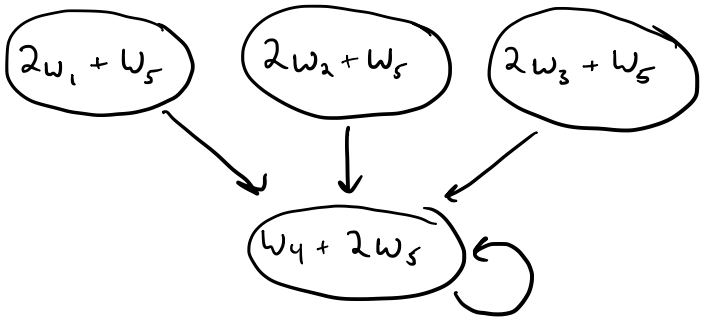


Figure 11.1: Baird's counterexample. The approximate state-value function for this Markov process is of the form shown by the linear expressions inside each state. The solid action usually results in the seventh state, and the dashed action usually results in one of the other six states, each with equal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



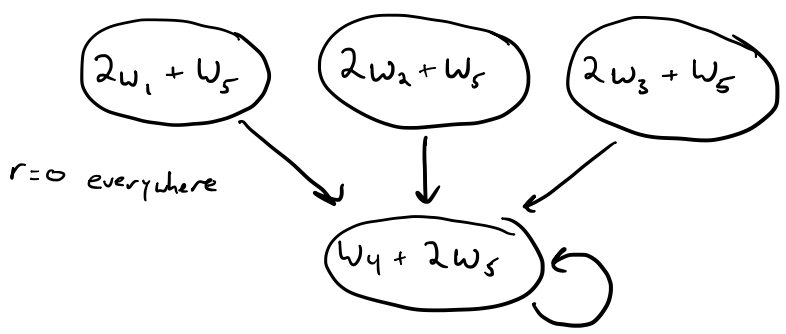
$\Rightarrow$

$$x: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1 \ 1 \ 1 \ 10 \ 1]^T$ then,

$$\hat{V} = [3 \ 3 \ 3 \ 12]^T$$

Scott's slide



$\Rightarrow$

$$X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1\ 1\ 1\ 10\ 1]^T$ then,

$\hat{V} = [3\ 3\ 3\ 12]^T$

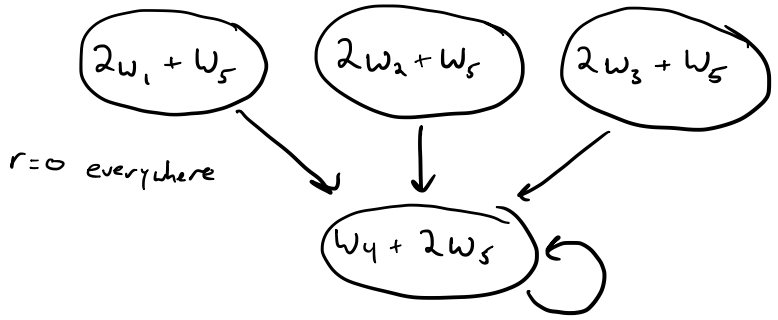
DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi[R_{t+1} + \gamma \hat{v}(S_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$

Step size alpha = 1

Discount factor gamma = 1

Scott's slide



$\Rightarrow$

$$X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1\ 1\ 1\ 10\ 1]^T$ then,

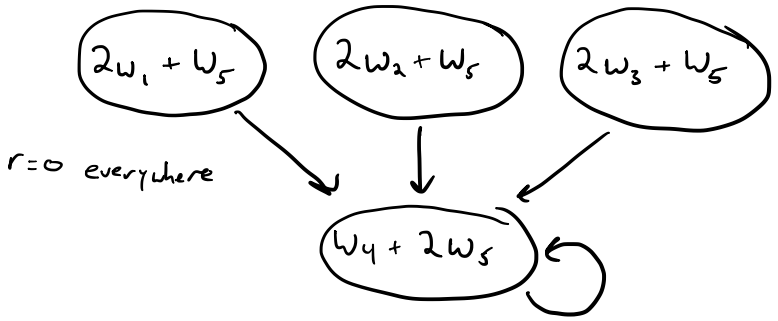
$\hat{V} = [3\ 3\ 3\ 12]^T$

DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi [R_{t+1} + \gamma \hat{v}(S_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$

$$= w_t + 1/4 \Big[(12-3)[2\ 0\ 0\ 0\ 1]^T + (12-3)[0\ 2\ 0\ 0\ 1]^T + (12-3)[0\ 0\ 2\ 0\ 1]^T + (12-12)[0\ 0\ 0\ 1\ 2]^T \Big]$$

