Agenda

- TD (bootstrapping) with approximation
- Off-policy with approximation

Value function approximation II

Konpat Preechakul
Chulalongkorn University
September 2019

Recap

Approximate value, approximate policy

We can show that:

$$v_\pi(s) \geq v^*(s) - \frac{2\epsilon}{1-\gamma}$$

- $v^*(s)$ is the optimal policy performance
- $v_\pi(s)$ is our policy (using $q_\theta(s,a)$)
- ϵ the maximum error between $q_\theta(s,a)$ and $q^*(s,a)$
- Our policy has a lower bound depending on the error!

Intuitive interpretation

$$v_\pi(s) \geq v^*(s) - \frac{2\epsilon}{1-\gamma}$$

- Per-step error:

① $v^*(s) - q^*(s, \pi(s)) \leq 2\epsilon$

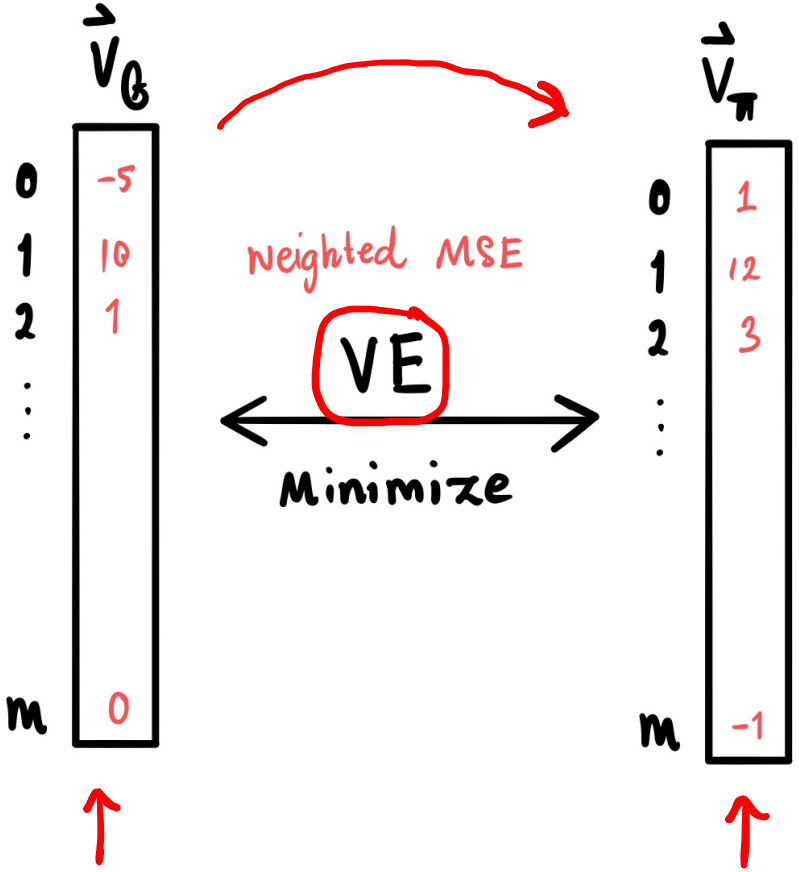
$q^*(s, \pi^*(s))$

- Trajectory error:

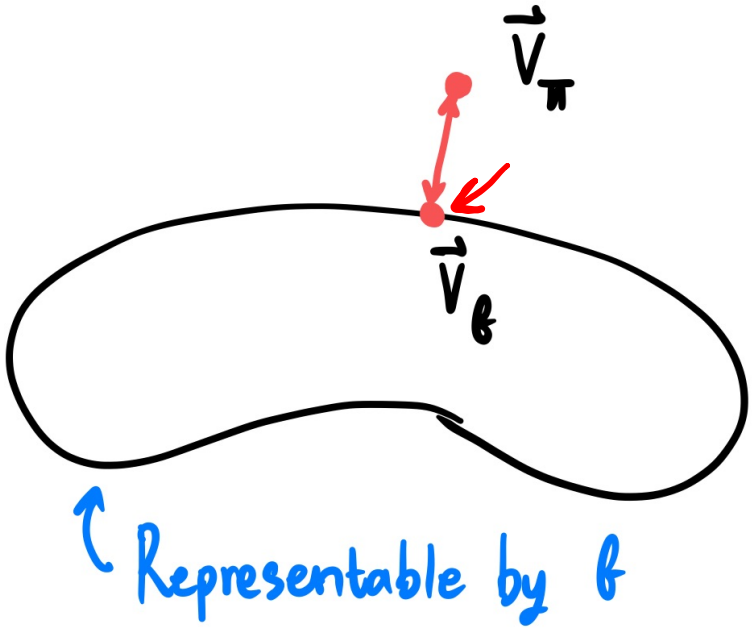
② $v_\pi(s) \geq v^*(s) - \frac{2\epsilon}{1-\gamma}$

$\gamma \uparrow$ $\gamma \downarrow$

Views of approximation



$$\underset{\theta}{\operatorname{argmin}} \| \vec{v}_\pi - \vec{v}_\theta \|^2_{p^\pi}$$



