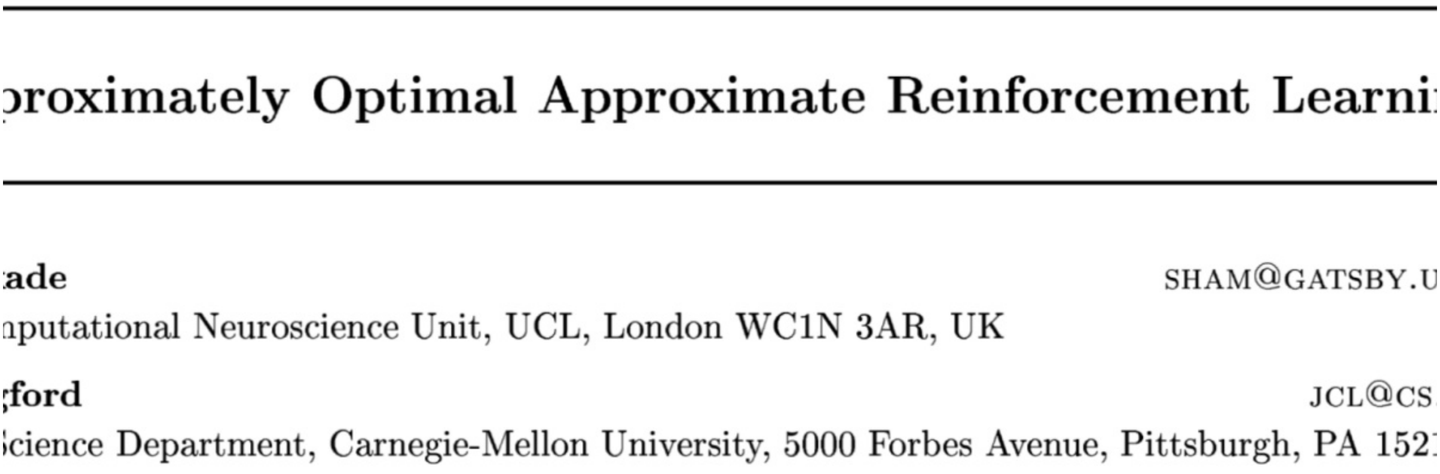
roximately Optimal Approximate Reinforcement Learni
ade SHAM@GATSBY.U
nputational Neuroscience Unit, UCL, London WC1N 3AR, UK
ford JCL@CS.
cience Department, Carnegie-Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 152

【强化学习 91】Kakade&Langford 02'

张楚珩
清华大学 交叉信息院博士在读

这么有名的一篇工作竟然没有笔记，今天重新看了一下，补一下笔记。

原文传送门

Kakade, Sham, and John Langford. "Approximately optimal approximate reinforcement learning." ICML. Vol. 2. 2002.

特色

提出了 conservative policy iteration（CPI），在理论分析上使用 restart distribution 来处理探索难的问题，并且把策略评估步（policy evaluation）和策略改进步（policy improvement ）打包形成一个 greedy policy chooser。规定这个 greedy policy chooser 的误差为 ，，代表这两步综合的 approximation error。文章证明了，当给定 ，时，CPI 方法：1）能在某种 metric 下保证 policy improvement；2）能在有限次调用 greedy policy chooser 之后结束；3）最终的策略 near-optimal。

过程

1. 主要想法

这篇文章提出 restart distribution 和 greedy policy chooser。具体细节后面都会再提到。

- Restart distribution：策略梯度方法把 exploration 和 exploitation 交杂在一起，即使用当前的（随机）策略进行 rollout 产生样本，然后计算在样本上的策略梯度，并用于更新当前样本。因此对于状态空间的探索全靠当前策略产生，如果当前策略使得某一些重要的状态很难被探索到，则很难学习到最优策略。Restart distribution 的主要想法是，既然全凭当前策略无法充分覆盖状态空间，那么可以让 rollout 的初始状态分布尽可能覆盖到整个状态空间，这样就从另一个角度解决了探索的难题（至少在理论分析上）。

- Greedy policy chooser：如果策略估计步估计准确（exact case），那么每次就选择相对于所估计的价值函数的 greedy policy 即可（one step improvement），这样的操作可以保证 policy improvement 和 optimality，这其实就是 dynamic programming。但是如果策略估计步不准确，那么每次还选择相对于这个不准确估计的 greedy policy，就不能再保证 policy improvement 了。如果策略估计步使用函数近似（approximate case），那么就不能保证最后选择出来的 greedy policy 对应最大的 one step improvement。文章假设在每一步上，该 greedy policy chooser 选择到的策略距离最大的 one step improvement 相差小于 ϵ 。
- 考虑以上两个部分之后，文章设计了 CPI 使得：1）能在某种 metric 下保证 policy improvement；2）能在有限次调用 greedy policy chooser 之后结束；3）最终的策略 near-optimal。

2. 之前的方法

文章说，希望能够设计一个算法使得理论上能够对于以下三个问题有保障：

(1) Is there some performance measure that is guaranteed to improve at every step?
(2) How difficult is it to verify if a particular update improves this measure?
(3) After a reasonable number of policy updates, what performance level is obtained?

