Planning and learning with continuous state and actions paces

Corrado Possieri

Machine and Reinforcement Learning in Control Applications

Introduction

- Up to now, we assumed to have a finite (small) actions space
- What if we have infinitely many (uncountable) actions?
- We need to resort to adaptive and optimal control theory
  - linear systems;
  - nonlinear systems.

The linear quadratic regulator problem

- Consider the linear dynamical system

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t,$$

with $\mathbf{x}_t \in \mathbb{R}^n$ and $\mathbf{u}_t \in \mathbb{R}^m$.

- The associated performance index is

$$G_t = \sum_{i=t}^{\infty} \left(\mathbf{x}_i^\top \mathbf{Q} \mathbf{x}_i + \mathbf{u}_i^\top \mathbf{R} \mathbf{u}_i \right).$$

- The objective is to **minimize** G_t
  - everything still holds with $\max \leftarrow \min$.

The Bellman equation for the LQR problem

- Let $\mathbf{u}_t = \pi(\mathbf{x}_t)$, where $\pi(\cdot)$ is a given policy
  - $\pi : \mathbb{R}^n \to \mathbb{R}^m$.

- The *state value function* is in this case

$$v_\pi(\mathbf{x}_t) = \sum_{i=t}^{\infty} \left(\mathbf{x}_i^\top \mathbf{Q} \mathbf{x}_i + \pi^\top(\mathbf{x}_i) \mathbf{R} \pi(\mathbf{x}_i) \right).$$

- We have a Bellman equation for $v_\pi(\mathbf{x}_t)$

$$\begin{aligned} v_\pi(\mathbf{x}_t) &= \left(\mathbf{x}_t^\top \mathbf{Q} \mathbf{x}_t + \pi^\top(\mathbf{x}_t) \mathbf{R} \pi(\mathbf{x}_t) \right) + \sum_{i=t+1}^{\infty} \left(\mathbf{x}_i^\top \mathbf{Q} \mathbf{x}_i + \pi^\top(\mathbf{x}_i) \mathbf{R} \pi(\mathbf{x}_i) \right) \\ &= \left(\mathbf{x}_t^\top \mathbf{Q} \mathbf{x}_t + \pi^\top(\mathbf{x}_t) \mathbf{R} \pi(\mathbf{x}_t) \right) + v_\pi(\mathbf{x}_{t+1}). \end{aligned}$$

The Bellman equation is a Lyapunov equation

- Letting $\pi(\mathbf{x}) = \mathbf{K}\mathbf{x}$, assume that

$$v_\pi(\mathbf{x}_t) = \mathbf{x}_t^\top \mathbf{P}_\pi \mathbf{x}_t.$$

- We thus have that

$$v_\pi(\mathbf{x}_t) = \mathbf{x}_t^\top \mathbf{P}_\pi \mathbf{x}_t = \mathbf{x}_t^\top \mathbf{Q} \mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t + \mathbf{x}_{t+1}^\top \mathbf{P}_\pi \mathbf{x}_{t+1}.$$

- Further, since this holds for all state trajectories

$$\mathbf{P}_\pi = \mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K} + (\mathbf{A} + \mathbf{B}\mathbf{K})^\top \mathbf{P}_\pi (\mathbf{A} + \mathbf{B}\mathbf{K}),$$

that is $\mathbf{P}_\pi$ solves a Lyapunov equation.

Optimal policy in the LQR

- The TD error (Hamiltonian) in the LQR is

$$H(\mathbf{x}_k, \mathbf{u}_k) = -\mathbf{x}_t^\top \mathbf{P}_\pi \mathbf{x}_t + \mathbf{x}_t^\top \mathbf{Q} \mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t + \mathbf{x}_{t+1}^\top \mathbf{P}_\pi \mathbf{x}_{t+1}.$$

- A necessary condition for optimality is $\frac{\partial H(\mathbf{x}_k,\mathbf{u}_k)}{\partial \mathbf{u}_k} = 0$.
- The optimal control is thus given by

$$\mathbf{K}_* = -(\mathbf{B}^\top \mathbf{P}_* \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{P}_* \mathbf{A},$$

  where $\mathbf{P}_*$ solves the algebraic Riccati equation

$$\mathbf{P}_* = \mathbf{A}^\top \mathbf{P}_* \mathbf{A} + \mathbf{Q} - \mathbf{A}^\top \mathbf{P}_* \mathbf{B} (\mathbf{B}^\top \mathbf{P}_* \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{P}_* \mathbf{A}.$$