Value Error + SGD

on-policy

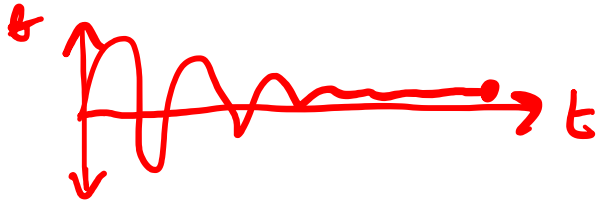
$$\mathcal{L}(\theta) = \mathrm{VE}(\theta) = \sum_s \mathrm{P}^{\pi}(s) \left[\frac{1}{2} (G(s) - v_\theta(s))^2 \right]$$

$$\nabla_\theta \mathcal{L}(\theta) = -(G - v_\theta(s)) \nabla_\theta v_\theta(s)$$

$$S_0 \sim \mathrm{P}_{s_0}, S \sim \pi | S_0, G \sim \pi | S$$

$$\theta \leftarrow \theta + \alpha \left(G - v_\theta(S) \right) \nabla_\theta v_\theta(S)$$

Convergence and fixed point



Convergence

- A training process converges to a fixed point where there is no further progress

Fixed point

- The solution when the training process converges

TD with approximation

Prediction and control

$V_\pi(s)$? π

optimal π

$\pi \xrightarrow{\text{impv}} \pi' \longrightarrow \pi''$

Prediction

- Get V (estimate of the policy)
- A single policy in concern

↑ VE → $V_\theta(s)$

Control

- Prediction + improvement
- A series of policies

Linear vs non-linear

- Linear

$$v_\theta(s) = \theta^T \phi(s)$$

feature vector

$$\nabla_\theta v_\theta(s) = \phi(s)$$

- Non-linear

$$v_\theta(s) = f_\theta(\phi(s))$$

TD Prediction with approximation

Semi-gradient one-step TD

$$\mathrm{VE}(\theta) = \mathbb{E}_{S \sim \mathrm{P}^{\pi}(s)} \left[\frac{1}{2} \left(v_{\pi}(S) - v_{\theta}(S) \right)^2 \right]$$

$$\theta \leftarrow \theta + \alpha \left(v_{\pi}(s_t) - v_{\theta}(s_t) \right) \nabla_{\theta} v_{\theta}(s_t)$$

We don't have a return, we use "bootstrapping" target instead

$$\theta \leftarrow \theta + \alpha \left(r_{t+1} + \gamma v_{\theta}(s_{t+1}) - v_{\theta}(s_t) \right) \nabla_{\theta} v_{\theta}(s_t)$$

Why we call “semi-gradient”?

$$\theta \leftarrow \theta + \alpha \left(R + \gamma v_\theta(S') - v_\theta(S)\right) \nabla_\theta v_\theta(S)$$

- The gradient is “incomplete”
- We assume that v(s’) is “independent” from theta
- This is a false assumption
- **Semi-gradient doesn’t share a usual SGD convergence guarantee**

Semi-gradient TD fixed point

$$\theta \leftarrow \theta + \alpha \left(R + \gamma v_\theta(S') - v_\theta(S) \right) \nabla_\theta v_\theta(S)$$

