Can Sparser Rewards Improve Offline Reinforcement Learning?

Franklyn Wang

April 17, 2021

Table of Contents

Reinforcement Learning Motivation

- Many real-world tasks are based on decision making, not just predictions.
- **Examples:**

Reinforcement Learning Motivation

- Many real-world tasks are based on decision making, not just predictions.
- **Examples:**
    - Self-driving Cars (not just sense, but avoid)

Reinforcement Learning Motivation

- Many real-world tasks are based on decision making, not just predictions.
- **Examples:**
  - Self-driving Cars (not just sense, but avoid)
  - Selling Ads (not just predict clicks, but decide on what to serve)
- *Reinforcement Learning* is a natural framework to tackle such problems.

Reinforcement Learning Primer

- Reinforcement Learning centers around a *Markov Decision Process*: an agent has a state s, and by taking actions a it acquires rewards and moves to other states.
- Formally, it has a policy $\pi(a|s)$ which dictates its actions, a transition function $p(s'|a, s)$ which dictates the dynamics, and a reward function $r(s, a)$.
- Generally, reinforcement learning works through environmental interaction, and we steadily improve the policy that way.

Problem

It took AlphaGo 2 million games of Go to be good at Go. Can we drive cars off mountains 2 million times?
–Yann LeCun, 2019

Offline RL

- In offline RL, we collect the data beforehand from real agents (think: human drivers). Then we train the agent, who is deployed into the real world without any further training.

- This is a very difficult task – fortunately, we can often make it easier by incorporating a model of the real world into our predictions. This manifests itself through learning $p(s'|s, a)$.

Table of Contents

The exploitation problem

- The big problem with model-based reinforcement learning is that the model sometimes can't be trusted. Thus, the policy may learn to "fool" the model instead of actually learning the task.
- Thus, a lot of work has centered around finding ways of bounding the *discrepancy* – the difference between the reward on the true MDP versus that of the learned MDP.

Prior Work

So far, the prior work has focused on creating *pessimal MDPs*: that is, MDPs so that the reward of any policy on that MDP is lower than that on the true MDP. With a pessimal MDP, it suffices to simply use planning.

- In [YTY+20], they construct an MDP where the reward function is simply $\hat{r}(s, a) = r(s, a) - \alpha h(s, a)$ where $h(s, a)$ is a measure of the ucnertainty in the model on (s, a).
- In [KRNJ20], they construct an MDP where high uncertainty actions are sent to an absorbing state with infinite negative reward.

Table of Contents

Theorem

The analysis in the two papers is quite similar, so we summarize the analysis in [KRNJ20].

Theorem ([KRNJ20])

Assume that $D_{TV}(\hat{p}(s'|s,a), p(s'|s,a)) \leq \alpha$ for all (s,a) pairs. Then for all policies π, we have

$$|\eta[\pi] - \hat{\eta}[\pi]| \leq \frac{2\gamma\alpha R_{max}}{(1-\gamma)^2}$$

where η and $\hat{\eta}$ are expected cumulative rewards under the true dynamics and learned dynamics.

A spectral proof

Proof.

Let W be the random walk matrix induced on the $(s, a) \mapsto (s', a')$ space under the true model and the policy π, and W' be the random walk matrix induced on the $(s, a) \mapsto (s', a')$ space under the learned model and policy π. Then we can write $W = (1 - \alpha)X + \alpha\mathcal{E}$ and $W' = (1 - \alpha)X + \alpha\mathcal{E}'$ where $\|X\|_1 \leq 1, \|\mathcal{E}\|_1 \leq 1, \|\mathcal{E}'\|_1 \leq 1$. Here, the reward on the true model will be

$$\sum_t r^\top \gamma^t W^t x,$$

where x is the initial distribution of state-action pairs. $\square$

A spectral proof, cont'd

Proof.

$$
\begin{aligned}
\eta[\pi] - \hat{\eta}[\pi] &= \sum_t \gamma^t (\mathbb{E}_{a\sim\pi(\tau^{(t)})}[R(\tau^{(t)}, a)] - \mathbb{E}_{a\sim\pi(\hat{\tau}^{(t)})}[R(\hat{\tau}^{(t)}, a)]) \\
&= r^\top \sum_t \gamma^t (W^t - W'^t)x \\
&= r^\top \sum_t \gamma^t (((1-\alpha)X + \alpha\mathcal{E})^t - ((1-\alpha)X + \alpha\mathcal{E}')^t)x \\
&= \|r\|_\infty \left\| \sum_t \gamma^t (((1-\alpha)X + \alpha\mathcal{E})^t - ((1-\alpha)X + \alpha\mathcal{E}'))^t)x \right\|_1 \\
&\le \|r\|_\infty \sum_t 2\gamma^t (1 - (1-\alpha)^t) \le \frac{2\gamma\alpha R_{\max}}{(1-\gamma)^2}
\end{aligned}
$$

$\square$

Weak Bounds?