Policy evaluation and policy improvement

- The Bellman equation can be used to evaluate a policy
  - solve the Bellman equation

$$\mathbf{P}_\pi = \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} + (\mathbf{A} + \mathbf{B}\mathbf{K})^\top \mathbf{P}_\pi (\mathbf{A} + \mathbf{B}\mathbf{K});$$

  - the value of the policy $\pi(\mathbf{x}_t) = \mathbf{K}\mathbf{x}_t$ is

$$v_\pi(\mathbf{x}_t) = \mathbf{x}_t^\top \mathbf{P}_\pi \mathbf{x}_t.$$

- We can improve our policy by letting

$$\begin{aligned}\pi(\mathbf{x}_t) &\leftarrow \arg\min_{\mathbf{u}_t} \left\{ \mathbf{x}_t^\top \mathbf{Q}\mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t + \mathbf{x}_{t+1}^\top \mathbf{P}_\pi \mathbf{x}_{t+1} \right\} \\ &= -(\mathbf{B}^\top \mathbf{P}_\pi \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{P}_\pi \mathbf{A}\mathbf{x}_{t+1}.\end{aligned}$$

Policy iteration for the LQR problem

Policy iteration for the LQR problem

Input: matrices $\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}$
Output: $\mathbf{P}_*, \mathbf{K}_*$

Initialization

$\mathbf{K} \leftarrow$stabilizing gain

Loop

solve $\mathbf{P} = \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} + (\mathbf{A} + \mathbf{B}\mathbf{K})^\top \mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{K})$ in $\mathbf{P}$
$\mathbf{K} \leftarrow -(\mathbf{B}^\top \mathbf{P}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}$.

- Note that the iteration

$$\mathbf{P} \leftarrow \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} + (\mathbf{A} + \mathbf{B}\mathbf{K})^\top \mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{K})$$

converges to the solution of the Lyapunov equation.

Update of the value function

- We can directly update the value function.
- We still use the Bellman equation

$$\mathbf{P} \leftarrow \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} + (\mathbf{A} + \mathbf{B}\mathbf{K})^\top \mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{K}),$$

  with

$$\mathbf{K} \leftarrow -(\mathbf{B}^\top \mathbf{P}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}.$$

- It is a single value update followed by a policy update.
- It does not require a stabilizing gain $\mathbf{K}$.

Value iteration for the LQR problem

Value iteration for the LQR problem

Input: matrices $\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}$
Output: $\mathbf{P}_*, \mathbf{K}_*$

Initialization

$\mathbf{P} \leftarrow$arbitrarily

Loop

$\mathbf{K} \leftarrow -(\mathbf{B}^\top \mathbf{P} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{P} \mathbf{A}$
$\mathbf{P} \leftarrow \mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K} + (\mathbf{A} + \mathbf{B}\mathbf{K})^\top \mathbf{P} (\mathbf{A} + \mathbf{B}\mathbf{K})$

- We can obtain GPI by performing multiple updates of $\mathbf{P}$ with a single update of $\mathbf{K}$.

The difference Riccati equation

- Consider the difference Riccati equation

$$\mathbf{P}_{t+1} = \mathbf{Q} + \mathbf{A}^\top \mathbf{P}_t \mathbf{A} - \mathbf{A}^\top \mathbf{P}_t \mathbf{B}(\mathbf{B}^\top \mathbf{P}_t \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{P}_t \mathbf{A}.$$

- The DRE arises from the LQR restricted to

$$G_{t:t+T} = \sum_{i=t}^{t+T} \left(\mathbf{x}_i^\top \mathbf{Q} \mathbf{x}_i + \mathbf{u}_i^\top \mathbf{R} \mathbf{u}_i \right).$$

- The updates are exactly those of VI.

