From Approximate Policy Iteration to Policy Gradients

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Recap

Unknown MDP, learn from roll-outs

Fitted Q Iteration, Q-learning

Problem of Bootstrapping:
Errors in fitting Q feedback leading to more errors.
Further exacerbated by the min()

What about policy iteration?

Policy Iteration

Init with some policy π

Repeat forever

Evaluate policy π

$$Q^\pi(s,a) = c(s,a)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} Q^\pi(s', \pi(s'))] \quad \forall (s,a)$$

Improve policy

$$\pi^+(s) = \arg\min_a Q^\pi(s,a) \qquad \forall s$$

Things to like about policy iteration

Can potentially converge much faster value iteration

Easy to initialize it with a starting policy

(e.g. use BC to initialize)

Why does policy iteration work at all?

If I select a new policy π^+

$$\pi^+(s) = \arg\min_a Q^\pi(s, a) \quad \forall s$$

is the new policy better than π

$$V^{\pi^+}(s_0) - V^\pi(s_0) \leq 0$$

i.e. do I get monotonic improvement?

Performance Difference Lemma

Summary

Performance Difference Lemma (PDL)

$$V^{\pi^+}(s_0) - V^{\pi}(s_0) = \sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^+}} A^{\pi}(s_t, \pi^+)$$

If I select a new policy π^+

$$\pi^+(s) = \arg\min_a Q^{\pi}(s, a) \quad \forall s$$

→

Then advantage must be negative

$$A^{\pi}(s, \pi^+) \leq 0 \quad \forall s$$

↓

Monotonic improvement $\quad V^{\pi^+}(s_0) - V^{\pi}(s_0) \leq 0$

What about *approximate* policy iteration?

Approximate Policy Iteration

Init with some policy π

Repeat forever

Evaluate policy π

Rollout π, collect data (s, a, s', a'), fit a function $Q^{\pi}_{\theta}(s, a)$

Improve policy

$$\pi^{+}(s) = \arg\min_{a} Q^{\pi}_{\theta}(s, a)$$

Does approximate policy iteration give me monotonic improvement?

When poll is active respond at **PollEv.com/sc2582**

Send **sc2582** to **22333**

Performance Difference Lemma (PDL)

$$V^{\pi^+}(s_0) - V^{\pi}(s_0) = \sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^+}} A^{\pi}(s_t, \pi^+)$$

Approximate Policy Iteration

Collect data (s, a, s', a') using policy π

Fit a Q on the data: $Q_\theta^\pi(s,a) = c(s,a)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} Q_\theta^\pi(s', a')]$

Improve: $\pi^+(s) = \arg\min_a Q_\theta^\pi(s,a)$

Problem with Approximate Policy Iteration

$$V^{\pi^+}(s_0) - V^{\pi}(s_0) = \sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^+}} A^{\pi}(s_t, \pi^+)$$

PDL requires accurate Q_θ^π on states that π^+ will visit! $(d_t^{\pi^+})$

But we only have states that π visits (d_t^{π})

If π^+ changes drastically from π, then $|d_t^{\pi^+} - d_t^{\pi}|$ is big!

Be stable

Slowly change policies

Policy Gradients

Policy Gradients

At the end of the day, all we care about is finding a good policy

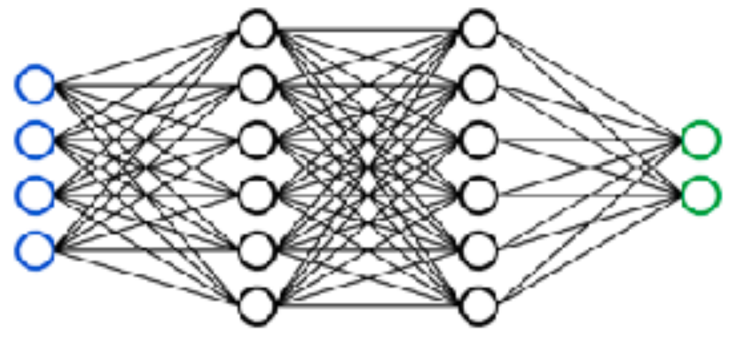
Directly learn parameters of such a policy π_θ

