CS 4/5789 7 March 2022 Prof Sarah Dean

Lecture 12: Supervision via Bellman

In this lecture we consider an alternative method for supervising (i.e. finding target labels for) Q functions. First we start with a fundamental Lemma.

1) Performance Difference Lemma

Goal: understand V^π vs. $V^{\pi'}$ in terms of the difference between π vs. π'.

Lemma (Performance Difference):

$$V^\pi(s_0) - V^{\pi'}(s_0) = \frac{1}{1-\gamma} \mathbb{E}_{s\sim d_{s_0}^\pi}\Big[\underbrace{\mathbb{E}_{a\sim\pi(s)}[Q^{\pi'}(s,a)] - V^{\pi'}(s)}_{A^{\pi'}(s,a)}\Big]$$

For $r(s,a)\in[0,1]$,

$$|V^\pi(s_0) - V^{\pi'}(s)| \le \frac{1}{(1-\gamma)^2} \mathbb{E}_{s\sim d_{s_0}^\pi}\Big[\underbrace{\sum_{a\in\mathcal{A}} |\pi(a|s) - \pi'(a|s)|}_{\|\pi(\cdot|s)-\pi'(\cdot|s)\|_1}\Big]$$

The first expression inspires us to define

Def (Advantage) $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$

The "advantage" of taking action a at state s rather than following π.

Notice that $A^\pi(s,\pi(s)) = 0$.

Also notice $\arg\max_a A^\pi(s,a) = \arg\max_a Q^\pi(s,a)$

Proof of PDL:

$$V^{\pi}(s_0)-V^{\pi'}(s_0)= V^{\pi}(s_0) - \mathbb{E}_{a_0\sim\pi(s_0)}\Big[r(s_0,a_0)+\gamma\, \mathbb{E}_{s_1\sim P(s_0,a_0)}[V^{\pi'}(s_1)]\Big] + \mathbb{E}_{a_0\sim\pi(s_0)}\Big[r(s_0,a_0)+\gamma\, \mathbb{E}_{s_1\sim P(s_0,a_0)}[V^{\pi'}(s_1)]\Big] - V^{\pi'}(s_0)$$

$$= \gamma\, \mathbb{E}_{\substack{a_0\sim\pi(s_0)\\ s_1\sim P(s_0,a_0)}}\Big[V^{\pi}(s_1)-V^{\pi'}(s_1)\Big] + \mathbb{E}_{a_0\sim\pi(s_0)}\Big[Q(a_0,s_0)-V^{\pi'}(s_0)\Big]$$

The first statement in the Lemma follows by iteration (similar to Simulation Lemma)

$$\mathbb{E}_{a\sim\pi(s)}\Big[Q^{\pi'}(s,a)-V^{\pi'}(s)\Big] = \mathbb{E}_{a\sim\pi(s)}\Big[Q^{\pi'}(s,a)\Big] - \mathbb{E}_{a\sim\pi'(s)}\Big[Q^{\pi'}(s,a)$$

$$= \sum_{a\in\mathcal{A}} \big(\pi(a|s)-\pi'(a|s)\big)\, Q^{\pi'}(s,a)$$

Therefore,

$$|V^{\pi}(s_0)-V^{\pi'}(s_0)| \leq \frac{1}{1-\gamma}\, \mathbb{E}_{s\sim d^{\pi}_{s_0}}\Big[\sum_{a\in\mathcal{A}} |\pi(a|s)-\pi'(a|s)|\, Q^{\pi'}(s,a)\Big]$$

The second statement follows by noting $0\leq Q^{\pi'}(s,a) \leq \frac{1}{1-\gamma}$ $\square$

We can use the PDL to prove monotonic improvement of policy iteration (HW2).

$$V^{\pi^{t+1}}(s_0) - V^{\pi^{t}}(s_0) = \frac{1}{1-\gamma}\, \mathbb{E}_{s\sim d^{\pi^{t+1}}_{s_0}}\Big[A^{\pi^{t}}(s,\pi^{t+1}(s))\Big]$$

2) Supervision via Bellman Equation

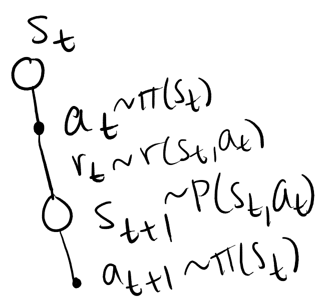
Recall the Bellman Expectation Equation:

$$Q^{\pi}(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(s,a)} \left[V^{\pi}(s') \right]$$

$$= r(s,a) + \gamma \mathop{\mathbb{E}}_{\substack{s' \sim P(s,a) \\ a' \sim \pi(s')}} \left[Q^{\pi}(s',a') \right]$$

possibly stochastic

IDEA: we can bootstrap a label for supervision with just one time step!



At time t, we are at s_t and sample & take action $a_t \sim \pi(s_t)$. As a result we observe r_t and s_{t+1}. Then we sample $a_{t+1} \sim \pi(s_{t+1})$.