Scott's slide



$r = 0$ everywhere

$\Rightarrow$

$$X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1\ 1\ 1\ 10\ 1]^T$ then,

$\hat{V} = [3\ 3\ 3\ 12]^T$

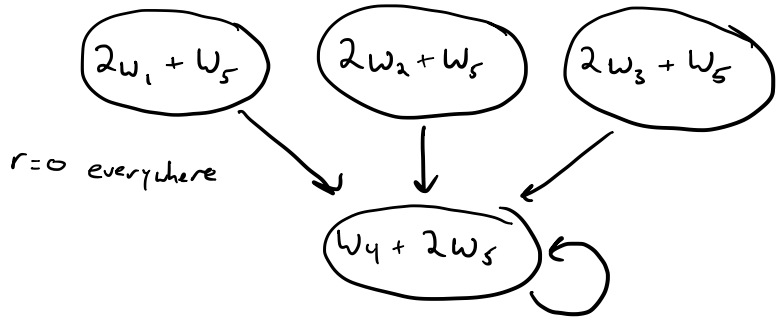
DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi [R_{t+1} + \gamma \hat{v}(S_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$

$$= w_t + 1/4 \Big[(12-3)[2\ 0\ 0\ 0\ 1]^T + (12-3)[0\ 2\ 0\ 0\ 1]^T + (12-3)[0\ 0\ 2\ 0\ 1]^T + (12-12)[0\ 0\ 0\ 1\ 2]^T \Big]$$

$$= [1\ 1\ 1\ 10\ 1]^T + 1/4 \cdot [18\ 18\ 18\ 0\ 54]^T$$

Scott's slide



$\Rightarrow$ X: $\begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$

if $w_0 = [1\ 1\ 1\ 10\ 1]^T$ then,

$\hat{V} = [3\ 3\ 3\ 12]^T$

DP update:

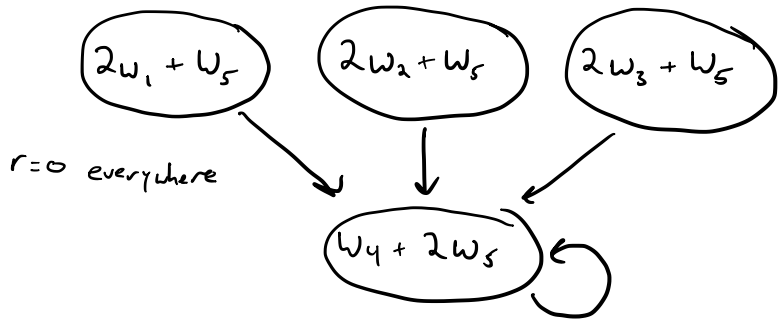
$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi[R_{t+1} + \gamma \hat{v}(S_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$

$$= w_t + 1/4 \Big[(12-3)[2\ 0\ 0\ 0\ 1]^T + (12-3)[0\ 2\ 0\ 0\ 1]^T + (12-3)[0\ 0\ 2\ 0\ 1]^T + (12-12)[0\ 0\ 0\ 1\ 2]^T \Big]$$

$$= [1\ 1\ 1\ 10\ 1]^T + 1/4 \cdot [18\ 18\ 18\ 0\ 54]^T$$

$$= [5.5\ 5.5\ 5.5\ 10\ 15]^T$$

Scott's slide



$\Rightarrow$ X: $\begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$

if $w_0 = [1\ 1\ 1\ 10\ 1]^T$ then,

$\hat{V} = [3\ 3\ 3\ 12]^T$

DP update:

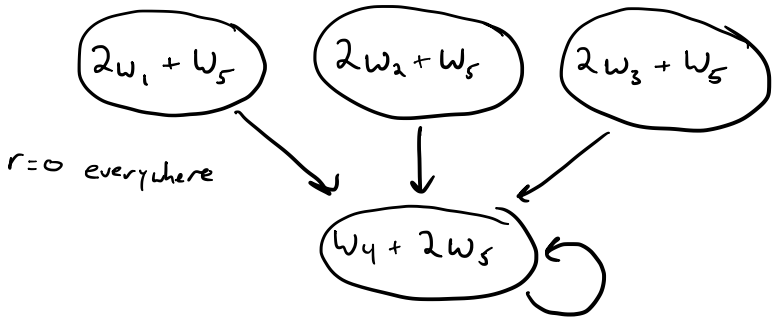
$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi [R_{t+1} + \gamma \hat{v}(S_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$

$$= w_t + 1/4 \Big[(12-3)[2\ 0\ 0\ 0\ 1]^T + (12-3)[0\ 2\ 0\ 0\ 1]^T + (12-3)[0\ 0\ 2\ 0\ 1]^T + (12-12)[0\ 0\ 0\ 1\ 2]^T \Big]$$

$$= [1\ 1\ 1\ 10\ 1]^T + 1/4 \cdot [18\ 18\ 18\ 0\ 54]^T$$

$$= [5.5\ 5.5\ 5.5\ 10\ 15]^T \Rightarrow \hat{v} = [26\ 26\ 26\ 40]^T$$

Scott's slide



$$\Rightarrow \quad X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1 \; 1 \; 1 \; 10 \; 1]^T$ then,

$\hat{V} = [3 \; 3 \; 3 \; 12]^T$

DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi [R_{t+1} + \gamma \hat{v}(S_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$

$$= w_t + 1/4 \Big[(12-3)[2\,0\,0\,0\,1]^T + (12-3)[0\,2\,0\,0\,1]^T + (12-3)[0\,0\,2\,0\,1]^T + (12-12)[0\,0\,0\,1\,2]^T \Big]$$

$$= [1\;1\;1\;10\;1]^T + 1/4 \cdot [18\;18\;18\;0\;54]^T$$

$$= [5.5\;5.5\;5.5\;10\;15]^T \quad \Rightarrow \quad \hat{v} = [26\;26\;26\;40]^T$$

- DP updates with off-policy state dist (uniform), but policy π always follows solid lines
- Updates top states more often than it should
- Bottom state looks better and raises value of top state
- ... which raises value of bottom state, which ...

Scott's slide

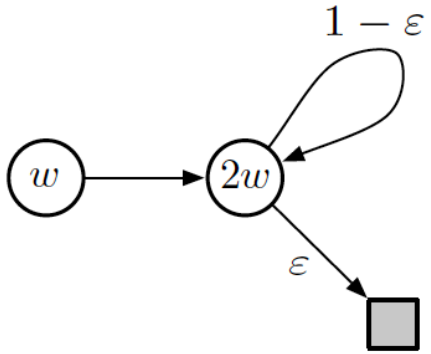
Reading Responses

[Krystal An]
In the Fig.11.1 How does the behavior policy b contribute to the divergence in Baird's counterexample, and what does this imply about the importance of policy choices in function approximation settings?

Reading Responses

[Krystal An]
In the Fig.11.1 How does the behavior policy b contribute to the divergence in Baird's counterexample, and what does this imply about the importance of policy choices in function approximation settings?
Divergence occurs because our target policy concentrates on the bottom state while DP updates all states uniformly. These policies are not the same, and hence DP gives an off-policy update. If we updated according to the current policy, we would converge.

Example 11.1: Tsitsiklis and Van Roy's Counterexample This example shows that linear function approximation would not work with DP even if the least-squares solution was found at each step. The counterexample is formed by extending the w-to-$2w$ example (from earlier in this section) with a terminal state, as shown to the right. As before, the estimated value of the first state is w, and the estimated value of the second state is $2w$. The reward is zero on all transitions, so the true values are zero at both states, which is exactly representable with $w = 0$. If we set w_{k+1} at each step so as to minimize the $\overline{\text{VE}}$ between the estimated value and the expected one-step return, then we have



$$\begin{aligned} w_{k+1} &= \operatorname*{arg\,min}_{w\in\mathbb{R}} \sum_{s\in\mathcal{S}} \Big(\hat{v}(s,w) - \mathbb{E}_\pi\big[R_{t+1} + \gamma\hat{v}(S_{t+1},w_k) \;\big|\; S_t = s\big]\Big)^2 \\ &= \operatorname*{arg\,min}_{w\in\mathbb{R}} \big(w - \gamma 2w_k\big)^2 + \big(2w - (1-\varepsilon)\gamma 2w_k\big)^2 \\ &= \frac{6-4\varepsilon}{5}\gamma w_k. \end{aligned} \tag{11.10}$$