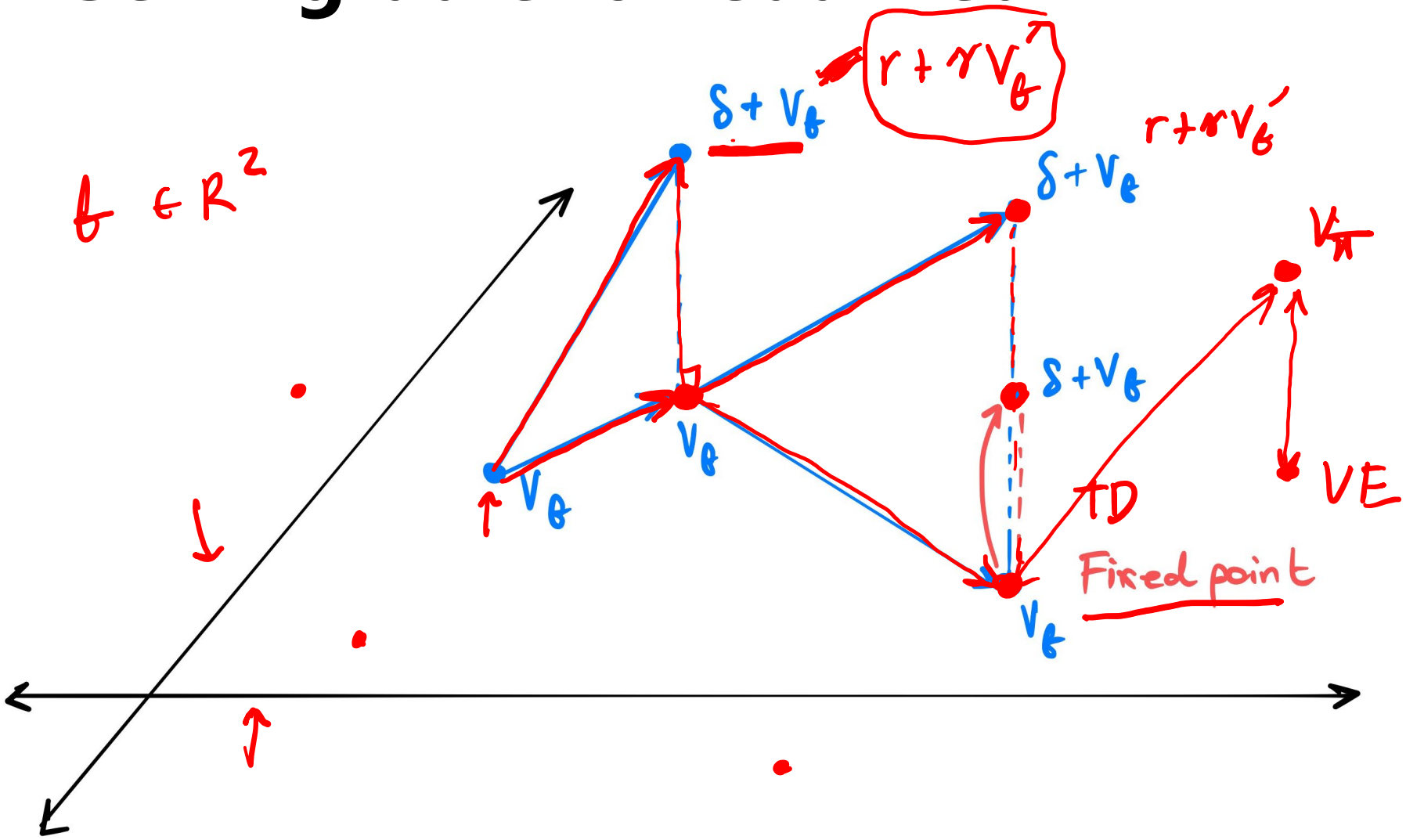
- No further update

$$\theta = \theta + \alpha \mathbb{E} \left[\delta \nabla_\theta v_\theta(S) \right]$$

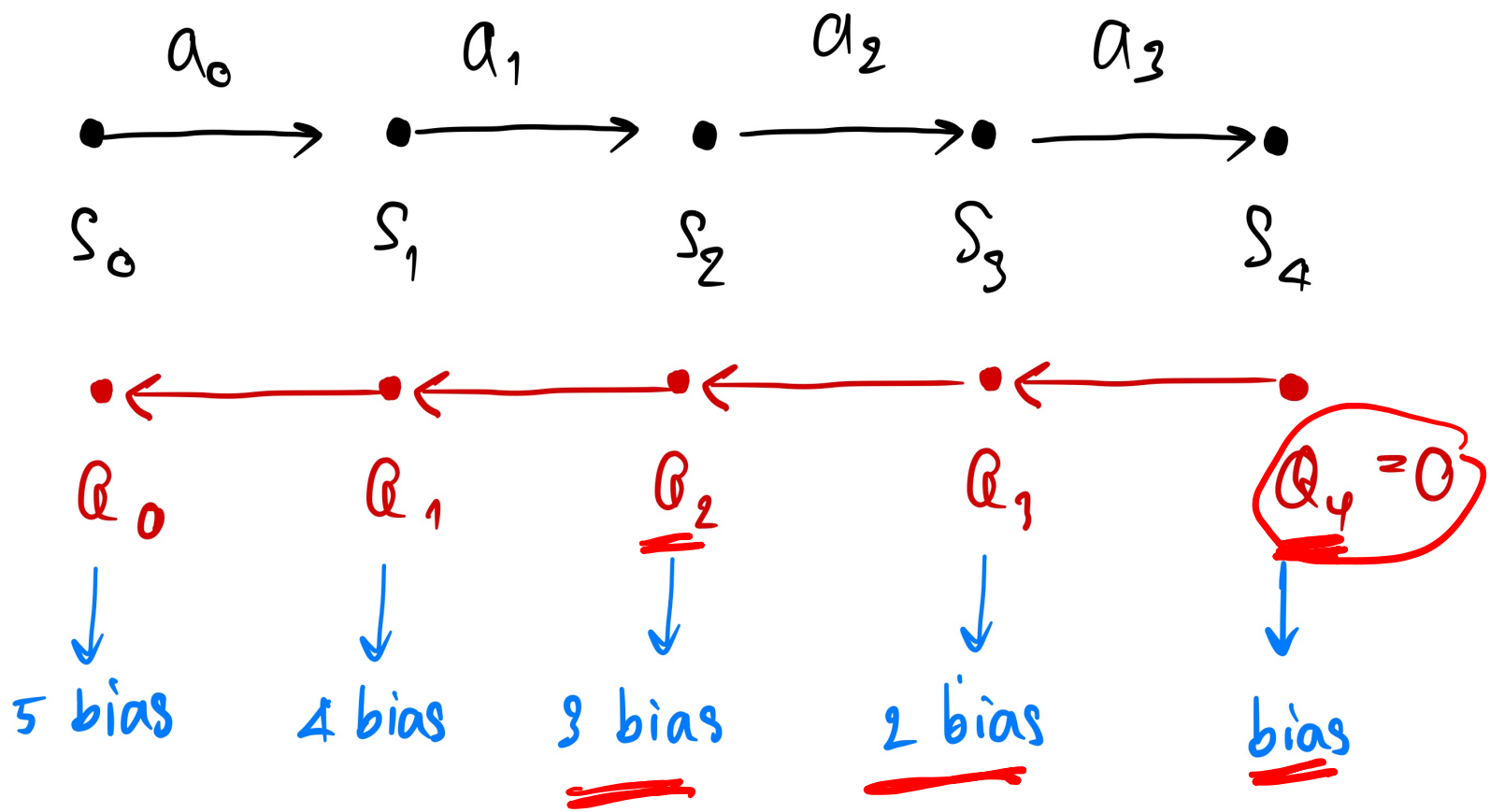
$$0 = \mathbb{E} \left[\delta \nabla_\theta v_\theta(S) \right]$$

- Not much could be said...

