From Approximate Policy Iteration to Policy Gradients

Sanjiban Choudhury
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 $\pi$

Repeat forever

Evaluate policy $\pi$

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

Improve policy

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

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

is the new policy better than $\pi$

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

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 \( \pi^+ \)

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

Then advantage must be negative

\[
A^\pi(s, \pi^+) \leq 0 \quad \forall s
\]

Monotonic improvement

\[
V^{\pi^+}(s_0) - V^\pi(s_0) \leq 0
\]
What about approximate policy iteration?
Approximate Policy Iteration

Init with some policy $\pi$

Repeat forever

Evaluate policy $\pi$

Rollout $\pi$, collect data $(s, a, s', a')$, fit a function $Q_\theta^\pi(s, a)$

Improve policy

$$\pi^+(s) = \operatorname{arg}
\min_a Q_\theta^\pi(s, a) \quad \forall s$$
Does approximate policy iteration give me monotonic improvement?

Performance Difference Lemma (PDL)

\[ V_{\pi^+}^\pi(s_0) - V_{\pi}^\pi(s_0) = \sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_{\pi^+}^t} A_{\pi}(s_t, \pi^+) \]

Collect data \((s, a, s', a')\) using policy \(\pi\)

Fit a \(Q\) on the data:

\[ Q_{\theta}^\pi(s, a) = c(s, a)) + \gamma \mathbb{E}_{s' \sim T(s, a)} Q_{\theta}^\pi(s', a') \]

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 \( Q^\pi_\theta \) on states that \( \pi^+ \) will visit! \((d_t^{\pi^+})\)

But we only have states that \( \pi \) visits \((d_t^\pi)\)

If \( \pi^+ \) changes drastically from \( \pi \), 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 $\pi_\theta$

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