Parameters allow us to slowly update the policy

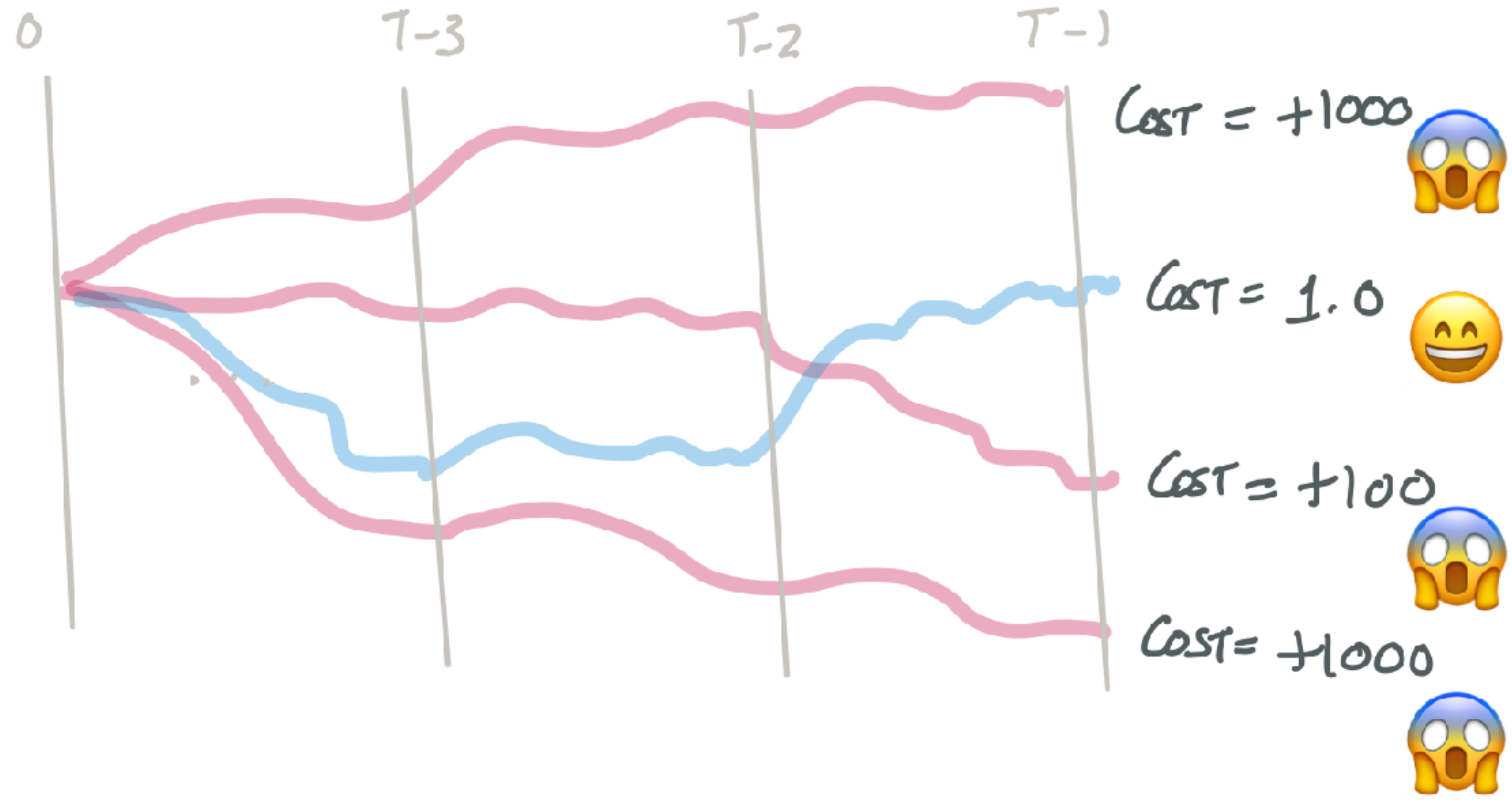
Led to powerful modern RL algorithms like TRPO, PPO, etc

Policy Gradient

$$\pi_\theta : s_t \rightarrow a_t$$



Learn a mapping from states to actions



Roll-out policies in the real-world to estimate value

Let's derive policy gradients