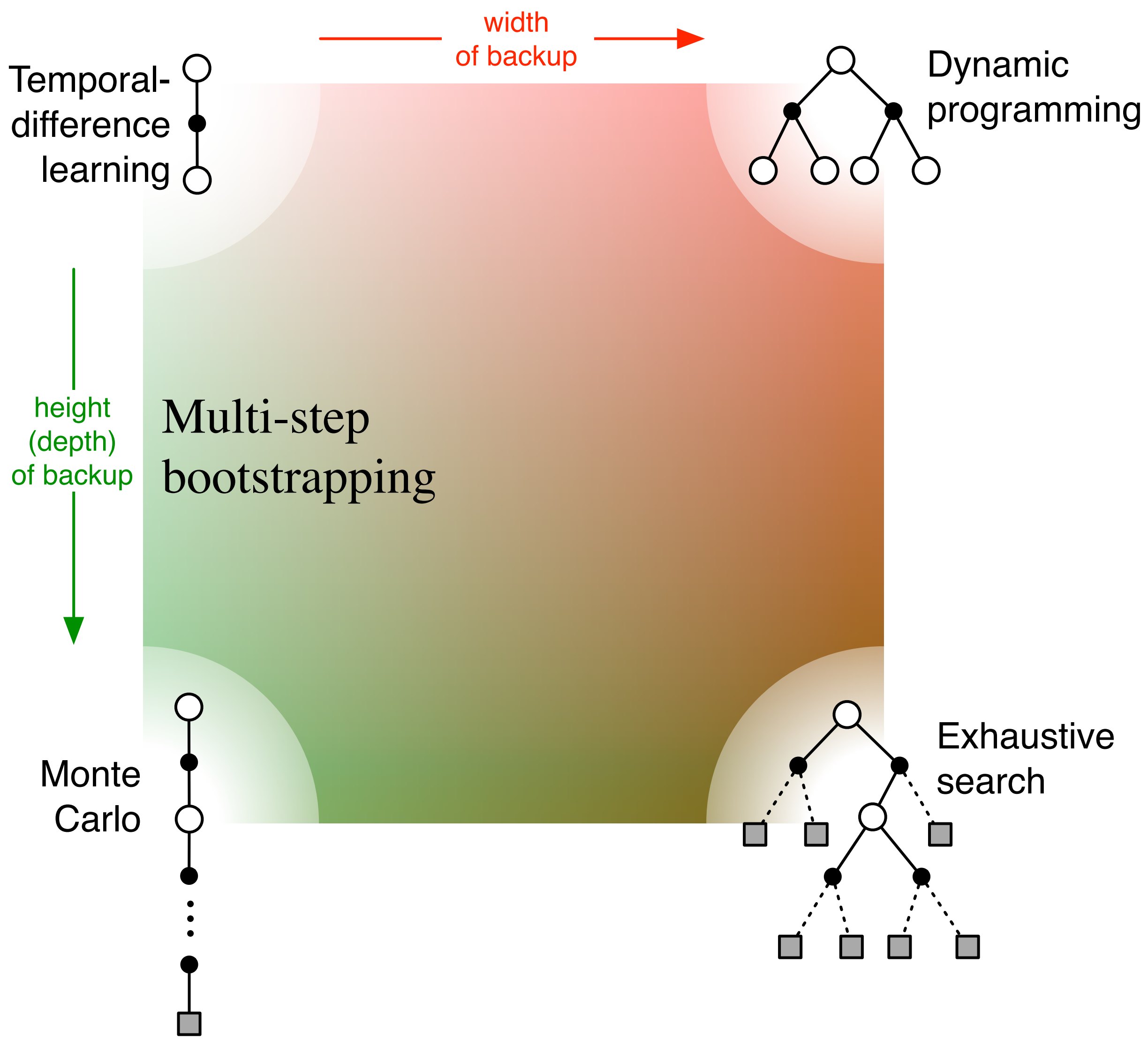
Lecture 9: Why do policy evaluation algorithms work? Control

Recall: TD-family for policy evaluation

- The TD family of methods is between MC and DP
- Interpolating in terms of credit assignment length!
- With bootstrapping (TD), we don’t get true gradient descent methods with function approximation
  - this complicates the analysis
  - but learning is can be *much faster*

