Reinforcement Learning in Continuous Time and Space

- From K. Doya, Neural Computation 12,219-245, 2000

Continuous Time Discounted Value Function

- Continuous time dynamical system: $\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$
- Reward: $r(t) = r\left(\mathbf{x}(t), \mathbf{u}(t)\right)$
- Policy: $\mathbf{u}(t) = \boldsymbol{\mu}\left(\mathbf{x}(t)\right)$
- The policy's decaying (i.e. discounted) value function:

$$V^{\mu}\left(\mathbf{x}(t)\right) = \int_t^{\infty} e^{-\frac{s-t}{r}} r\left(\mathbf{x}(s), \mathbf{u}(s)\right) ds$$

- Optimal policy's value function

$$V^{*}\left(\mathbf{x}(t)\right) = \max_{\mathbf{u}[t,\infty)} \left[\int_t^{\infty} e^{-\frac{s-t}{r}} r\left(\mathbf{x}(s), \mathbf{u}(s)\right) ds\right]$$

Continuous Time HJB for Discounted Rewards

- Separate integral into [t,t+Δt] and [t+Δt,∞):

$$V^*\left(\mathbf{x}(t)\right) = \max_{\mathbf{u}[t,t+\Delta t]} \left[\underbrace{\int_t^{t+\Delta t \infty} e^{-\frac{s-t}{r}} r\left(\mathbf{x}(s), \mathbf{u}(s)\right) ds}_{\approx r(\mathbf{x}(t),\mathbf{u}(t))\Delta t} + e^{-\frac{\Delta t}{r}} V^*\left(\mathbf{x}(t+\Delta t)\right) \right]$$

- Approximate $V^*(\mathbf{x}(t+\Delta t))$ by Taylor 1st degree

$$V^*\left(\mathbf{x}(t+\Delta t)\right) \approx V^*\left(\mathbf{x}(t)\right) + \frac{\partial V^*}{\partial \mathbf{x}(t)} \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \Delta t$$

- Plug in and rearrange a bit

$$\left(1 - e^{-\frac{\Delta t}{r}}\right) V^*\left(\mathbf{x}(t)\right) = \max_{\mathbf{u}[t,t+\Delta t]} \left[r\left(\mathbf{x}(t), \mathbf{u}(t)\right) \Delta t + e^{-\frac{\Delta t}{r}} \frac{\partial V^*}{\partial \mathbf{x}(t)} \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \Delta t \right]$$

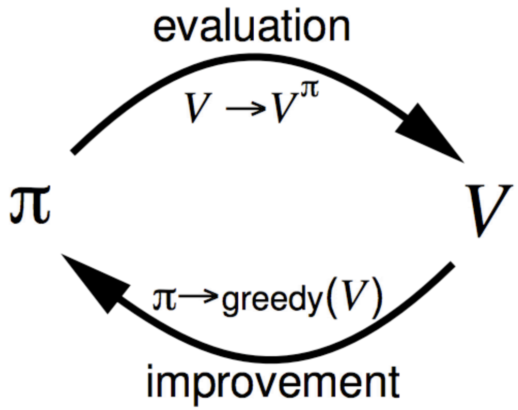
- Take $\Delta t \rightarrow 0$

$$\frac{1}{\tau} V^* \left(\mathbf{x}(t)\right) = \max_{\mathbf{u}(t)} \left[r\left(\mathbf{x}(t), \mathbf{u}(t)\right) + \frac{\partial V^*}{\partial \mathbf{x}(t)} \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \right]$$

✳ compare with original HJB

$$-\frac{\partial J^0}{\partial t} = \min_{\mathbf{u}(t)} \left\{ L(\mathbf{x}, \mathbf{u}(t), t) + \frac{\partial J^0}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}(t), t) \right\}$$

- Solution approach: GPI



- Must use function approximators!