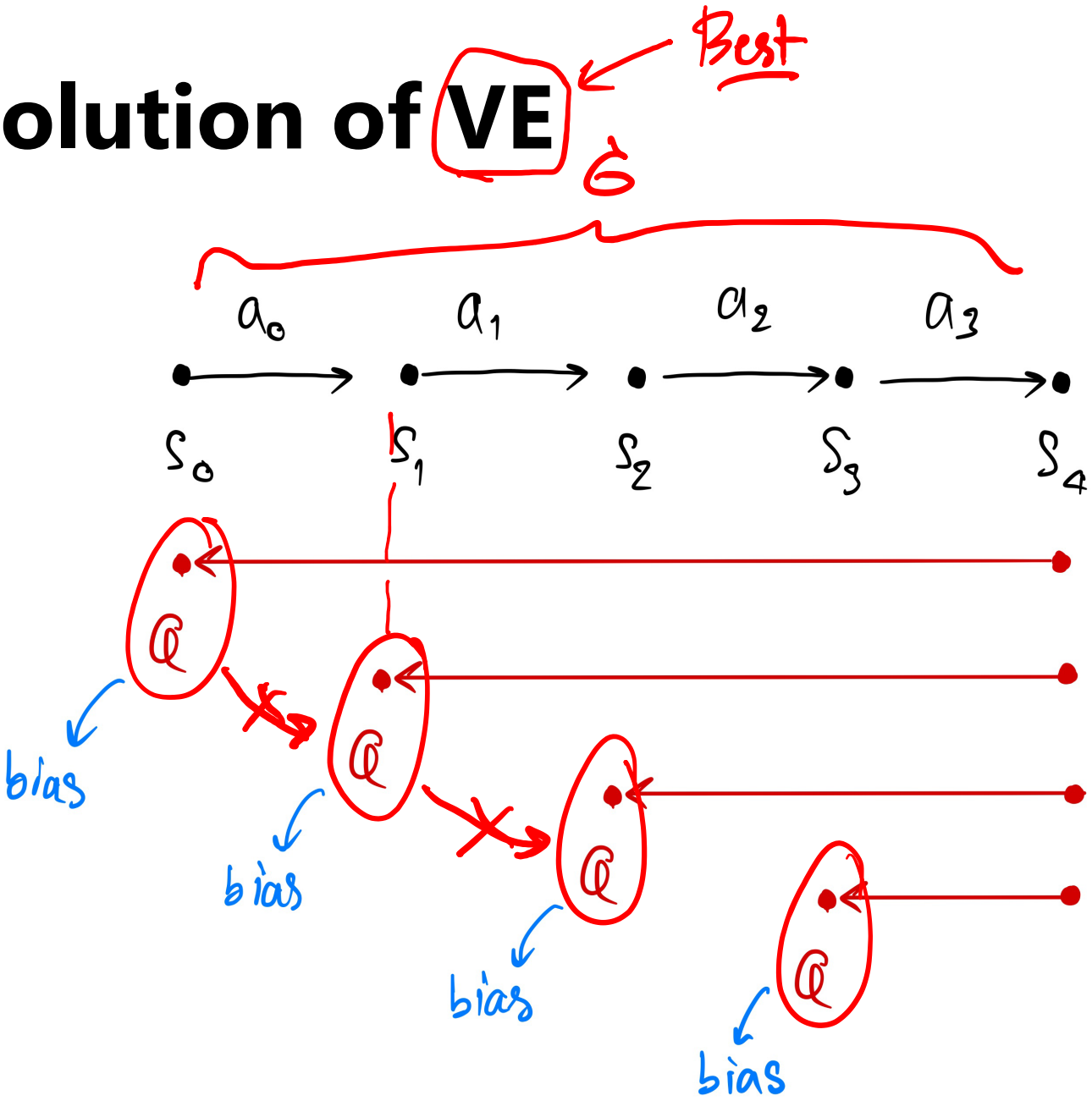
Semi-gradient visualized



Solution of semi-gradient TD

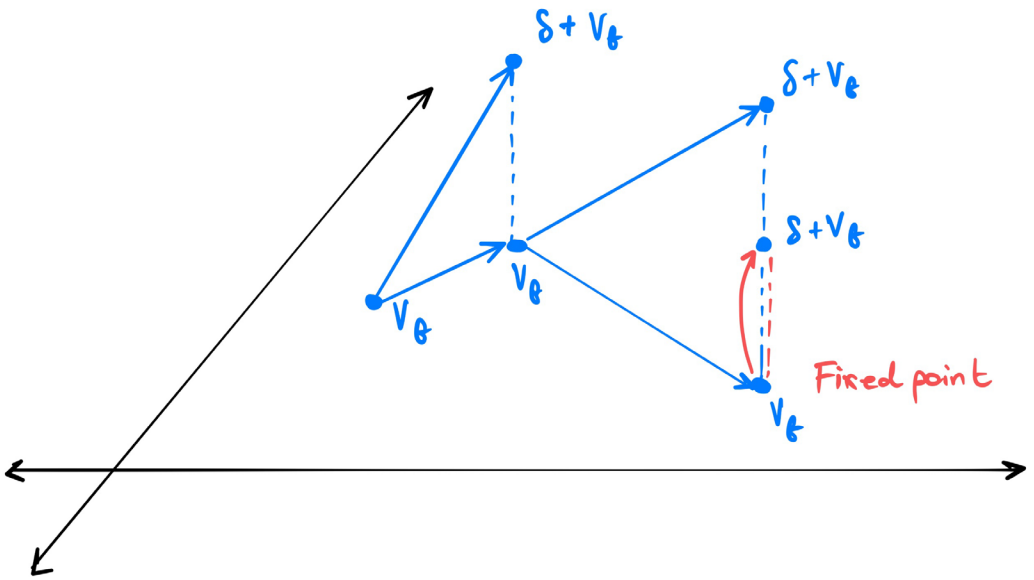


Solution of VE



Solution of semi-gradient TD

- Poorer solution that VE
- Propagation of errors
  - Each step incurs some error, many steps large error
  - Due to projection onto representable space