Recall: Unified View



Recall: Different Targets

- Monte Carlo: $G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$
- TD: $G_t^{(1)} \doteq R_{t+1} + \gamma V_t(S_{t+1})$
  - Use V_t to estimate remaining return
- *n*-step TD:
  - 2 step return: $G_t^{(2)} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 V_t(S_{t+2})$
  - *n*-step return: $G_t^{(n)} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_t(S_{t+n})$

    with $G_t^{(n)} \doteq G_t \text{ if } t + n \geq T$

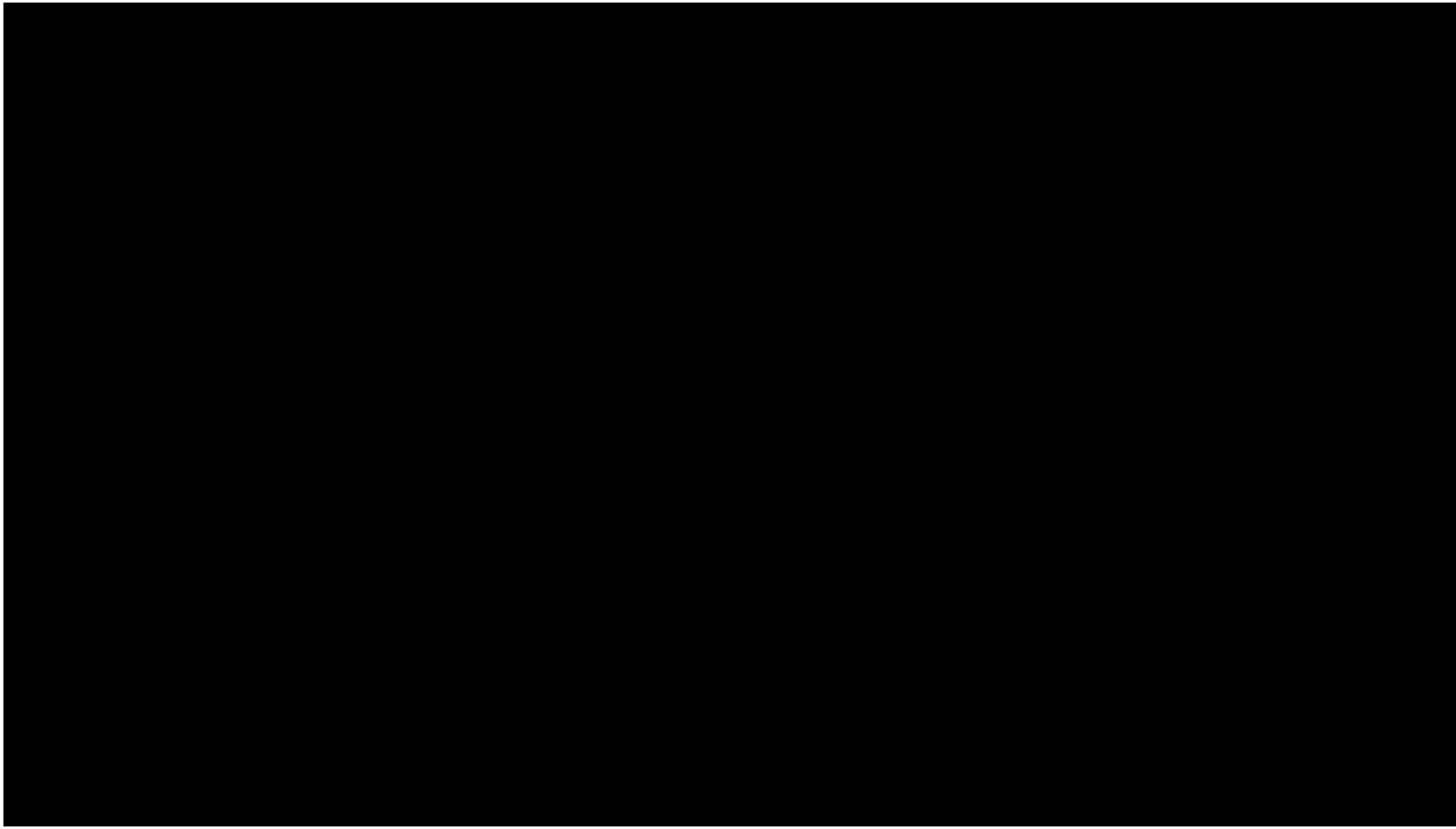
Do policy evaluation methods converge?

- If so, under what circumstances? To what solution?
  - DP policy evaluation in the tabular case
  - TD methods in the tabular case
  - TD methods with function approximation
  - MC with function approximation

Setup: Finite MDPs

- Recall in general expected reward is a function $r(s, a)$
- This can be represented as a matrix: $r \in \mathbb{R}^{|S| \times |A|}$
- Transitions $P(s' \mid s, a)$ can be represented as a matrix: $P \in \mathbb{R}^{|S| \times |A| \times |S|}$
- Suppose we have a fixed policy
- Policy can be represented as a matrix containing, for every state, a row containing $\pi(a \mid s)$: $\pi \in \mathbb{R}^{|S| \times |A|}$
- (Reason for this coming soon)
- Value function can be represented as a vector of size number of states: $v_\pi \in \mathbb{R}^{|S|}$