State-action value function for the LQR problem

- The function $q_\pi(\mathbf{x}_t, \mathbf{u}_t)$ is defined as always
  - value gathered using $\mathbf{u}_t$ when at $\mathbf{x}_t$ and following π thereafter;
  - it is simply given by

$$q_\pi(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^\top \mathbf{Q} \mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t + \mathbf{x}_{t+1}^\top \mathbf{P}_\pi \mathbf{x}_{t+1}.$$

- We can define this function in matrix form

$$\begin{aligned} q_\pi(\mathbf{x}_t, \mathbf{u}_t) &= \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{bmatrix}^\top \begin{bmatrix} \mathbf{A}^\top \mathbf{P}_\pi \mathbf{A} + \mathbf{Q} & \mathbf{A}^\top \mathbf{P}_\pi \mathbf{B} \\ \mathbf{B}^\top \mathbf{P}_\pi \mathbf{A} & \mathbf{B}^\top \mathbf{P} \mathbf{B} + \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{bmatrix}^\top \begin{bmatrix} \mathbf{S}_{xx} & \mathbf{S}_{xu} \\ \mathbf{S}_{xu}^\top & \mathbf{S}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{bmatrix}. \end{aligned}$$

- The policy improvement is given by

$$\mathbf{K} \leftarrow -\mathbf{S}_{uu}^{-1} \mathbf{S}_{xu}^\top.$$

Learning the state-action value function

- q_π is homogeneous and quadratic
  - letting $\mathbf{z} = [\ \mathbf{x}^\top \quad \mathbf{u}^\top\ ]^\top$, we have $\mathbf{q}_\pi = \mathbf{z}^\top \mathbf{S}_\pi \mathbf{z}$.
- Letting $\mathbf{X}(\mathbf{z})$ be the vector consisting of all quadratic terms in the elements of $\mathbf{z}$, we can write

$$q_\pi(\mathbf{z}) = \mathbf{w}^\top \mathbf{X}(\mathbf{z}).$$

- The Bellman equation can be rewritten as

$$\mathbf{w}^\top (\mathbf{X}(\mathbf{z}_t) - \mathbf{X}(\mathbf{z}_{t+1})) = \mathbf{x}_t^\top \mathbf{Q}\mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t.$$

- $\mathbf{w}$ can be estimated via recursive least squares.

Adaptive policy iteration for the LQR problem

Adaptive policy iteration for the LQR problem

Input: $\epsilon > 0$, horizon T
Output: $\mathbf{K}_*$

Initialization

$\mathbf{K} \leftarrow$stabilizing gain

Loop

initialize $\mathbf{x}_0$
generate an episode following $\pi(\mathbf{x}) = \mathbf{K}\mathbf{x}$: $\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \dots, \mathbf{x}_T$
$\mathbf{M} \leftarrow \epsilon^{-1}\mathbf{I}$
$\mathbf{h} \leftarrow \mathbf{0}$
for $t = 0, \dots, T-1$ **do**

$$\mathbf{M} \leftarrow \mathbf{M} - \frac{\mathbf{M}(\mathbf{X}_t - \mathbf{X}_{t+1})(\mathbf{X}_t - \mathbf{X}_{t+1})^\top \mathbf{M}}{1 + (\mathbf{X}_t - \mathbf{X}_{t+1})^\top \mathbf{M}(\mathbf{X}_t - \mathbf{X}_{t+1})}$$

$$\mathbf{h} \leftarrow \mathbf{h} + (\mathbf{X}_t - \mathbf{X}_{t+1})(\mathbf{x}_t^\top \mathbf{Q}\mathbf{x}_t + \mathbf{u}_t \mathbf{R}\mathbf{u}_t)$$

$\mathbf{w} \leftarrow \mathbf{M}\mathbf{h}$
reshape $\mathbf{w}$ to obtain $\mathbf{S}$
$\mathbf{K} \leftarrow -\mathbf{S}_{uu}^{-1}\mathbf{S}_{xu}^\top$

Backup of the value action function

- The Bellman backup equation can be rewritten as

$$\mathbf{w}^\top \mathbf{X}(\mathbf{z}_t) = \mathbf{x}_t^\top \mathbf{Q}\mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t + \mathbf{x}_{t+1}^\top \mathbf{P}_\pi \mathbf{x}_{t+1}.$$

- The Schur complement of $\mathbf{S}$ is

$$\begin{aligned}&\mathbf{S}_{xx} - \mathbf{S}_{xu}\mathbf{S}_{uu}^{-1}\mathbf{S}_{xu}^\top \\ &\quad = \mathbf{Q} + \mathbf{A}^\top \mathbf{P}_\pi \mathbf{A} - \mathbf{A}^\top \mathbf{P}_\pi \mathbf{B}(\mathbf{B}^\top \mathbf{P}_\pi \mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{P}_\pi \mathbf{A}.\end{aligned}$$

- This is exactly one step of the DRE.

- We can use it to perform an update of the value function.