Theorem

Assume that $D_{TV}(\hat{p}(s,a), p(s,a)) \leq \alpha$ for all (s,a) pairs. Then for all policies π, we have

$$|\eta[\pi] - \hat{\eta}[\pi]| \leq \frac{2\gamma\alpha R_{max}}{(1-\gamma)^2}$$

where η and $\hat{\eta}$ are rewards under the true dynamics and learned dynamics.

We note that these results are not particularly strong - as a trivial bound is given by

$$\frac{2\gamma R_{\max}}{1-\gamma}$$

– so the bound given is only better when $\alpha < 1 - \gamma$.

Stronger Bounds

- In practice, offline learners perform far better than these bounds would seem to indicate.
- If we write the discrepancy as

$$r^\top(\Delta p)$$

where r is the reward vector and (Δp) represents the difference in distributions of state-action pairs, the high-level approach taken by [KRNJ20], [YTY$^+$20], [LXL$^+$18] is to bound

$$r^\top(\Delta p) \leq \|r\|_\infty \|\Delta p\|_1 .$$

- What if we did

$$r^\top(\Delta p) \leq \|r\|_2 \|\Delta p\|_2?$$

L2 Bounds

- In some cases, the L2 bound will be quite weak. However, in some cases the L2 bound might be acceptable.
- The $\|r\|_2$ can be very small, specifically when the rewards are *sparse*, an often studied case in RL.
- In the extreme case, $\|r\|_2 \approx \|r\|_\infty$, in which case our bound is essentially without loss!

The Significance of Sparsity

In some situations, rewards are sparse, which creates challenges for standard reinforcement learning algorithms.

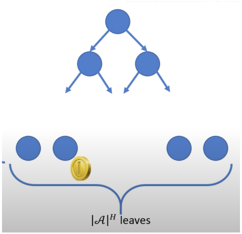

Figure: A hard example for online RL [KAL16]

For an online method, this is very hard because it can't look into the future - but a planning method can!

Stronger Bounds

Sometimes, however, the L2 bound may appear to be quite weak. Consider the cartpole task, in which an agent gets $+1$ reward for every timestep that the cartpole is upright.

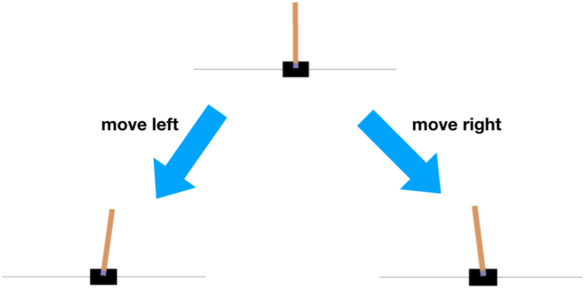

Figure: The Cartpole Task

Stronger Bounds

While the $\|r\|_2$ of the reward vector for this MDP appears to be high, observe that we can use vector decomposition –

$$r^\top(\Delta p) = (r^\parallel + r^\perp)^\top(\Delta p) = (r^\perp)^\top(\Delta p)$$

which allows us to bound

$$r^\top(\Delta p) \leq \left\| r^\perp \right\|_2 \|\Delta p\|_2$$

Furthermore, using L_2 norms allows us to more easily use results of spectral graph theory. To wit:

Our Result

Theorem ([KRNJ20])

Assume that $D_{TV}(\hat{p}(s'|s,a), p(s'|s,a)) \leq \alpha$ for all (s,a) pairs. Then for all policies π, we have

$$|\eta[\pi] - \hat{\eta}[\pi]| \leq \frac{2\gamma\alpha R_{max}}{(1-\gamma)^2}$$

where η and $\hat{\eta}$ are rewards under the true dynamics and learned dynamics.

Theorem (W.)

Under the same assumptions,

$$|\eta[\pi] - \hat{\eta}[\pi]| \leq \frac{2\gamma\alpha \left\|r^{\perp}\right\|_2 \left\|x_0^{\perp}\right\|_2}{(1-\gamma\omega)^2}$$

where η and $\hat{\eta}$ are rewards under the true dynamics and learned dynamics and $\omega = 1-\gamma$, where γ is the expansion of the MDP.

Conclusion

- It is possible that many MDPs that have low discrepancies in practice have an underlying sparse reward structure which explains their lower discrepancies.
- We intend to try our reward shaping on the tasks in the OpenAI Gym environment like in [FKN$^+$20], where we replace a reward for "speed" with a penalty for not having crossed the finish line.

Acknowledgements

- Yang Liu for helpful discussions.
- Salil Vadhan for his helpful suggestions, and for teaching this class.

References I

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine.
D4rl: Datasets for deep data-driven reinforcement learning.
arXiv preprint arXiv:2004.07219, 2020.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford.
Contextual-mdps for pacreinforcement learning with rich observations.
arXiv preprint arXiv:1602.02722, 2016.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims.
Morel: Model-based offline reinforcement learning.
NeurIPS, 2020.

References II

Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma.
Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees.
In *International Conference on Learning Representations*, 2018.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma.
Mopo: Model-based offline policy optimization.
NeurIPS, 2020.