Example



Bellman equation in vector form

- Let $P_\pi = \text{ np.einsum}('sa, sax \rightarrow sx', \pi, P)$ be a $|S| \times |S|$ matrix of probabilities of transitions between states under policy π
- Let $r_\pi = \text{np.einsum}('sa, sa \rightarrow s', \pi, r)$ be a size $|S|$ column vector representing the expected immediate reward from every state
- The Bellman equation for policy evaluation can then be re-written as: $v_\pi = r_\pi + \gamma P_\pi v_\pi$
- As discussed before, is we know the model (r_π, P_π), this is a linear system of equations
- This system of equations has a unique solution: $v_\pi = (I - \gamma P_\pi)^{-1} r_\pi$
- The inverse exists because P_π is a stochastic matrix (rows

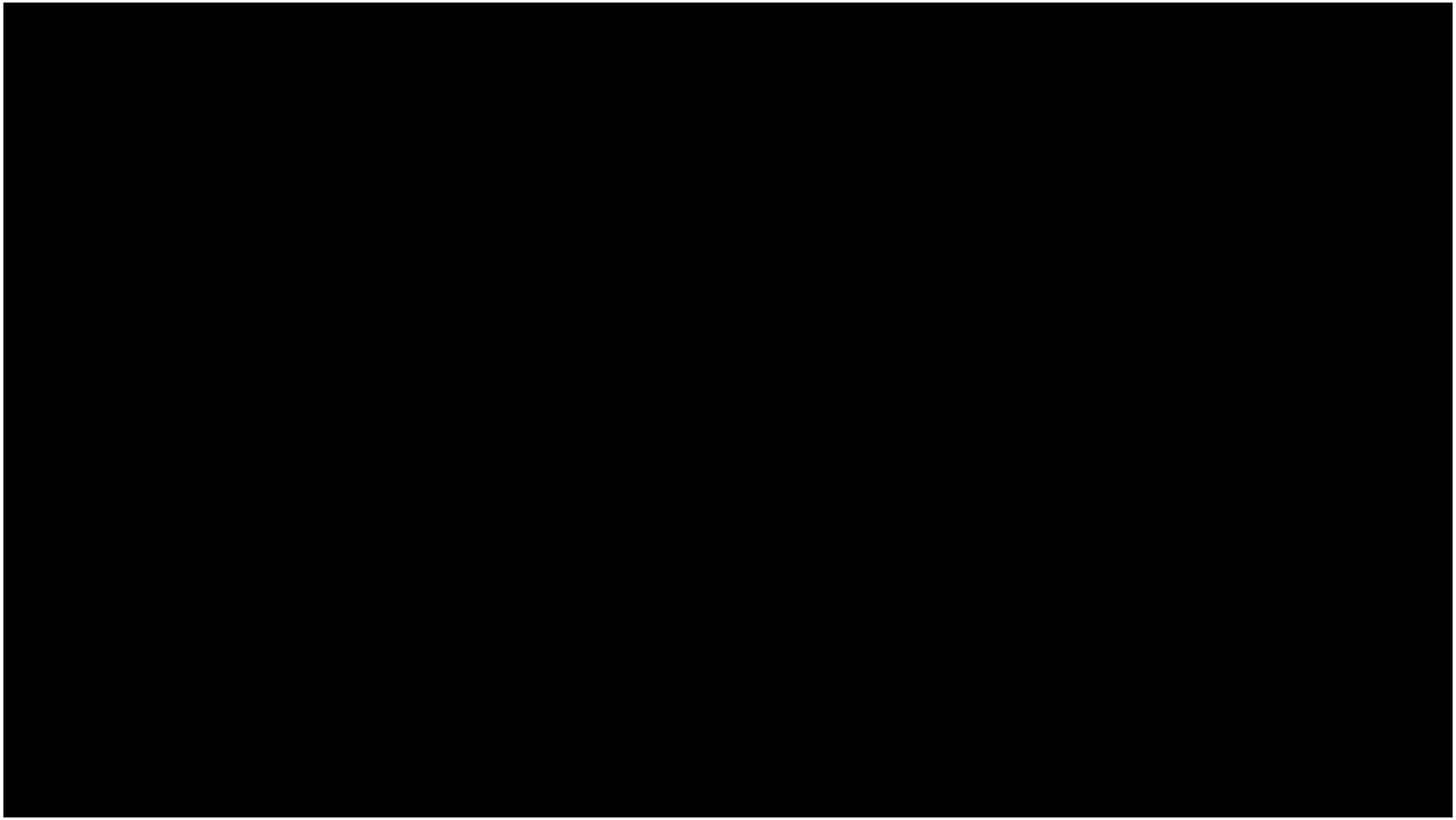
DP for policy evaluation

- The algorithm we had before can be summarized as:

  $$v_0 = 0$$

  Repeat: $v_{k+1} = r_\pi + \gamma P_\pi v_k$
- Does the value function end up approximating the true unique value function v_π?