Learning the Value Function

- Use function approximator with parameter vector **w:** $V^{\mu}(\mathbf{x}(t)) \simeq V(\mathbf{x}(t); \mathbf{w})$
- by HJB: $\frac{1}{\tau} V^{\boldsymbol{\mu}}(\mathbf{x}(t)) = r(\mathbf{x}(t), \boldsymbol{\mu}(\mathbf{x}(t))) + \underbrace{\frac{\partial V^{\boldsymbol{\mu}}}{\partial \mathbf{x}(t)} \mathbf{f}(\mathbf{x}(t), \boldsymbol{\mu}(\mathbf{x}(t)))}_{=\dot{V}^{\mu}(t)}$

  i.e. $\dot{V}^{\mu}(\mathbf{x}(t)) = \frac{1}{\tau} V^{\mu}(\mathbf{x}(t)) - r(t)$
- Define the inconsistency (TD error) as $\delta(t) \equiv r(t) - \frac{1}{\tau} V(t) + \dot{V}(t)$
- Reduce inconsistency by correcting weights:

$$\dot{\mathbf{w}} = \eta \delta(t) \frac{\partial V(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}}$$

  where η is a scaling factor
- This is TD(0)

Learning Value Func. by TD(λ)

- Correction decays exponentially. I.e. the desired correction due to the current discrepancy is

$$\hat{V}(t) = \begin{cases} \delta(t_0) e^{-\frac{t_0 - t}{\tau}} & t \leq t_0, \\ 0 & t > t_0, \end{cases}$$

- The weights should therefore be updated by

$$\dot{\mathbf{w}} = \eta \delta(t_0) \underbrace{\int_{-\infty}^{t_0} e^{-\frac{t_0 - t}{\tau}} \frac{\partial V(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} dt}_{\text{eligibility}, \ \mathbf{e}(t)}$$

- The eligibility can be computed as a linear (time varying) dynamical system

$$\begin{aligned} \dot{\mathbf{w}} &= \eta \delta(t) \mathbf{e}(t) \\ \dot{\mathbf{e}}_i(t) &= -\frac{1}{\kappa} \mathbf{e}(t) + \frac{\partial V(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} \end{aligned}$$

where κ is a time decay constant

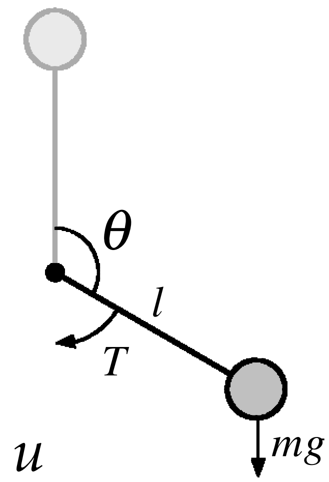
Policy Improvement by Value Gradient

- If we know $r(\mathbf{x}(t),\mathbf{u})$ and $\mathbf{f}(\mathbf{x}(i),\mathbf{u})$ we can select action that maximizes expected reward:

$$\mathbf{u}(t) = \mu\left(\mathbf{x}(t)\right) = \arg\max_{\mathbf{u}\in U}\left[r(\mathbf{x}(t),\mathbf{u}) + \frac{\partial V(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}(t),\mathbf{u})\right]$$

- This is not full DP because it is only done on visited states
- Can be difficult in general. If r is convex in $\mathbf{u}$ and $\mathbf{f}$ is linear in $\mathbf{u}$ the solution is unique and easy to find.
- If $\mathbf{u}$ is bounded (say by ± 1) we can either clip the result or do it more smoothly with a sigmoid, $s(\mathbf{u}) = {}^{2}/_{\pi}\arctan(c\mathbf{u})$ (where c determines sensitivity).
- $\mathbf{f}$ and r can be learned on line (with function approximators) and used here instead of the "true" pair.

Pendulum Swing-Up, Limited Torque



- State = $[\theta, \omega = {}^{d\theta}/_{dt}]$
- Control u= torque = ${}^{d\omega}/_{dt}$
- Model is known: $\dot{\theta} \;=\; \omega$ and $ml^2\dot{\omega} \;=\; -\mu\,\omega + mgl\sin\theta \;+ u$
- Value function approximated by a normalized Gaussian network

$$V(\mathbf{x}, \mathbf{w}) = \frac{\sum_{k=1}^{K} w_k e^{-\frac{\|\mathbf{x}-\mathbf{c}_k\|^2}{\sigma_k^2}}}{\sum_{k=1}^{K} e^{-\frac{\|\mathbf{x}-\mathbf{c}_k\|^2}{\sigma_k^2}}}$$

- Reward = cos(θ)-0.1u-0.1$|\omega|$
- Used eligibility trace (time constant κ = 0.7)
- Model simulated by Runge Kutta 4 with dt = 0.07.
  Learning dynamics (eligibility trace) simulated by Euler method