The sequence $\{w_k\}$ diverges when $\gamma > \frac{5}{6-4\varepsilon}$ and $w_0 \neq 0$. ■

Reading Responses

[Saloni Modi]
Is there a reason we have empirical evidence but no theoretical analysis for why Q-learning doesn't diverge when the behavior policy is sufficiently close to the target policy (ex: when it's the epsilon-greedy policy) (referring to page 263, section 11.2)?

There are also counterexamples similar to Baird's showing divergence for Q-learning. This is cause for concern because otherwise Q-learning has the best convergence guarantees of all control methods. Considerable effort has gone into trying to find a remedy to this problem or to obtain some weaker, but still workable, guarantee. For example, it may be possible to guarantee convergence of Q-learning as long as the behavior policy is sufficiently close to the target policy, for example, when it is the ε-greedy policy. To the best of our knowledge, Q-learning has never been found to diverge in this case, but there has been no theoretical analysis. In the rest of this section we present several other ideas

Reading Responses

[Saloni Modi]
Is there a reason we have empirical evidence but no theoretical analysis for why Q-learning doesn't diverge when the behavior policy is sufficiently close to the target policy (ex: when it's the epsilon-greedy policy) (referring to page 263, section 11.2)?
Not necessarily!

Near Optimal Provable Uniform Convergence in Off-Policy Evaluation for Reinforcement Learning

Ming Yin [1,3], Yu Bai[2], and Yu-Xiang Wang[3]

[1]Department of Statistics and Applied Probability, UC Santa Barbara
[2]Salesforce Research
[3]Department of Computer Science, UC Santa Barbara
ming_yin@ucsb.edu yu.bai@salesforce.com yuxiangw@cs.ucsb.edu

On Convergence of Average-Reward Off-Policy Control Algorithms in Weakly Communicating MDPs

Yi Wan WAN6@UALBERTA.CA
University of Alberta
Richard S. Sutton RSUTTON@UALBERTA.CA
University of Alberta, DeepMind

Doubly Robust Off-Policy Actor-Critic: Convergence and Optimality

Tengyu Xu [1] **Zhuoran Yang** [2] **Zhaoran Wang** [3] **Yingbin Liang** [1]

Dr Jekyll and Mr Hyde: The Strange Case of Off-Policy Policy Updates

Romain Laroche* Microsoft Research Montréal, Canada
Rémi Tachet des Combes* Microsoft Research Montréal, Canada

The projection matrix

For a linear function approximator, the projection operation is linear, which implies that it can be represented as an $|\mathcal{S}| \times |\mathcal{S}|$ matrix:

$$\Pi \doteq \mathbf{X}\left(\mathbf{X}^\top \mathbf{D}\mathbf{X}\right)^{-1}\mathbf{X}^\top \mathbf{D}, \tag{11.13}$$

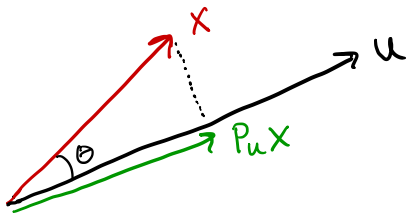
where, as in Section 9.4, $\mathbf{D}$ denotes the $|\mathcal{S}| \times |\mathcal{S}|$ diagonal matrix with the $\mu(s)$ on the diagonal, and $\mathbf{X}$ denotes the $|\mathcal{S}| \times d$ matrix whose rows are the feature vectors $\mathbf{x}(s)^\top$, one for each state s. If the inverse in does not exist, then the pseudoinverse is substituted. Using these matrices, the norm of a vector can be written