Example: Bellman update



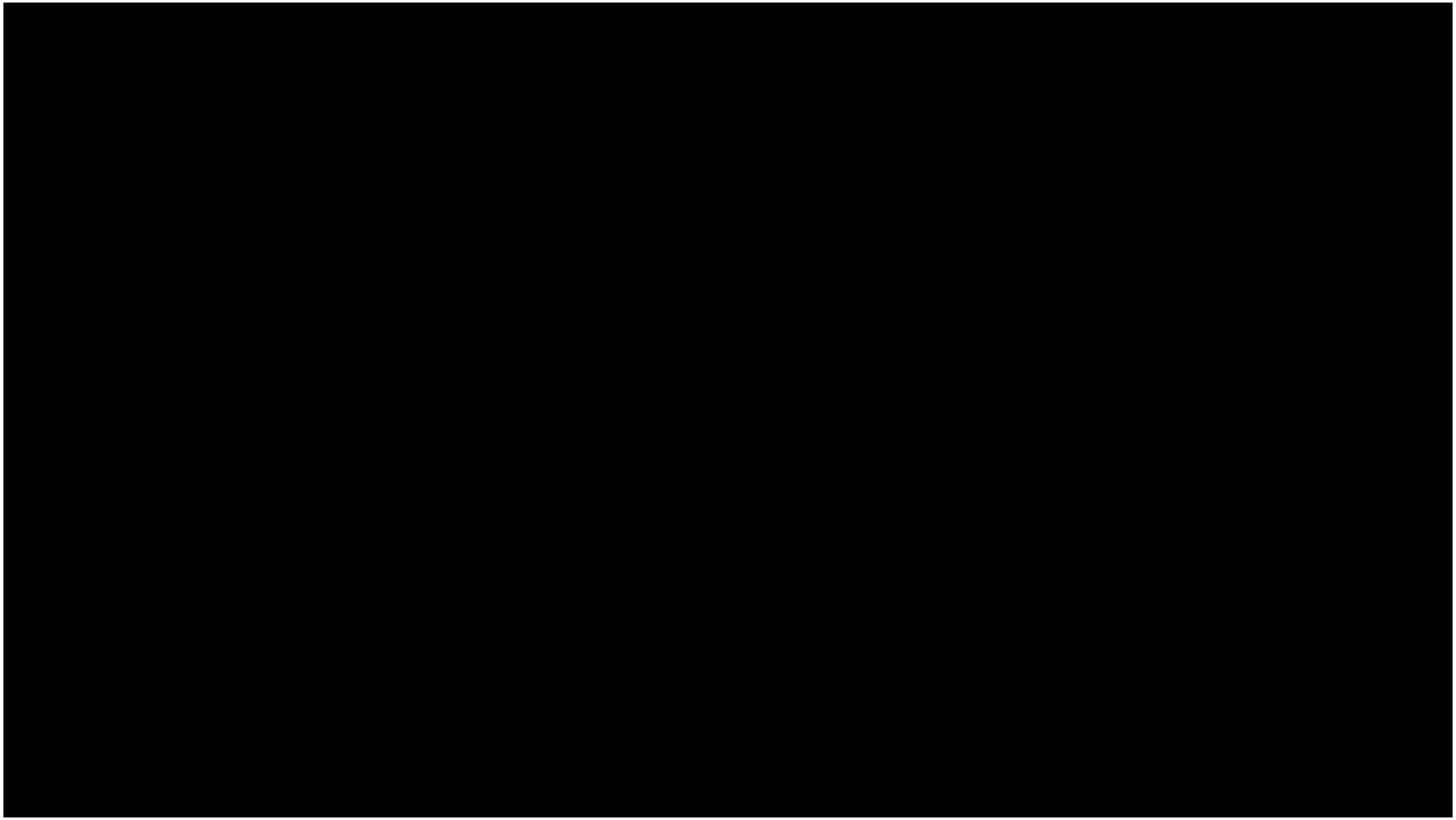
∞-norm

- If we have two vectors u and v, we can define their distance as the largest absolute difference in corresponding values: $||u - v||_\infty = \max_{s \in |S|} |u(s) - v(s)|$
- Very useful for analyzing stability of algorithms!
- If the max difference is decreasing as we consider vectors obtained through iteration, then all other differences are also decreasing
- So we will have convergence!

DP policy evaluation convergence

- We know $v_\pi = r_\pi + \gamma P_\pi v_\pi$
- Subtract from this the DP update: $v_{k+1} = r_\pi + \gamma P_\pi v_k$
- We have:
- $$\begin{aligned} ||v_\pi - v_{k+1}||_\infty &= ||r_\pi + \gamma P_\pi v_\pi - r_\pi - \gamma P_\pi v_k||_\infty \\ &= ||\gamma P_\pi (v_\pi - v_k)||_\infty \\ &\le ||\gamma P_\pi || |v_\pi - v_k||_\infty ||_\infty \\ &\le \gamma ||v_\pi - v_k||_\infty \end{aligned}$$
- So the ∞-norm of the error at each iteration decreases by at least a factor of γ!
- This is called a *contraction*
- By induction, at iteration k: $||v_\pi - v_{k+1}||_\infty \le \gamma^k ||v_\pi - v_0||_\infty$
- So the error becomes 0 in the limit! And decreases fast

Example



What about TD?