Adaptive value iteration for the LQR problem

Adaptive value iteration for the LQR problem

Input: $\epsilon > 0$, horizon T
Output: $\mathbf{P}_*$

Initialization

$\mathbf{P}$ ←arbitrarily

Loop

initialize $\mathbf{x}_0$
generate an episode: $\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_T$
$\mathbf{M} \leftarrow \epsilon^{-1}\mathbf{I}$
$\mathbf{h} \leftarrow \mathbf{0}$
for $t = 0, \ldots, T-1$ **do**
$\quad \mathbf{M} \leftarrow \mathbf{M} - \frac{\mathbf{M}\mathbf{X}_t\mathbf{X}_t^\top\mathbf{M}}{1+\mathbf{X}_t^\top\mathbf{M}\mathbf{X}_t}$
$\quad \mathbf{h} \leftarrow \mathbf{h} + \mathbf{X}_t(\mathbf{x}_t^\top\mathbf{Q}\mathbf{x}_t + \mathbf{u}_t\mathbf{R}\mathbf{u}_t + \mathbf{x}_{t+1}^\top\mathbf{P}\mathbf{x}_{t+1})$
$\mathbf{w} \leftarrow \mathbf{M}\mathbf{h}$
reshape $\mathbf{w}$ to obtain $\mathbf{S}$
$\mathbf{P} \leftarrow \mathbf{S}_{xx} - \mathbf{S}_{xu}\mathbf{S}_{uu}^{-1}\mathbf{S}_{xu}^\top$

Nonlinear systems

- Similar results hold for nonlinear systems

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\mathbf{u}_t.$$

- The objective is to minimize the cost

$$G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r(\mathbf{x}_i, \mathbf{u}_i),$$

where

$$r(\mathbf{x}, \mathbf{u}) = \ell(\mathbf{x}) + \mathbf{u}^\top \mathbf{R}\mathbf{u}.$$

- In this case, a policy is

$$\pi : \mathbb{R}^n \to \mathbb{R}^m.$$

Bellman equation in the nonlinear case

- Given a policy π, the Bellman equation reads as

$$v_\pi(\mathbf{x}_t) = r(\mathbf{x}_t, \pi(\mathbf{x}_t)) + \gamma v_\pi(\mathbf{x}_{t+1}).$$

- The optimal value function thus satisfies

$$v_*(\mathbf{x}_t) = \min_\pi \left\{ r(\mathbf{x}_t, \pi(\mathbf{x}_t)) + \gamma v_*(\mathbf{x}_{t+1}) \right\}.$$

- The above relation is the *Hamilton-Jacobi-Bellman* equation.

- The optimal policy is given by

$$\pi_*(\mathbf{x}_t) = \arg\min_\pi \left\{ r(\mathbf{x}_t, \pi(\mathbf{x}_t)) + \gamma v_*(\mathbf{x}_{t+1}) \right\}.$$

Policy iteration in the nonlinear case

- We can adapt policy iteration to the nonlinear case.
- Letting π be a stabilizing policy, the value estimation step consists in determining v_π such that

$$v_\pi(\mathbf{x}_t) = r(\mathbf{x}_t, \pi(\mathbf{x}_t)) + \gamma v_\pi(\mathbf{x}_{t+1}).$$

- The policy update step consists in updating π as

$$\begin{aligned}\pi(\mathbf{x}_t) &\leftarrow \arg\min_\pi \{r(\mathbf{x}_t, \pi(\mathbf{x}_t)) + \gamma v_\pi(\mathbf{x}_{t+1})\} \\ &= -\frac{\gamma}{2}\mathbf{R}^{-1}\mathbf{g}^\top(\mathbf{x}_t)\nabla v_\pi(\mathbf{x}_{t+1}).\end{aligned}$$

Value iteration in the nonlinear case

- We can adapt aslo value iteration to the nonlinear case.
- Given the policy π, update the value function as

$$v_\pi(\mathbf{x}_t) \leftarrow r(\mathbf{x}_t, \pi(\mathbf{x}_t)) + \gamma v_\pi(\mathbf{x}_{t+1}).$$

- Update the policy as

$$\pi(\mathbf{x}_t) \leftarrow -\frac{\gamma}{2}\mathbf{R}^{-1}\mathbf{g}^\top(\mathbf{x}_t)\nabla v_\pi(\mathbf{x}_{t+1}).$$

- In this case, the initial policy need not be stabilizing.