$$\|v\|_\mu^2 = v^\top \mathbf{D} v, \tag{11.14}$$

and the approximate linear value function can be written

$$v_{\mathbf{w}} = \mathbf{X}\mathbf{w}. \tag{11.15}$$

Orthogonal Projection

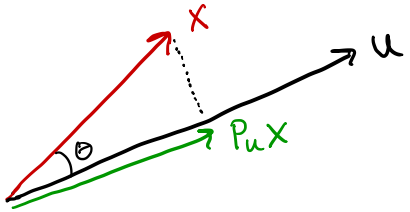


Projection to a unit vector : $P_u = uu^T$

So: $P_u x = uu^T x$

Scott's slide

Orthogonal Projection



Projection to a unit vector : $P_u = uu^T$

So: $P_u x = uu^T x$

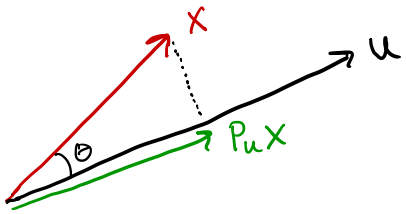
Why?

$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

Scott's slide

Orthogonal Projection



Projection to a unit vector : $P_u = uu^T$

So: $P_u x = uu^T x$

Why?

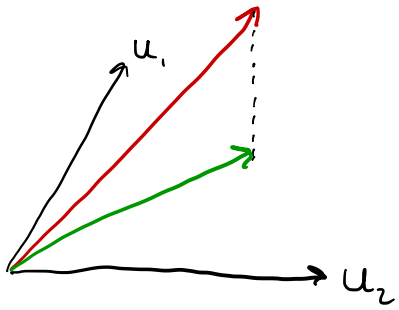
$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

More Generally:

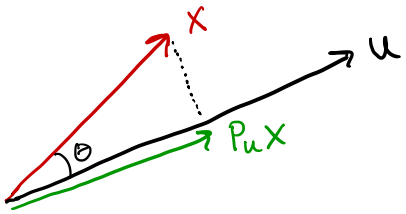
if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$P_A = AA^T$



Scott's slide

Orthogonal Projection



Projection to a unit vector: $P_u = uu^T$

So: $P_u x = uu^T x$

Why?

$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

What if $u_1 \dots u_k$ not orthonormal?

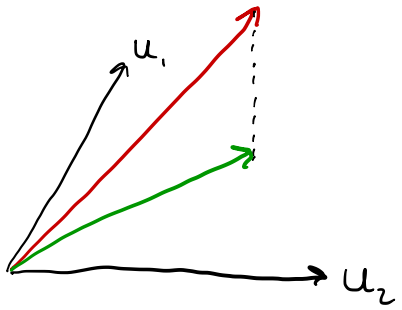
$$P_A = A(A^T A)^{-1} A^T$$

normalizing factor

More Generally:

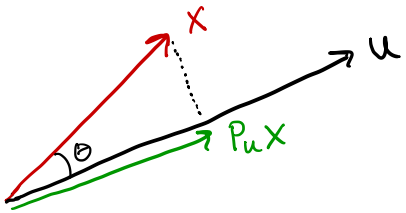
if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$P_A = AA^T$



Scott's slide

Orthogonal Projection



Projection to a unit vector: $P_u = uu^T$

So: $P_u x = uu^T x$

Why?

$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

What if $u_1 \dots u_k$ not orthonormal?

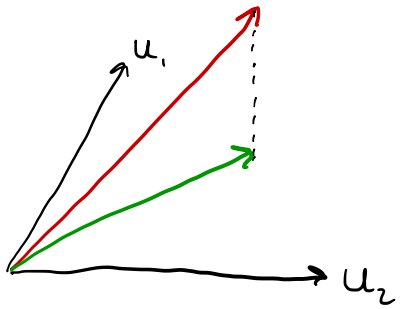
$$P_A = A(A^T A)^{-1} A^T$$

normalizing factor

More Generally:

if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$$P_A = AA^T$$



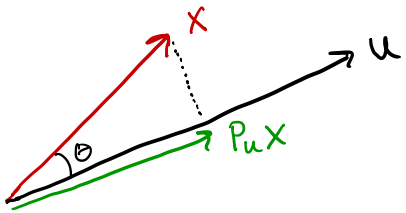
What about other inner products?

$$P_A = A(A^T D A)^{-1} A^T D$$

$\langle x, y \rangle = y^T D x$

Scott's slide

Orthogonal Projection



Projection to a unit vector: $P_u = uu^T$

So: $P_u x = uu^T x$

Why?