N-step semi-gradient

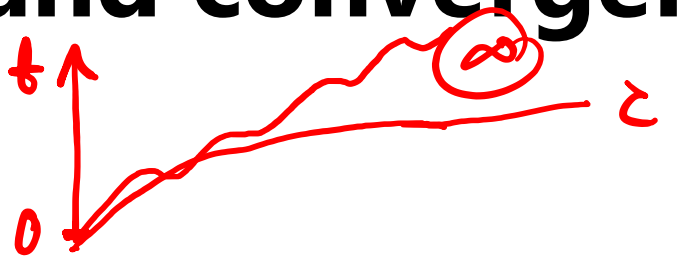
1) $\theta \leftarrow \theta + \alpha \left(r_{t+1} + \gamma v_\theta(s_{t+1}) - v_\theta(s_t)\right) \nabla_\theta v_\theta(s_t)$

2) $\theta \leftarrow \theta + \alpha \left(g_{t:t+n} - v_\theta(s_t)\right) \nabla_\theta v_\theta(s_t)$

Tree backup, λ

- Replace the target with an n-step return
- You could use any kind of target here
- All returns are semi-gradients (except full-return)

Stability and convergence



Stability

- Weights don't explode to **infinity**
- Proof is easier

Convergence

- Weights converge to a fixed point where objective function is minimized

- Convergence != good fixed point

VE > gen gradient TD

Stability of semi-gradient linear TD(0)

- We describe the update in terms of "matrix" multiplication

$$\theta_{t+1} = M\theta_t + c$$

- We show that the matrix is a **"contraction"** mapping
  - Output is smaller than the input vector

$$\|M\theta\| < \|\theta\|$$

Stability of semi-gradient linear TD(0)

$$\theta \leftarrow \theta + \alpha \left(r_{t+1} + \gamma v_\theta(s_{t+1}) - v_\theta(s_t) \right) \nabla_\theta v_\theta(s_t)$$

$$\theta_{t+1} = \theta_t + \alpha \left(r_{t+1} + \gamma \theta_t^T x_{t+1} - \theta_t^T x_t \right) x_t$$

Goal:

$$\theta_{t+1} = M\theta_t + c$$

$$\|M\theta\| < \|\theta\|$$

Stability of semi-gradient linear TD(0)

$$\theta_{t+1} = M\theta_t + c$$

(1) $$\theta_{t+1} = \theta_t + \alpha\left(r_{t+1} + \gamma\theta_t^T x_{t+1} - \theta_t^T x_t\right) x_t$$

$$\alpha\left(r_{t+1}x_t + \gamma x_t\theta_t^T x_{t+1} - x_t\theta_t^T x_t\right)$$

$$\alpha\left(\quad + x_t\theta_t^T(\gamma x_{t+1} - x_t)\right)$$

$$\alpha\left(r_{t+1}x_t - (x_t\theta_t^T(x_t - \gamma x_{t+1}))\right)$$

$$\theta_{t+1} = \theta_t + \left(\alpha(r_{t+1}x_t - x_t(x_t - \gamma x_{t+1})^T\theta_t)\right)$$

Stability of semi-gradient linear TD(0)

$$\theta_{t+1} = M\theta_t + c$$

$$\theta_{t+1} = \theta_t + \alpha\left(r_{t+1}x_t - x_t(x_t - \gamma x_{t+1})^T\theta_t\right)$$

$$= \theta_t + \alpha r_{t+1} x_t - \alpha x_t (x_t - \gamma x_{t+1})^T \theta_t$$

$$= (I - \alpha A)\theta_t + \alpha b$$

Stability of semi-gradient linear TD(0) — on-policy

$$\mathbb{E}\theta_{t+1} = (I - \alpha A)\theta_t + \alpha b \qquad b = \mathbb{E}_{\pi}[r_{t+1}x_t]$$

$$A = \mathbb{E}_{\pi}\left[x_t(x_t - \gamma x_{t+1})^T\right]$$

- $I - \alpha A$ is a contraction
- By showing that $I - \alpha A$ has eigenvalues between 0 and 1
- By showing that A is positive-definite matrix
- **Under on-policy assumption**

Semi-gradient linear TD fixed-point

$$\mathbb{E}\theta_{t+1} = \theta_t + \alpha b - \alpha A \theta_t \qquad \mathbb{E} = x_t, x_{t+1} \sim \mathbb{P}^\pi, R \sim \pi$$

Signs of on-policy

$$b = \mathbb{E}\left[r_{t+1} x_t\right]$$

$$A = \mathbb{E}\left[x_t (x_t - \gamma x_{t+1})^T\right]$$

No further update

$$\theta = \theta + \alpha b - \alpha A \theta$$

$$0 = \alpha b - \alpha A \theta$$

$$0 = b - A\theta$$

$$b = A\theta$$

$$A^{-1} b = \theta$$

LSTD

Semi-gradient TD property

$$\mathbb{E}\theta_{t+1} = \theta_t + \alpha\mathbb{E}\left[\delta_t \nabla_\theta v_\theta(s_t)\right]$$