Pendulum Results

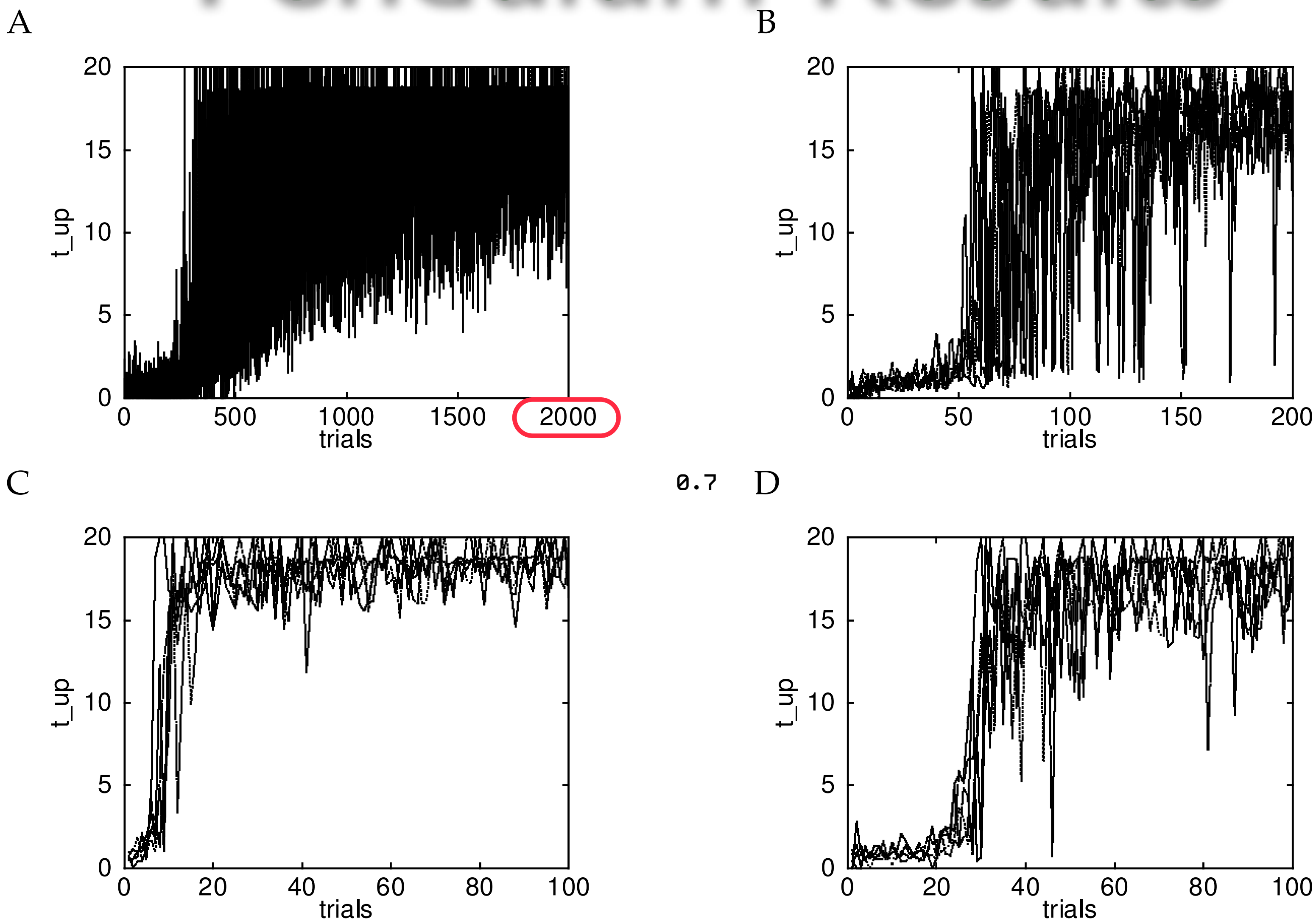


Figure 4: Comparison of the time course of learning with different control schemes: (A) discrete actor-critic, (B) continuous actor-critic, (C) value-gradient-based policy with an exact model, (D) value-gradient policy with a learned model (note the different scales). t_{up}: time in which the pendulum stayed up. In