第一个问题是讲，能不能保障 policy improvement，该 policy improvement 可以不是最后关心的那个 policy performance，而是自己定义的某种 policy performance measure。第二个问题是讲，如果能够保障 policy improvement，那么有没有什么判定依据使得我们知道这一轮还能不能产生相应的 policy improvement。它是在第一个问题基础上提出的，即只有第一个问题有肯定的回答，才会有第二个问题。第三个问题是说，能不能在有限次更新之后得到一个 optimality 的保障，即希望有一个 finite iteration analysis 而不是一个笼统的 convergence and asymptotic analysis。

2.1 Exact value function methods (e.g. policy iteration)

对于一个策略 π ，每次计算得到 $Q_\pi(s,a)$ ，然后把策略更新为相对于 $Q_\pi(s,a)$ 的 greedy policy

$$\pi'(a;s) \text{ such that } \pi'(a;s) = 1 \text{ iff } a \in \text{argmax}_a Q_\pi(s,a).$$

(Puterman, 1994) 书中说明这种方法能够保证收敛到 optimal。

2.2 Approximate value function methods

不再能够得到一个准确的价值函数估计，而是得到一个近似估计 $\tilde{V}(s)$ ，满足

$$\varepsilon = \max_s |\widetilde{V}(s) - V_\pi(s)|$$

π' 是 $\tilde{V}(s)$ 对应的 greedy policy（这里不太明白，V 函数如何对应相应的 greedy policy，大概认为这里是用的类似 Sutton 书 Chap 6.8 里面讲的 afterstate value function 吧）。这种情况下，不能保证 policy improvement，只能保证性能减小的不太多，即

$$(3.1) \qquad V_{\pi'}(s) \geq V_\pi(s) - \frac{2\gamma\varepsilon}{1-\gamma}.$$

从上式可以看出，exact case 时 $\epsilon=0$ ，能保证 policy improvement。太

这说明，该方法不能回答前述第 1、2 个问题。

2.3 Policy gradient methods

文章中说明要准确估计 policy gradient 的方向所需要的样本会非常地多。

文章举了一个 sparse reward 的例子，即，除了一个目标状态之外， 其他状态上的奖励都为零。

- On-policy：在该例子中，在策略 far from optimal 的时候，如果让策略去随机采样产生 rollout，需要指数级多的样本才能够采样到目标状态。如果采样不到目标状态，那么估计的策略梯度为零，这样策略也不会做任何更新。
- Off-policy：另外一种自然的想法就是使用更容易到达目标状态的 off-policy 的轨迹来计算策略梯度，同时为了弥补相应的分布的不同，再乘上 importance weight，这其中对应的 importance weight 就会指数级地小，这样在有用的策略梯度方向上只会产生一个很小的值。（文章中说这种情况下 importance weight 会指数级地大，我觉得是弄反了）

有文章提到策略梯度能够较为准确地被估计，不过注意到在文章中的例子上，策略梯度为零其实是一个比较准确的估计，但是它可能使得收敛指数级地慢。

3. Conservative policy iteration

3.1 Conservative policy update and policy improvement lower bound

这篇文章最重要的观察是：对于一个任意的策略 π' ，只要它能产生一个正的 one step improvement，那么就能够找到一个相应的 conservative update π_{new} ，使得新策略相比于旧策略有 policy improvement。具体的细节如下。

Policy improvement 可以写作（performance difference lemma）：

Lemma 6.1. *For any policies $\tilde{\pi}$ and π and any starting state distribution μ,*

$$\eta_\mu(\tilde{\pi}) - \eta_\mu(\pi) = \frac{1}{1-\gamma} E_{(a,s)\sim\tilde{\pi} d_{\tilde{\pi},\mu}} \left[A_\pi(s,a)\right]$$

而 one step improvement（policy advantage）可以写作：

$$\mathbb{A}_{\pi,\mu}(\pi') \equiv E_{s\sim d_{\pi,\mu}} \left[E_{a\sim\pi'(a;s)} \left[A_\pi(s,a)\right]\right] .$$

注意到两者的区别在于 state-action 的分布。在文中，最后关心的性能是在初始状态分布 D 下得到的性能，即 $\eta_D(\pi)$ ；这里考虑的是一个在状态空间中分布更为均匀的 restart distribution μ 下得到的性能，即 $\eta_\mu(\pi)$ 。