Value function approximation

- Assuming that v_π is sufficiently smooth, the Weierstrass Theorem guarantees that there is a basis $\mathbf{X}(\mathbf{x})$ such that

$$v_\pi(\mathbf{x}) \simeq \mathbf{w}^\top \mathbf{X}(\mathbf{x}).$$

- In policy iteration, we can estimate $\mathbf{w}$ on the basis of

$$\mathbf{w}_{k+1}^\top \left(\mathbf{X}(\mathbf{x}_t) - \gamma \mathbf{X}(\mathbf{x}_{t+1})\right) = r(\mathbf{x}_t, \pi(\mathbf{x}_t)).$$

- In value iteration, we update $\mathbf{w}$ on the basis of

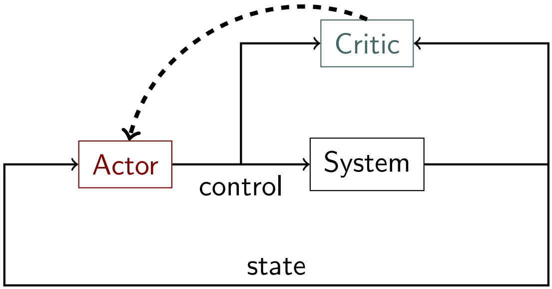
$$\mathbf{w}_{k+1}^\top \mathbf{X}(\mathbf{x}_t) = r(\mathbf{x}_t, \pi(\mathbf{x}_t)) + \gamma \mathbf{w}_k^\top \mathbf{X}(\mathbf{x}_{t+1}).$$

- In both cases the policy update is

$$\pi(\mathbf{x}_t) \leftarrow -\frac{\gamma}{2}\mathbf{R}^{-1}\mathbf{g}^\top(\mathbf{x}_t)\nabla \mathbf{X}^\top(\mathbf{x}_{t+1})\mathbf{w}_{k+1}.$$

- The obtained policy cannot be implemented since it is acausal.

Actor critic structure



Actor: implements the control policy.

Critic: evaluates the current policy.

Actor dynamics

- So far, we designed the *critic* dynamics.
- We can introduce an *actor* to implement the control policy

$$\mathbf{u}_t = \mathbf{v}^\top \mathbf{Y}(\mathbf{x}_t).$$

- The actor weights can be tuned using gradient descent

$$\mathbf{v} \leftarrow \mathbf{v} + \beta \left(\mathbf{v}^\top \mathbf{Y}(\mathbf{x}_t) + \frac{\gamma}{2} \mathbf{R}^{-1} \mathbf{g}^\top(\mathbf{x}_t) \nabla \mathbf{X}^\top(\mathbf{x}_{t+1}) \mathbf{w}_{k+1} \right) \mathbf{Y}(\mathbf{x}_t).$$

- We still need to know $\mathbf{g}(\mathbf{x})$.

Action value function in the nonlinear case

- To overcome the requirement about $\mathbf{g}$, consider

$$\begin{aligned} q_\pi(\mathbf{x}_t, \mathbf{u}_t) &= r(\mathbf{x}_t, \mathbf{u}_t) + \gamma v_\pi(\mathbf{x}_{t+1}) \\ &= r(\mathbf{x}_t, \mathbf{u}_t) + \gamma q_\pi(\mathbf{x}_{t+1}, \pi(\mathbf{x}_{t+1})). \end{aligned}$$

- Letting $\mathbf{z} = [\ \mathbf{x}^\top \quad \mathbf{u}^\top\ ]^\top$, we have $q_\pi(\mathbf{z}_t) = \mathbf{w}^\top \mathbf{Z}(\mathbf{z}_t)$.
- Policy iteration
  - evaluation: determine $\mathbf{w}_{k+1}$ such that

$$\mathbf{w}_{k+1}^\top (\mathbf{Z}(\mathbf{z}_t) - \gamma \mathbf{Z}(\mathbf{z}_{t+1})) = r(\mathbf{x}_t, \pi(\mathbf{x}_t));$$

  - policy improvement: the improved policy is

$$\pi(\mathbf{x}_t) = \arg\min_{\mathbf{u}} \left\{ \mathbf{w}_{k+1}^\top \mathbf{Z}(\mathbf{x}_t, \mathbf{u}) \right\}.$$

- Value iteration

$$\mathbf{w}_{k+1}^\top \mathbf{Z}(\mathbf{z}_t) = r(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_k^\top \mathbf{Z}(\mathbf{z}_{t+1}).$$