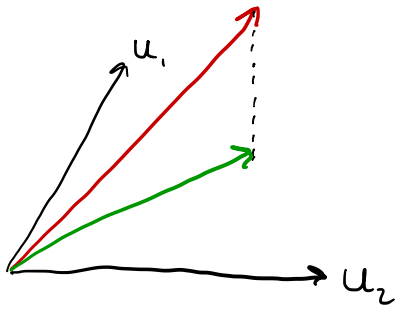
$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

More Generally:

if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$P_A = AA^T$



What if $u_1 \dots u_k$ not orthonormal?

$P_A = A(A^TA)^{-1}A^T$

normalizing factor

Linear regression:

$\hat{y} = X(x^Tx)^{-1}x^Ty$

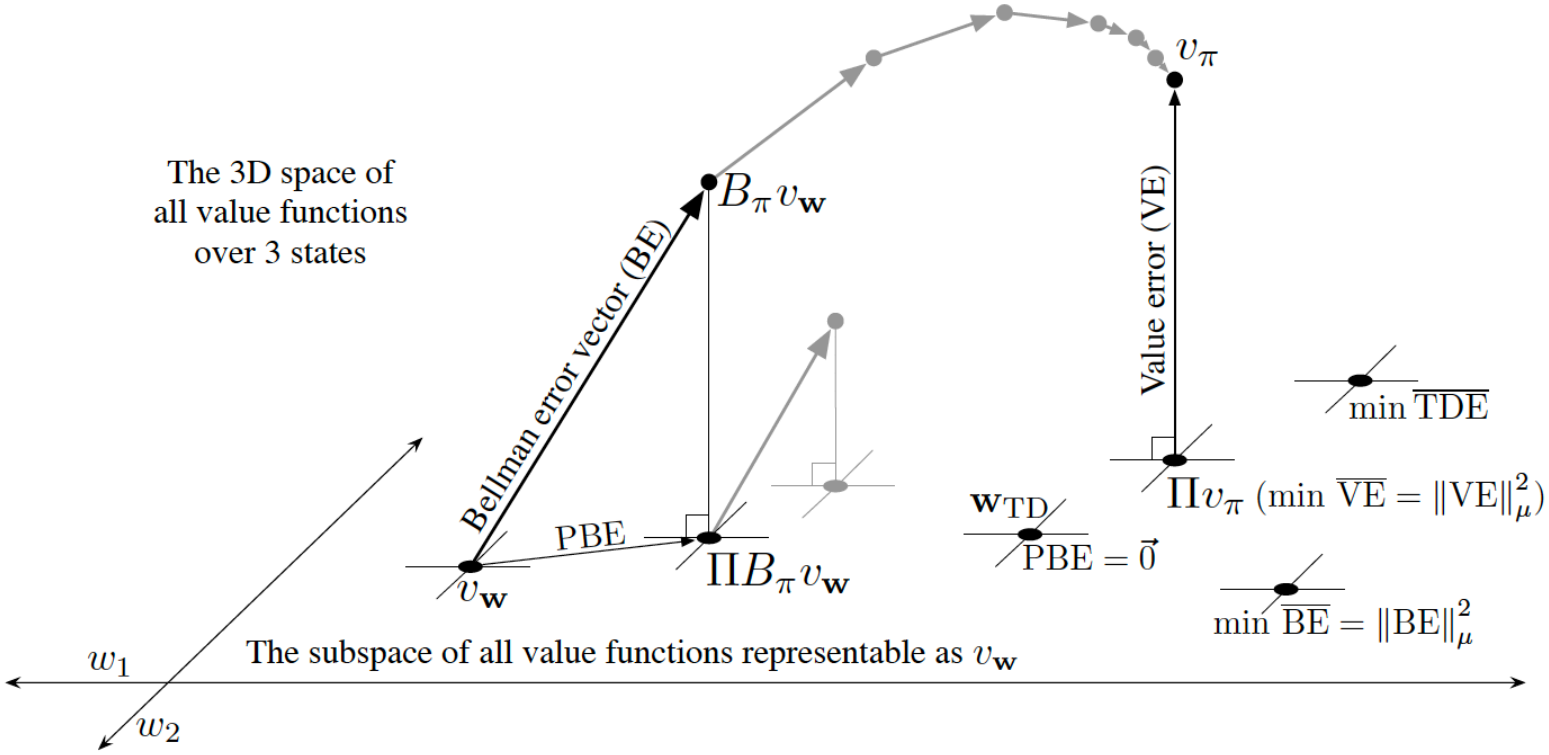
What about other inner products?