考虑一个 conservative update rule

$$\pi_{\text{new}}(a;s) = (1-\alpha)\pi(a;s) + \alpha\pi'(a;s) , \tag{4.1}$$

利用 policy gradient theorem （Sutton书）可以得到

$$\frac{\partial \eta_\mu}{\partial \alpha}\Big|_{\alpha=0} = \frac{1}{1-\gamma}\mathbb{A}_{\pi,\mu}$$

$$(4.2) \qquad \Delta\eta_\mu = \frac{\alpha}{1-\gamma}\mathbb{A}_{\pi,\mu}(\pi') + O(\alpha^2)\,.$$

这个性质告诉我们，只要这个任意的策略 π' 使得 one step improvement 大于零，那么我们至少能够用 $0<\alpha\ll 1$ 构造一个 conservative policy 使得该策略有大于零的 policy improvement。

那么选择一个怎样的 α 能够获取一个最大的 policy improvement lower bound 呢？ 顺着这个思路往下，考虑给定 α ，计算 policy improvement 的下界。

Theorem 4.1. *Let* $\mathbb{A}$ *be the policy advantage of* π' *with respect to* π *and* μ. *and let* $\varepsilon = \max_s |E_{a\sim\pi'(a;s)}[A_\pi(s,a)]|$. *For the update rule 4.1 and for all* $\alpha \in [0,1]$:

$$\eta_\mu(\pi_{new}) - \eta_\mu(\pi) \geq \frac{\alpha}{1-\gamma}\left(\mathbb{A} - \frac{2\alpha\gamma\varepsilon}{1-\gamma(1-\alpha)}\right).$$

由于这个定理比较重要，因此加了一个红框框。定理的推导也比较直观，policy improvement 和 one step policy improvement 差距产生的原因就在于策略不同导致相应的状态分布不一样。而根据 conservative policy update rule，这两个比较的策略每一步至少有 $1-\alpha$ 的概率选择相同的行动，从而产生相同的状态分布；而不同的那一部分遵循 π' ，而我们已知它所带来的 one step improvement；剔除这两部分之后，bound 剩余的项即可得到上述定理。

比较有意思的是当 时，上述定理变为

$$\eta_\mu(\pi_{\text{new}}) - \eta_\mu(\pi) \geq \frac{\mathbb{A}}{1-\gamma} - \frac{2\gamma\varepsilon}{1-\gamma}$$

其形式上和 (3.1) 类似，不过两处地方 ϵ 的含义不太一样，这里是 $\epsilon = \max_s |\mathbb{E}_{a\sim\pi'}[A_\pi(s,a)]| = \max_s |\mathbb{E}_{a\sim\pi'}[Q_\pi(s,a)] - V_\pi(s)|$ ， 而 (3.1) 中为 $\epsilon = \max_s |\tilde{V}(s) - V_\pi(s)|, \text{where } \pi' = greedy(\tilde{V})$ 。究竟是否讲的是一回事，有点分不清，原因是不太清楚 $\pi' = greedy(\tilde{V})$ 是怎么来的。

通过上述定理可以找到一个步长 α 使得 policy improvement lower bound 最大。

Corollary 4.2. *Let* R *be the maximal possible reward and* $\mathbb{A}$ *be the policy advantage of* π' *with respect to* π *and* μ. *If* $\mathbb{A} \geq 0$, *then using* $\alpha = \frac{(1-\gamma)\mathbb{A}}{4R}$ *guarantees the following policy improvement:*

$$\eta_\mu(\pi_{new}) - \eta_\mu(\pi) \geq \frac{\mathbb{A}^2}{8R}.$$

3.2 Greedy policy chooser

假设我们能够获得一个比较好的找到策略 π' 的方法，即能找到一个 π' 比最好的能找到的情况差注意到不了多少，即

Definition 4.3. An ε-**greedy policy chooser**, $G_\varepsilon(\pi, \mu)$, is a function of a policy π and a state distribution μ which returns a policy π' such that $\mathbb{A}_{\pi,\mu}(\pi') \geq \mathrm{OPT}(\mathbb{A}_{\pi,\mu}) - \varepsilon$, where $\mathrm{OPT}(\mathbb{A}_{\pi,\mu}) \equiv \max_{\pi'} \mathbb{A}_{\pi,\mu}(\pi')$.