- It converges to **TD fixed point** in linear case with on-policy

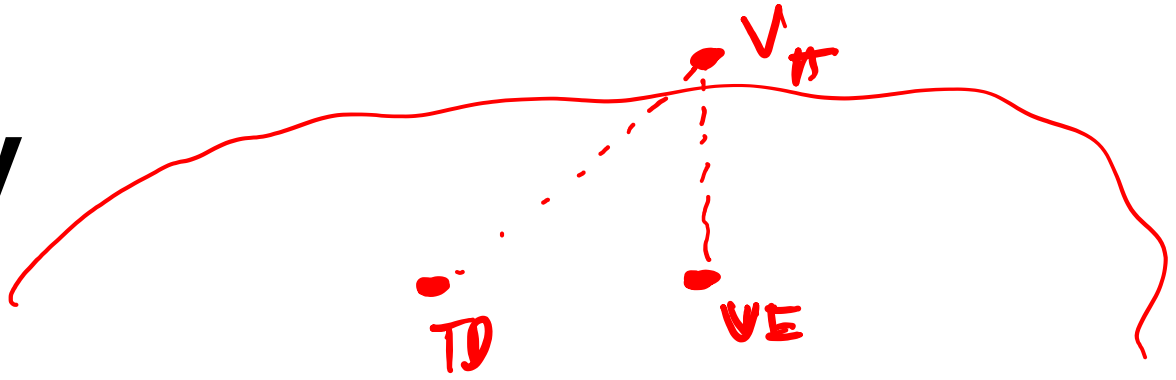
$$\theta = A^{-1}b$$

$$b = \mathbb{E}\left[r_{t+1}x_t\right]$$

$$A = \mathbb{E}\left[x_t(x_t - \gamma x_{t+1})^T\right]$$

- **TD fixed point is considered to be a "good" fixed point**
- It **doesn't** converge in non-linear case even with on-policy

Summary



- VE update

$$\theta \leftarrow \theta + \alpha \left(v_\pi(s_t) - v_\theta(s_t)\right) \nabla_\theta v_\theta(s_t)$$

- Semi-gradient TD update

semi-gradient

$$\theta \leftarrow \theta + \alpha \left(r_{t+1} + \gamma v_\theta(s_{t+1}) - v_\theta(s_t)\right) \nabla_\theta v_\theta(s_t)$$

- Converges to TD fixed point with on-policy (only linear case)

TD control with approximation

SARSA with approximation

1) $\theta \leftarrow \theta + \alpha \left(r_{t+1} + \gamma v_\theta(s_{t+1}) - v_\theta(s_t)\right) \nabla_\theta v_\theta(s_t)$

for until q_θ is stable **do**

1) take action according to $q_\theta(s, a)$ epsilon greedy
2) collect (s, a, r, s', a')
3) $\delta = r + \gamma q_\theta(s', a') - q_\theta(s, a)$

   $\theta \leftarrow \theta + \alpha \delta \nabla_\theta q_\theta(s, a)$

end for

- We need to approximate $q_\theta(s, a)$

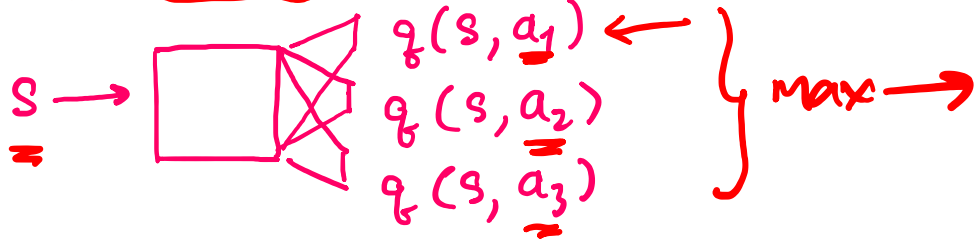
Implementation considerations

Implementation considerations

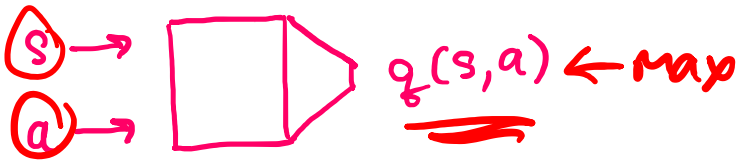
1. Architecture design
2. Co-adaptation nature of on-policy learning
3. Sample correlation
4. Forgetting problems