$P_A = A(A^TDA)^{-1}A^TD$

$\langle x, y \rangle = y^TDx$

Scott's slide

Linear Value-Function Geometry



Linear Value-Function Geometry

Given two value functions, the vector difference:

$$v = v_1 - v_2$$

Linear Value-Function Geometry

Given two value functions, the vector difference:

$$v = v_1 - v_2.$$

But not all states are considered equal! Taking into account on-policy distribution:

$$\|v\|_\mu^2 \doteq \sum_{s \in \mathcal{S}} \mu(s) v(s)^2$$

Linear Value-Function Geometry

Given two value functions, the vector difference:

$$v = v_1 - v_2.$$

But not all states are considered equal! Taking into account on-policy distribution:

$$\|v\|_\mu^2 \doteq \sum_{s \in \mathcal{S}} \mu(s) v(s)^2$$

Mean Squared Value Error (from Ch 9.2):

$$\overline{\mathrm{VE}}(\mathbf{w}) = \|v_{\mathbf{w}} - v_\pi\|_\mu^2.$$

Linear Value-Function Geometry

Given two value functions, the vector difference:

$$v = v_1 - v_2.$$

But not all states are considered equal! Taking into account on-policy distribution:

$$\|v\|_\mu^2 \doteq \sum_{s \in \mathcal{S}} \mu(s) v(s)^2$$

Mean Squared Value Error (from Ch 9.2):

$$\overline{\mathrm{VE}}(\mathbf{w}) = \|v_{\mathbf{w}} - v_\pi\|_\mu^2.$$

Solution found by Monte Carlo! (Slow)

Linear Value-Function Geometry

TD methods:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \left[r + \gamma v_\pi(s') \right], \qquad \text{for all } s \in \mathcal{S}.$$

Linear Value-Function Geometry

TD methods:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \left[r + \gamma v_\pi(s') \right], \qquad \text{for all } s \in \mathcal{S}.$$

Bellman error:

$$\begin{aligned} \bar{\delta}_{\mathbf{w}}(s) &\doteq \left(\sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \left[r + \gamma v_{\mathbf{w}}(s') \right] \right) - v_{\mathbf{w}}(s) \\ &= \mathbb{E}\left[R_{t+1} + \gamma v_{\mathbf{w}}(S_{t+1}) - v_{\mathbf{w}}(S_t) \;\middle|\; S_t = s, A_t \sim \pi \right], \end{aligned}$$

Bellman error is expectation of TD error!

Linear Value-Function Geometry

TD methods:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \left[r + \gamma v_\pi(s') \right], \qquad \text{for all } s \in \mathcal{S}.$$

Bellman error:

$$\begin{aligned} \bar{\delta}_{\mathbf{w}}(s) &\doteq \left(\sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \left[r + \gamma v_{\mathbf{w}}(s') \right] \right) - v_{\mathbf{w}}(s) \\ &= \mathbb{E}\left[R_{t+1} + \gamma v_{\mathbf{w}}(S_{t+1}) - v_{\mathbf{w}}(S_t) \;\middle|\; S_t = s, A_t \sim \pi \right], \end{aligned}$$

Mean Squared Bellman Error:

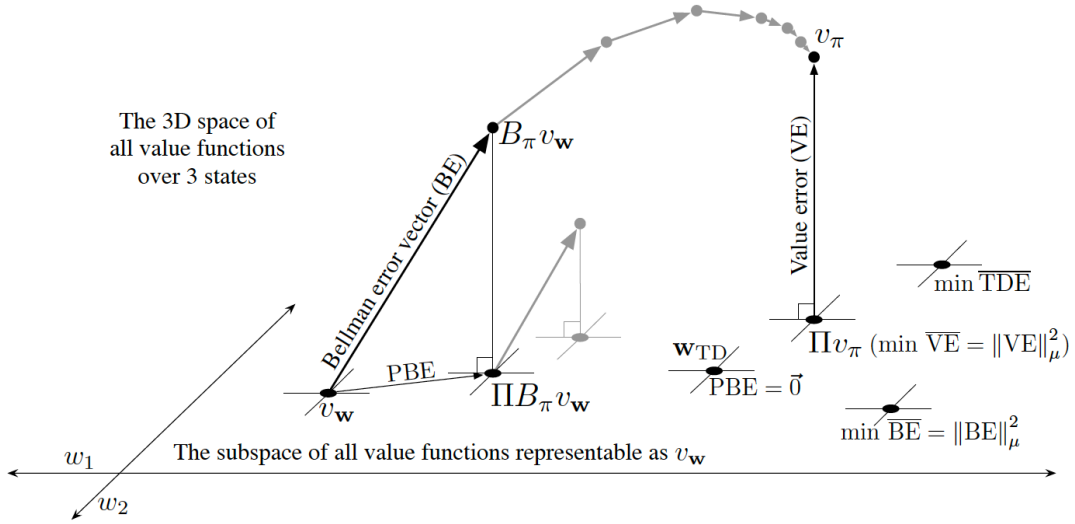
$$\overline{\mathrm{BE}}(\mathbf{w}) = \left\| \bar{\delta}_{\mathbf{w}} \right\|_\mu^2 .$$

Linear Value-Function Geometry

Intermediate value functions lie outside the subspace we can represent.

Mean Squared Projected Bellman error:

$$\overline{\text{PBE}}(\mathbf{w}) = \left\|\Pi\bar{\delta}_{\mathbf{w}}\right\|_{\mu}^{2}.$$



Reading Responses

[Jiaheng Hu]
It is relatively well-known that DQN is not guaranteed to converge but Q-learning is. Do we have any understanding about what kind of function approximations can converge and what cannot? It sounds like there might be some middle ground where the function approximation is more powerful than the tabular case, but restricted such that convergence is still guaranteed?

Reading Responses

[Jiaheng Hu]
It is relatively well-known that DQN is not guaranteed to converge but Q-learning is. Do we have any understanding about what kind of function approximations can converge and what cannot? It sounds like there might be some middle ground where the function approximation is more powerful than the tabular case, but restricted such that convergence is still guaranteed?
Showed that linear cannot. More powerful function approximation seems like it would help, but we don't know how to think about the "geometry" anymore.

Reading Responses

[Jeffrey Lai]
What are some partial modern solutions to avoid the deadly triad? It seems like function approximation and bootstrapping are essential to practical problems but are there ways to independently address either of these to alleviate their negative interaction?

Reading Responses

[Jeffrey Lai]
What are some partial modern solutions to avoid the deadly triad? It seems like function approximation and bootstrapping are essential to practical problems but are there ways to independently address either of these to alleviate their negative interaction?

Prevent extrapolation outside data support:

1. regularization to keep policy close to behavior policy (mitigating off-policy)
2. suppress out-of-distribution value estimates (mitigating bootstrapping)
3. Powerful function approximation (to keep BE and PBE close)

Reading Responses

[Jasper Lee]
Section 11.4 mentions several different types of errors that different algorithms try to minimize. Can anything be said about which types of errors are better to minimize in practice for which types of RL problems? Or is everything still in the air?

Reading Responses

[Jasper Lee]
Section 11.4 mentions several different types of errors that different algorithms try to minimize. Can anything be said about which types of errors are better to minimize in practice for which types of RL problems? Or is everything still in the air?
What error we're minimizing is dictated by what method we're using.
Typically it's TD error because Monte Carlo is too slow.

Final Logistics

Next week:
Chapter 12: Eligibility Traces
Reading assignments due **2PM Monday**

Office hours:
Mon: Michael 1-2PM GDC Basement TA Station #5
Tues: Caroline 11:15-12:15PM
Wed: Amy 2-3PM EER 6.878
Thurs: Haoran 11-12PM; Siddhant 5-6PM
Fri: Shuozhe 4-5PM

Final Logistics

Final project proposal due at **11:59pm on Thursday, 3/7**

Complete Homework for Chapters 10+11 on edx by **Friday 11:59 PM CST**

Complete Programming Assignment for Chapters 4,5,6,7 on edx by **Sunday at 11:59 PM CST**