Q-learning based on policy iteration

Q-learning based on policy iteration

Input: $\epsilon > 0$, horizon T, basis $\mathbf{Z}$
Output: π_*

Initialization

$\pi \leftarrow$stabilizing policy

Loop

initialize $\mathbf{x}_0$
generate an episode following π: $\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_T, \mathbf{u}_T$
$\mathbf{M} \leftarrow \epsilon^{-1}\mathbf{I}$
$\mathbf{h} \leftarrow \mathbf{0}$
for $t = 0, \ldots, T-1$ **do**

$\quad \mathbf{M} \leftarrow \mathbf{M} - \frac{\mathbf{M}(\mathbf{Z}_t - \gamma \mathbf{Z}_{t+1})(\mathbf{Z}_t - \gamma \mathbf{Z}_{t+1})^\top \mathbf{M}}{1 + (\mathbf{Z}_t - \gamma \mathbf{Z}_{t+1})^\top \mathbf{M}(\mathbf{Z}_t - \gamma \mathbf{Z}_{t+1})}$
$\quad \mathbf{h} \leftarrow \mathbf{h} + (\mathbf{Z}_t - \gamma \mathbf{Z}_{t+1}) r(\mathbf{x}_t, \mathbf{u}_t)$

$\mathbf{w} \leftarrow \mathbf{M}\mathbf{h}$
$\pi(\mathbf{x}_t) \leftarrow \arg\min_{\mathbf{u}}\{\mathbf{w}^\top \mathbf{Z}(\mathbf{x}_t, \mathbf{u})\}$

Q-learning based on value iteration

Q-learning based on value iteration

Input: $\epsilon > 0$, horizon T, basis $\mathbf{Z}$
Output: v_*

Initialization

$\mathbf{w} \leftarrow$arbitrarily

Loop

initialize $\mathbf{x}_0$
generate an episode following π: $\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_T, \mathbf{u}_T$
$\mathbf{M} \leftarrow \epsilon^{-1}\mathbf{I}$
$\mathbf{h} \leftarrow \mathbf{0}$
for $t = 0, \ldots, T-1$ **do**
$\quad \mathbf{M} \leftarrow \mathbf{M} - \frac{\mathbf{M}\mathbf{Z}_t\mathbf{Z}_t^\top\mathbf{M}}{1+\mathbf{Z}_t^\top\mathbf{M}\mathbf{Z}_t}$
$\quad \mathbf{h} \leftarrow \mathbf{h} + \mathbf{Z}_t(r(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}^\top\mathbf{Z}_{t+1}^\top)$
$\mathbf{w} \leftarrow \mathbf{M}\mathbf{h}$
$\pi(\mathbf{x}_t) \leftarrow \arg\min_{\mathbf{u}}\{\mathbf{w}^\top\mathbf{Z}(\mathbf{x}_t, \mathbf{u})\}$

Approximation by artificial neural networks

- We can use neural networks to approximate q.
- SGD can be used to train the critic.
- We can use a critic to approximate the optimal policy.
- SGD can be used to train the actor.

SARSA for nonlinear systems

SARSA for nonlinear systems

Input: critic $\hat{q}$, actor $\hat{\pi}$, $\alpha > 0$, $\beta > 0$
Output: q_*, π_*

Initialization

$\mathbf{w} \leftarrow$arbitrarily
$\mathbf{v} \leftarrow$arbitrarily
$\mathbf{x} \leftarrow$initial state
$\mathbf{u} \leftarrow \hat{\pi}(\mathbf{x}, \mathbf{v})$

Loop

pick control $\mathbf{u}$
observe next state $\mathbf{x}'$ and reward r
$\mathbf{u}' \leftarrow \hat{\pi}(\mathbf{x}', \mathbf{v})$
$\mathbf{w} \leftarrow \mathbf{w} + \alpha(r + \gamma\hat{q}(\mathbf{x}', \mathbf{u}', \mathbf{w}) - \hat{q}(\mathbf{x}, \mathbf{u}, \mathbf{w}))\nabla\hat{q}(\mathbf{x}, \mathbf{u}, \mathbf{w})$
$\mathbf{v} \leftarrow \mathbf{v} + \beta(\arg\min_{\mathbf{u}} \hat{q}(\mathbf{x}, \mathbf{u}, \mathbf{w}) - \hat{\pi}(\mathbf{x}, \mathbf{v}))\nabla\hat{\pi}(\mathbf{x}, \mathbf{v})$
$\mathbf{x} \leftarrow \mathbf{x}'$
$\mathbf{u} \leftarrow \mathbf{u}'$