- Every time you are in state s you update with a sample target: $R + \gamma V(s')$
- What is the *expected value of the target*?
- $\mathbb{E}_\pi \left[R + \gamma V(s') \,|\, s \right] = r_\pi(s' \,|\, s) + \gamma \sum_{s'} P_\pi(s' \,|\, s) V(s')$
- So the expected target is the same as for DP!
- *Therefore, contraction argument applies for the expected TD update!*
- Footnote: to show convergence of the incremental algorithm we also need to show that updates have finite variance, and impose Robins-Monroe conditions on the learning rate

What about 2-step TD?

- For 2-step, the expected target is: $r_\pi + \gamma P_\pi r_\pi + \gamma^2 P_\pi^2 V$
- Is this a contraction? If so with what factor?

What about n-step TD?

- Expected update: $r_\pi + \gamma P_\pi r_\pi + \ldots + \gamma^{n-1} P_\pi^{n-1} r_\pi + \gamma^n P_\pi^n V$

- So as we increase n, the influence of V (aka bias) decreases!
- And variance from the reward terms potentially increases
- In the limit of $n \to \infty$, we get MC! No bias from using V to bootstrap, but potentially high variance
- All of the n-step algorithms converge because of the same contraction argument

Recall: Eligibility traces (forward view)

- The λ-return can be rewritten as:

$$G_t^\lambda = \underbrace{(1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_t^{(n)}}_{\text{Until termination}} + \underbrace{\lambda^{T-t-1}G_t}_{\text{After termination}}$$

- If $\lambda = 1$, you get the MC target:

$$G_t^\lambda = (1-1)\sum_{n=1}^{T-t-1}1^{n-1}G_t^{(n)} + 1^{T-t-1}G_t = G_t$$

- If $\lambda = 0$, you get the TD(0) target:

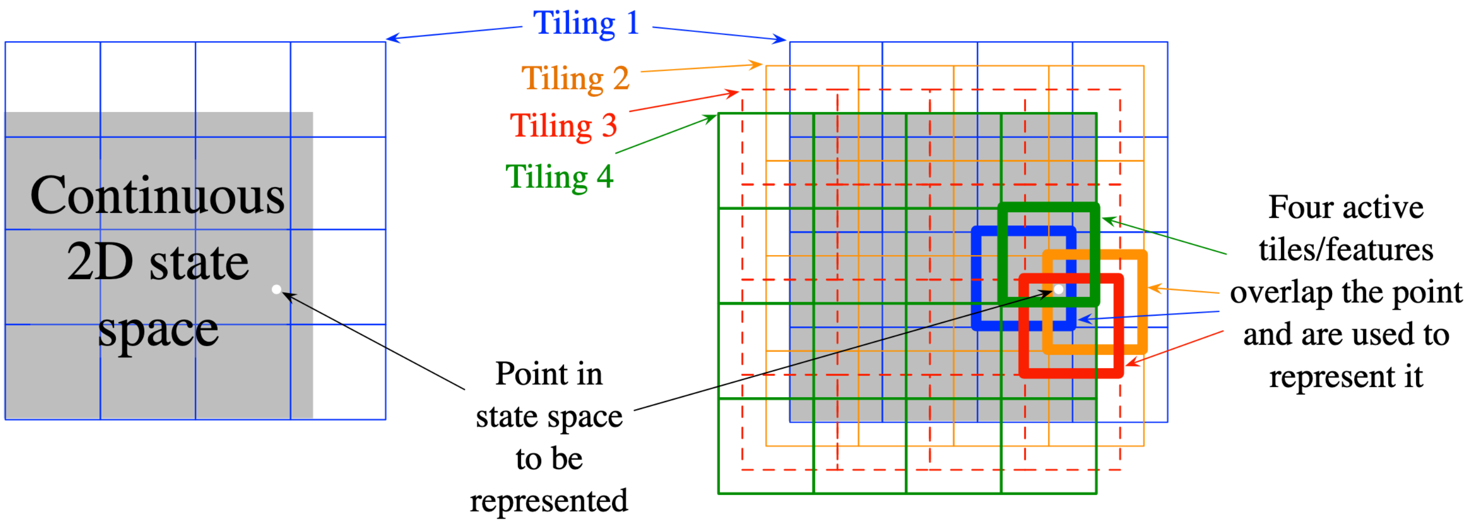
$$G_t^\lambda = (1-0)\sum_{n=1}^{T-t-1}0^{n-1}G_t^{(n)} + 0^{T-t-1}G_t = G_t^{(1)}$$

Convergence of $TD(\lambda)$

- This is a convex combination of n-step targets
- The expected value of each of those is a contraction with at least factor γ
- So the convex combination is also a contraction!
- Therefore in the tabular case we have convergence for all values of λ to v_π
- Footnote on variance and learning rates remains