Architecture design

1. Layer type
   - Conv layers for image inputs
2. Prediction head for Q function
   - Discrete actions = multi-head

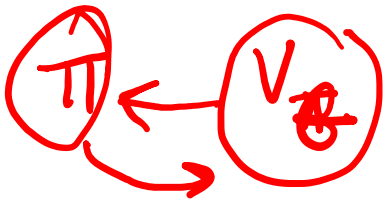


   - Continuous action = single-head

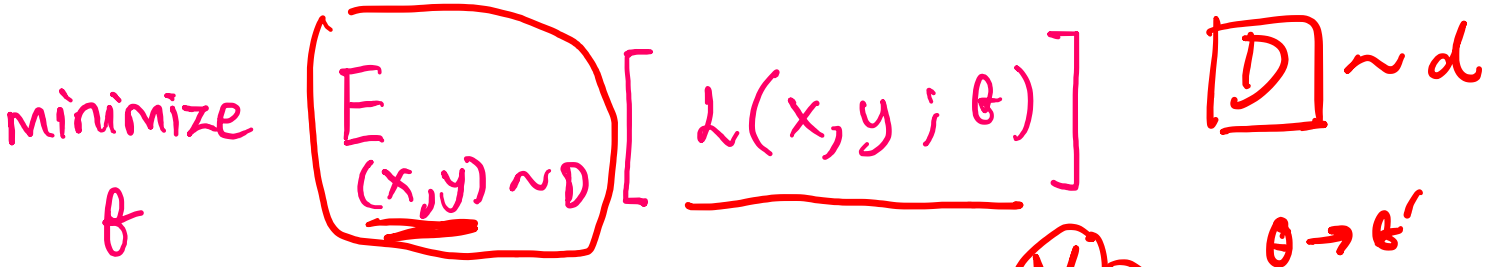


q(s,a) →

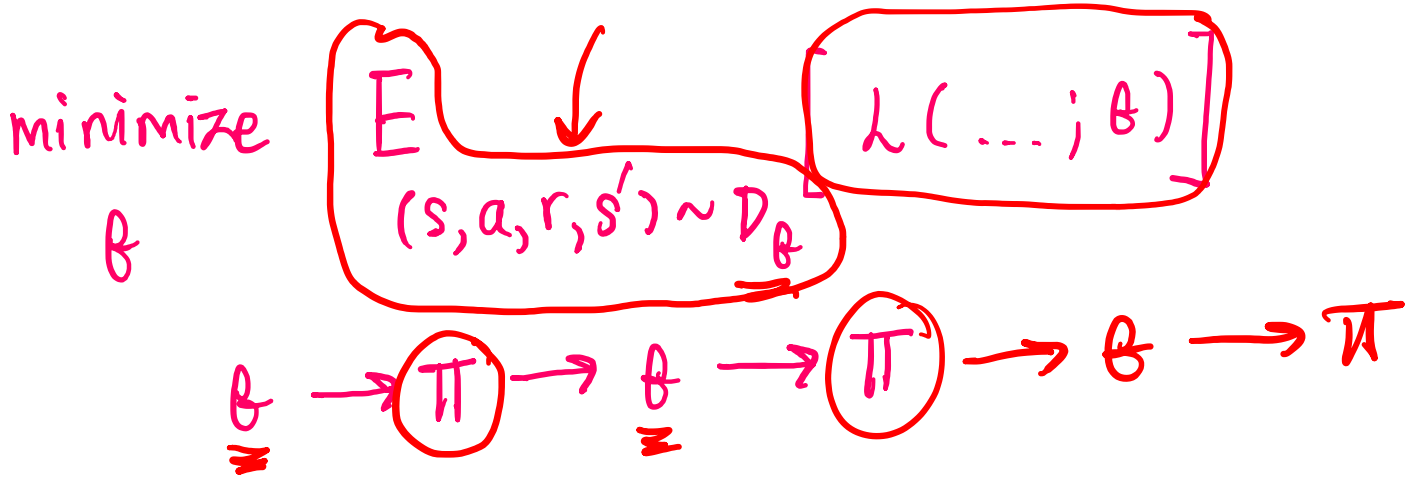
Co-adaptation nature



Supervised learning formulation

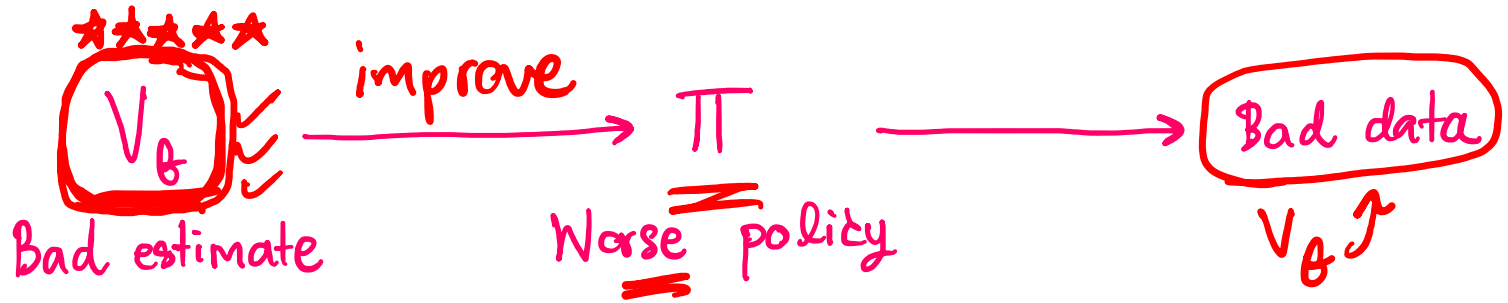


Reinforcement learning formulation



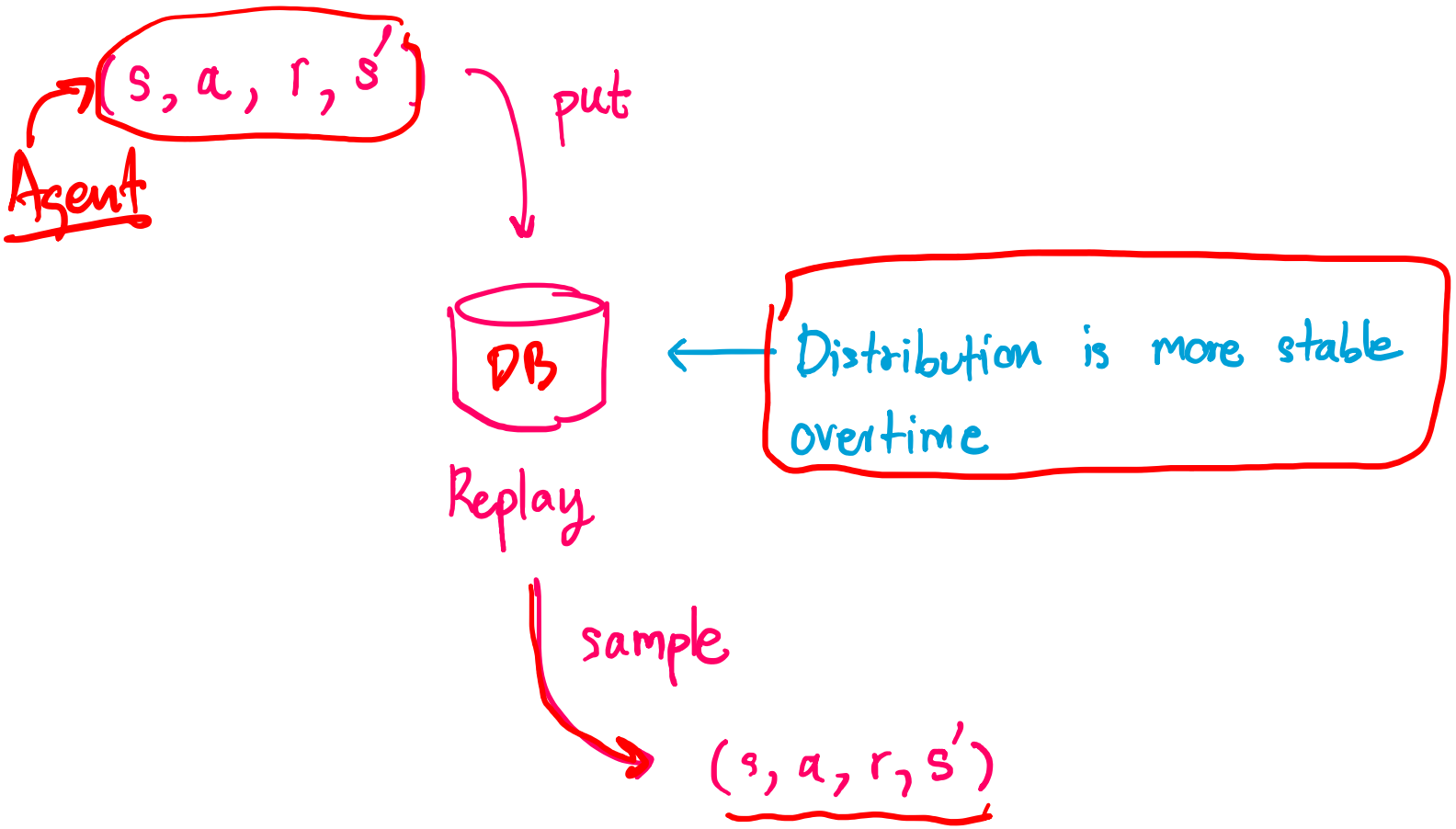
Co-adaptation nature

- The data distribution is constantly changing
  - Because the on-policy is constantly changing