Then our target/label is defined as:

$$y_t = r_t + \gamma \hat{Q}(s_{t+1}, a_{t+1}) . \quad (s_t, a_t, y_t)$$

$$y_t \approx Q^{\pi}(s_t, a_t)$$

This is sometimes called "Temporal Difference" target

The TD error is

$$r_t + \gamma \hat{Q}(s_{t+1}, a_{t+1}) - \hat{Q}(s_t, a_t)$$

In the tabular setting, a basic Algorithm:

Alg: SARSA subroutine ("state-action-reward-state-action")

initialize Q^0, $s_0 \sim \mu_0$, $a_0 \sim \pi(s_0)$

for $t = 0, 1, \ldots$

Take action a_t, observe $s_{t+1} \sim P(s_t, a_t)$ & $r_t \sim r(s_t, a_t)$

Sample $a_{t+1} \sim \pi(s_{t+1})$

Update $Q^{t+1}(s_t, a_t) = (1-\alpha) Q^t(s_t, a_t) + \alpha\left(r_t + \gamma Q^t(s_{t+1}, a_{t+1})\right)$

This subroutine can be incorporated into an approximate dynamic programming algorithm (ie as the sample based policy evaluation step)

Policy Improvement w/ ε-greedy

SARSA requires sufficient exploration to converge (for now a formal statement & proof are out of scope)

A common strategy is ε-greedy:

$$\pi(s) = \begin{cases} \arg\max_a Q(s,a) & \text{w.p. } 1-\varepsilon \\ a_0 & \text{w.p. } \varepsilon/A \\ a_1 & \text{w.p. } \varepsilon/A \\ \vdots & \vdots \end{cases}$$

or using slightly different notation:

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{A} & a = \arg\max Q(s,a) \\ \frac{\varepsilon}{A} & \text{o.w.} \end{cases}$$

Comparison with Rollout-based supervision(MC):

1) TD can update Q function online at every step, MC must wait untill end of rollout

2) TD is biased when $\hat{Q} \neq Q^{\pi}$

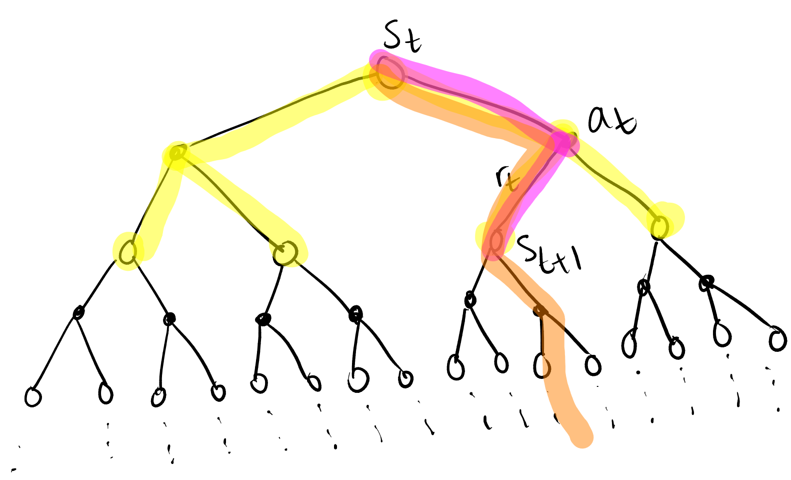
   $r_t + Q^{\pi}(s_{t+1}, a_{t+1})$ is unbiased, but we don't know Q^{π}!

   MC is unbiased

3) Variance of TD estimate due to one stochastic transition: $a_t \sim \pi(s_t)$, $s_{t+1} \sim P(s_t, a_t)$, $a_{t+1} \sim \pi(s_{t+1})$

   Variance of MC due to many transitions Therefore higher.

4) BOTH methods supervise Q^{π} using data collected from rollouts with π, ie. they are both on policy



Dynamic Programing Bellman Expectation: 1 time step, all possible outcomes

Rollout-based (MC): Many timestep, sampled outcome

Bellman-based (TD): One timestep, sampled outcome

3) Supervision with Bellman Optimality

So far, we focus on estimating Q^π. But we ultimately only care about Q^*. Can we focus on this directly?

recall: Bellman optimality:

$$Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\left[\max_{a'} Q^*(s',a')\right]$$

recall: Value Iteration:

An algorithm for finding an optimal policy that focused on Q^* directly

Init. Q^0
for $t=0,1,\ldots$
  $Q^{t+1} = \text{Bellman Op}(Q^t)$

BellmanOperator(Q^t):
$$Q^{t+1}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}\left[\max_{a'} Q^t(s',a')\right]$$

s_t a_t r_t s_{t+1}

Sample-based supervision:

$$y_t = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a) \qquad (s_t, a_t, y_t)$$

$$y_t \approx Q^*(s_t, a_t)$$

Alg: Q-learning in the tabular setting

initialize Q
for $t = 0,1,\ldots$
  e.g. ε greedy
  take action a_t & observe $s_{t+1} \sim P(s_t, a_t)$, $r_t \sim r(s_t, a_t)$
  $Q(s,a) \leftarrow (1-\alpha) Q(s,a) + \alpha\left(r_t + \gamma \max_{a'} Q(s_{t+1}, a')\right)$

Some properties of Bellman optimality based supervision

1) updates at every timestep
2) biased label when $Q \neq Q^*$
3) variance depends on randomness from one timestep
4) Not specific to a policy, so can use off policy data.

4) Function approximation

Bellman-based supervision (like rollout based) gives us labels that we can use to train models: $\{(s_i, a_i, y_i)\}_{i=1}^N$

ERM: $$\min_{Q \in \mathcal{Q}} \sum_{i=1}^N (Q(s_i,a_i) - y_i)^2$$

suppose parametrized model class
$$\mathcal{Q} = \{Q_\theta \mid \theta \in \mathbb{R}^d\}$$

Bellman-based supervision is online & incremental. So rather than full ERM minimization, it is common to do gradient descent updates to θ using incoming data.

$$\nabla_\theta (Q_\theta(s_i,a_i) - y_i)^2 = 2(Q_\theta(s_i,a_i) - y_i) \nabla_\theta Q_\theta(s_i,a_i)$$

update looks like
$$\theta \leftarrow \theta + \alpha(Q_\theta(s_i,a_i) - y_i) \nabla Q_\theta(s_i,a_i))$$

could be Bellman-exp (SARSA) or Bellman-opt (Q-learning)