Linear FA

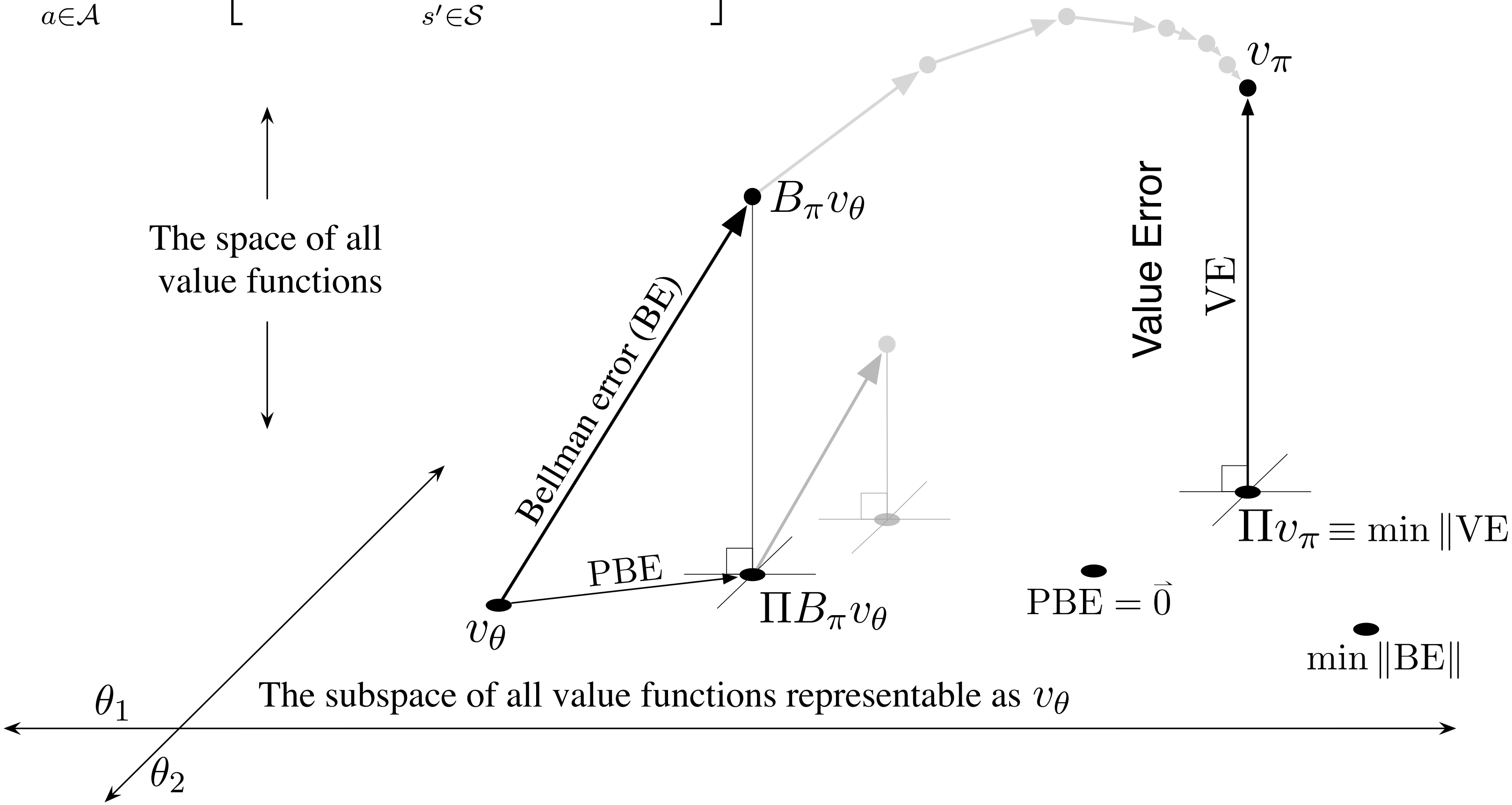
- Features define the plane and the quality of solutions!
- Example: Tile coding, Fourier basis



Geometric intuition

$v_{\boldsymbol{\theta}} \doteq \hat{v}(\cdot, \boldsymbol{\theta})$ as a giant vector $\in \mathbb{R}^{|\mathcal{S}|}$

$$(B_\pi v)(s) \doteq \sum_{a \in \mathcal{A}} \pi(s,a) \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v(s') \right]$$



TD converges to a fixed point a biased but interesting answer

TD(0) update:

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\Big(R_{t+1} + \gamma\boldsymbol{\theta}_t^\top\boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top\boldsymbol{\phi}_t\Big)\boldsymbol{\phi}_t$$
$$= \boldsymbol{\theta}_t + \alpha\Big(R_{t+1}\boldsymbol{\phi}_t - \boldsymbol{\phi}_t\big(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1}\big)^\top\boldsymbol{\theta}_t\Big)$$

Fixed-point analysis:

$$\mathbf{b} - \mathbf{A}\boldsymbol{\theta}_{TD} = \mathbf{0}$$
$$\Rightarrow \qquad \mathbf{b} = \mathbf{A}\boldsymbol{\theta}_{TD}$$
$$\Rightarrow \qquad \boldsymbol{\theta}_{TD} \doteq \mathbf{A}^{-1}\mathbf{b}$$