- This could lead to unstable learning loop



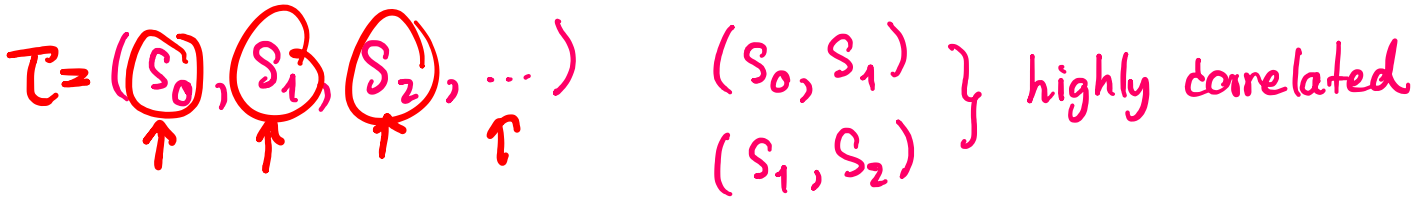
- Off-policy with more stable data distribution helps

Off policy with replay

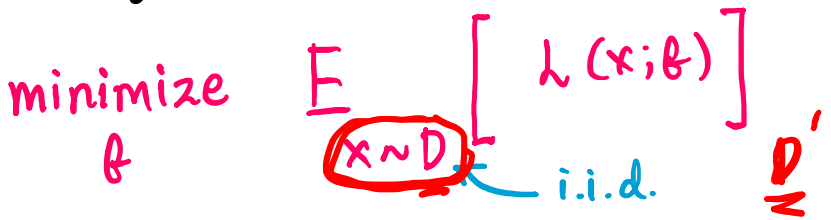


Sample correlation

- On-policy sample is highly correlated



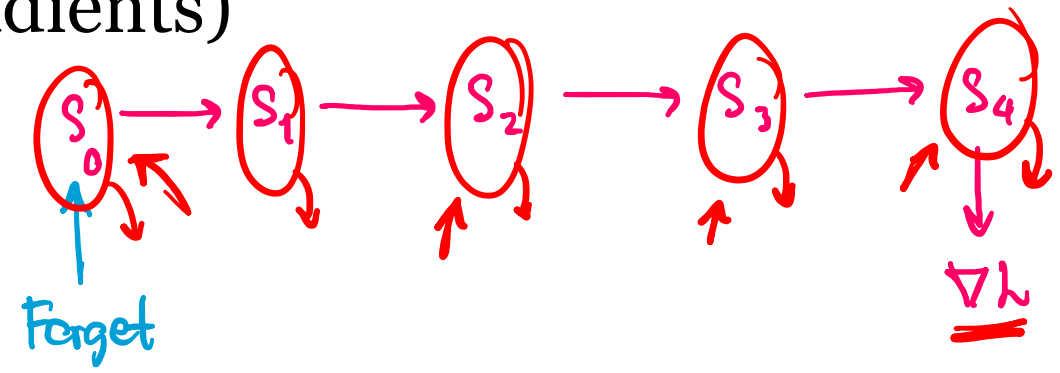
- SGD with independent assumption doesn't work very well



  - It might converge to sub-optimal minima
  - Very low learning rate is needed otherwise

Forgetting

- If the gradient is not representative (correlated gradients)



- To reduce we might need very small learning rate