注意，从这里开始一直到最后面，ϵ 都应该是代表 greedy policy chooser 的误差，和 (3.1) 中的 ϵ 以及 Theorem 4.1 中的 ϵ 都不一样。Theorem 4.1 中的 ϵ 用于得到 Corollary 4.2 之后就没再用到了。

该 greedy policy chooser 隐含了 policy evaluation 和 policy improvement 两项。它获取的方式可以先得到一个 value function approximator

$$E_{s\sim d_{\pi,\mu}} \max_a |A_\pi(s,a) - f_\pi(s,a)| .$$

如果它的误差能够控制到小于 $\epsilon/2$ ，那么选取关于它的 greedy policy 就可以组成一个 ϵ-greedy policy chooser。显然 greedy policy chooser 也是需要一定的 sample complexity 来完成的，文章把它作为一个黑盒子，没有具体分析。

3.3 Sketch of CPI

CPI 的大致步骤如下：

(1) Call $G_\varepsilon(\pi, \mu)$ to obtain some π'
(2) Estimate the policy advantage $\mathbb{A}_{\pi,\mu}(\pi')$
(3) If the policy advantage is small (less than ε), STOP and return π.
(4) Else, update the policy and go to (1).

由于对于 $\mathbb{A}$ 的估计可能会有一些误差，具体细节如下：

(1) Call $G_\varepsilon(\pi, \mu)$ to obtain some π'
(2) Use $O(\frac{R^2}{\varepsilon^2} \log \frac{R^2}{\delta\varepsilon^2})$ μ-restarts to obtain an $\frac{\varepsilon}{3}$-accurate estimate $\hat{\mathbb{A}}$ of $\mathbb{A}_{\pi,\mu}(\pi')$.
(3) If $\hat{\mathbb{A}} < \frac{2\varepsilon}{3}$, STOP and return π.
(4) If $\hat{\mathbb{A}} \geq \frac{2\varepsilon}{3}$, then update policy π according to equation 4.1 using $\frac{(1-\gamma)(\hat{\mathbb{A}}-\frac{\varepsilon}{3})}{4R}$ and return to step 1.

关于该算法有如下定理：

Theorem 4.4. *With probability at least $1-\delta$, conservative policy iteration: i) improves η_μ with every policy update, ii) ceases in at most $72\frac{R^2}{\varepsilon^2}$ calls to $G_\varepsilon(\pi,\mu)$, and iii) returns a policy π such that $OPT(\mathbb{A}_{\pi,\mu}) < 2\varepsilon$.*

该定理回答了之前的几个问题。

首先，虽然我们关心的是 但是在使用restart distribution μ 之后，可以保证 η_μ 有 policy improvement。

其次，第 2 步中的结论只需利用 Hoeffding 不等式即可。当第 4 步中的条件满足时，可以保证 $\mathbb{A}$ 的真实值一定大于 $\epsilon/3$ ，这样能够保证每次 improve 的量都不小于 $\frac{\epsilon^2}{72R}$ ，其中 R 为最大的 cumulative reward。另外注意到总体 performance 有上界，因此可以算出迭代次数的上界。

最后，当第 3 步中的条件满足时，可以证 $\mathbb{A}$ 的真实值一定小于 ϵ ，再考虑到 policy chooser 为 ϵ -greedy，因此可以得知 $OPT(\mathbb{A})<2\epsilon$ 。我们会看到，这个条件可以被转化为 optimality 条件。

3.4 Optimality

如果从某个 restart distribution 出发，某个策略对应的 $\mathbb{A}_{\pi,\mu}$ 再找不到可以使它大幅改进策略，那么该策略的性能比较接近最优策略的性能。不过前提是 restart distribution 需要和最优策略下的稳态分布比较接近。以下定理说明了这一点，注意把它和专栏前一篇里面的 gradient domination 作比较，gradient domination 说明了 gradient 小的时候接近最优。

Corollary 4.5. *Assume that for some policy π, $OPT(\mathbb{A}_{\pi,\mu}) < \varepsilon$. Let π^* be an optimal policy. Then*

$$\begin{aligned}\eta_D(\pi^*) - \eta_D(\pi) &\le \frac{\varepsilon}{(1-\gamma)}\left\|\frac{d_{\pi^*,D}}{d_{\pi,\mu}}\right\|_\infty \\ &\le \frac{\varepsilon}{(1-\gamma)^2}\left\|\frac{d_{\pi^*,D}}{\mu}\right\|_\infty .\end{aligned}$$

其证明过程和 gradient domination 类似。

4. 讨论

原则上来说存在一个最优策略使得关于 η_μ 和 能够被同时最大化，但是 CPI 只保证每回合 η_μ 有 policy improvement。那要是把算法里面的 μ 都换成 D 呢？首先，这样不能保证得到的策略较优。

当 μ 分布比较均匀的时候，能保证不过最优策略下的稳态分布如何，Corolloary 4.5 中的上界都比较小，而一个分布不太均匀的 D 不能保证这一点。其次，有时候能产生 large advantage 的状态不是从 D 出发能经常访问到的状态（考虑前面 sparse reward 的例子），因此相比于 $OPT(\mathbb{A}_{\pi,\mu})$ ，不容易得到一个较大的 $OPT(\mathbb{A}_{\pi,D})$ 。

如何选择 restart distribution 呢？可以根据先验选择最优策略容易访问到的状态。

发布于 2019-08-23