In expectation:

$$\mathbb{E}[\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t] = \boldsymbol{\theta}_t + \alpha(\mathbf{b} - \mathbf{A}\boldsymbol{\theta}_t),$$

where

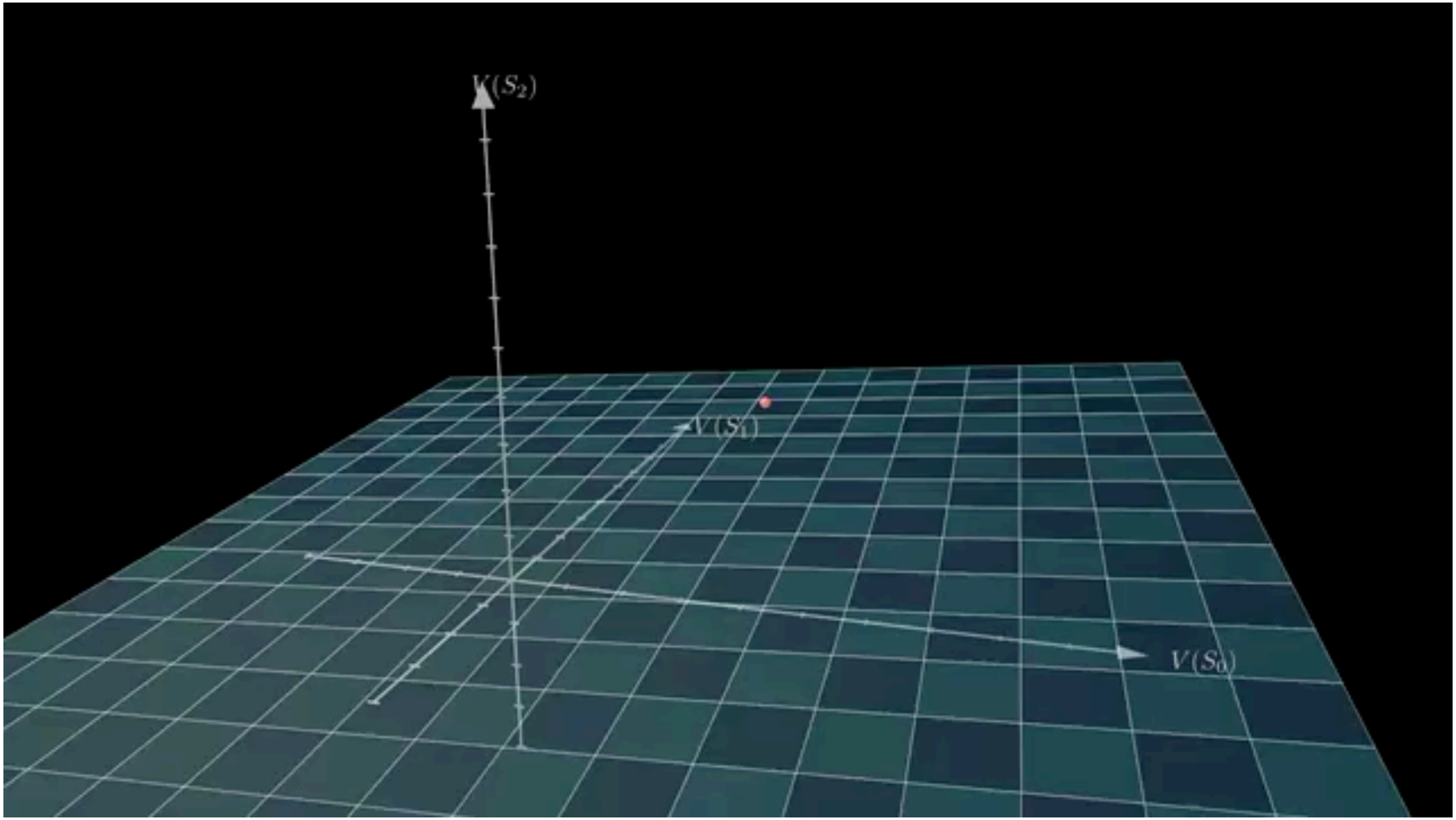
$$\mathbf{b} \doteq \mathbb{E}[R_{t+1}\boldsymbol{\phi}_t] \in \mathbb{R}^n \quad \text{and} \quad \mathbf{A} \doteq \mathbb{E}\Big[\boldsymbol{\phi}_t\big(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1}\big)^\top\Big] \in \mathbb{R}^n \times \mathbb{R}^n$$

Guarantee:

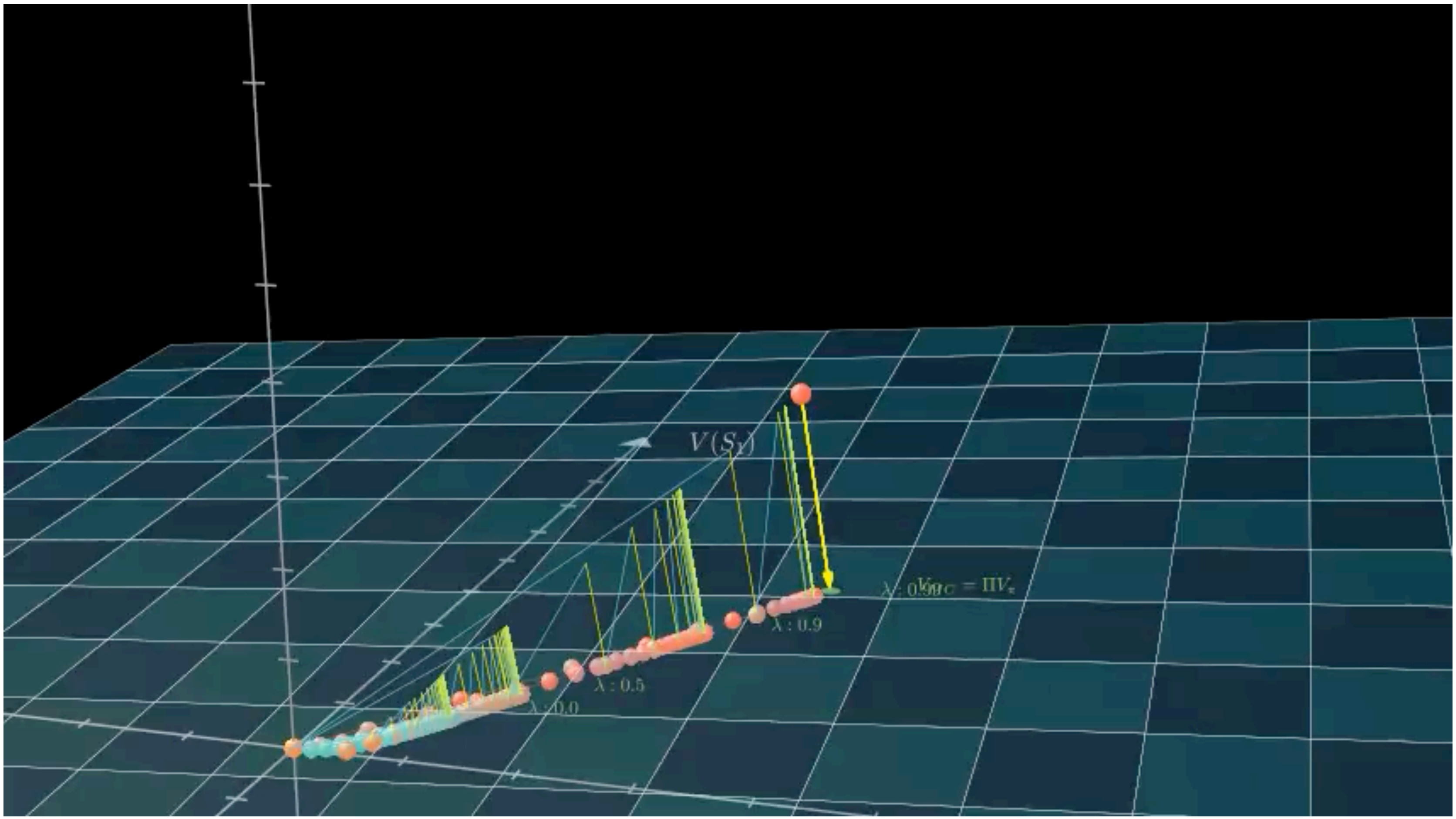
$$\mathrm{MSVE}(\boldsymbol{\theta}_{TD}) \leq \frac{1}{1-\gamma}\min_{\boldsymbol{\theta}} \mathrm{MSVE}(\boldsymbol{\theta})$$

$$\overline{\mathrm{VE}}(\mathbf{w}_\infty) \leq \frac{1-\gamma\lambda}{1-\gamma}\min_{\mathbf{w}} \overline{\mathrm{VE}}(\mathbf{w}).$$

Example: TD updates



$TD(\lambda)$ updates



Comments

- For n-step TD and $TD(\lambda)$, the parameters control how far the updates end up from the optimal projection

- Convergence happens if the Markov chain (resulting from the MDP plus the policy) is ergodic (ie we can get from any state to any other state with non-zero probability, not necessarily right away)

- MC always converges to the best L2 approximation of v_π on the plane defined by the features

What about non-linear function approximation?

- It's a mess!
- We don't have a nice plane on which to project, but rather some curved manifold
- Bootstrapping can be problematic in theory (more on this later)

What about action-value functions?

- They have a very strong relationship to v_π

- $v_\pi = \pi q_\pi$
- (Or $v_\pi(s) = \sum_a \pi(a \mid s) q_\pi(s, a)$)

- $q_\pi = r + \gamma P v_\pi = r + \gamma P \pi q_\pi$

- All contraction arguments still apply