

A Theory of Regularized Markov Decision Processes

Matthieu Geist¹ Bruno Scherrer² Olivier Pietquin¹

【强化学习 95】Regularized MDP



张楚珩

清华大学 交叉信息院博士在读

26 人赞同了该文章

正则化的 Bellman 算子，导出了一系列的常见算法，比如 TRPO、SQL、SAC、DPP 等。

原文传送门

Geist, Matthieu, Bruno Scherrer, and Olivier Pietquin. "A Theory of Regularized Markov Decision Processes." arXiv preprint arXiv:1901.11275 (2019).

特色

搞了一套理论能够涵盖之前的很多算法；用到了 Legendre-Fenchel transform；能够分析算法的 error propagation。

过程

1. Legendre-Fenchel transform

考虑一个内积 $\langle \cdot, \cdot \rangle: \Delta_{\mathcal{A}} \times \mathbb{R}^{\mathcal{A}} \rightarrow \mathbb{R}$ 。对于一个 strongly convex 的函数 $\Omega: \Delta_{\mathcal{A}} \rightarrow \mathbb{R}$ ，它的 convex conjugate $\Omega^*: \mathbb{R}^{\mathcal{A}} \rightarrow \mathbb{R}$ 为

$$\forall q_s \in \mathbb{R}^{\mathcal{A}}, \Omega^*(q_s) = \max_{\pi_s \in \Delta_{\mathcal{A}}} \langle \pi_s, q_s \rangle - \Omega(\pi_s).$$

可以证明它满足如下性质：

- i *Unique maximizing argument:* $\nabla \Omega^*$ is Lipschitz and satisfies $\nabla \Omega^*(q_s) = \operatorname{argmax}_{\pi_s \in \Delta_{\mathcal{A}}} \langle \pi_s, q_s \rangle - \Omega(\pi_s)$.
- ii *Boundedness:* if there are constants L_Ω and U_Ω such that for all $\pi_s \in \Delta_{\mathcal{A}}$, we have $L_\Omega \leq \Omega(\pi_s) \leq U_\Omega$, then $\max_{a \in \mathcal{A}} q_s(a) - U_\Omega \leq \Omega^*(q_s) \leq \max_{a \in \mathcal{A}} q_s(a) - L_\Omega$.
- iii *Distributivity:* for any $c \in \mathbb{R}$ (and $\mathbf{1}$ the vector of ones), we have $\Omega^*(q_s + c\mathbf{1}) = \Omega^*(q_s) + c$.
- iv *Monotonicity:* $q_{s,1} \leq q_{s,2} \Rightarrow \Omega^*(q_{s,1}) \leq \Omega^*(q_{s,2})$ 知乎 @张楚珩

【证明】第一条，令 $\pi' = \operatorname{argmax}_{\pi} \langle \pi, q_s \rangle - \Omega(\pi)$ ，然后又 $\Omega'(q_s) = \langle \pi', q_s \rangle - \Omega(\pi')$ ，两边求导可得 $\pi' = \nabla \Omega'(q_s)$ 。第二条容易。第三条展开写出来就可以了，注意到 $\langle \pi, \mathbf{1} \rangle = 1$ 。第四条注意到向量的小于等于表示每个元素都小于等于，如果只有 q 中的一个元素变大，根据 π 的非负性，最后的内积肯定也变大，因此有单调性。

2. 算子

Regularized Bellman Operator

$$T_{\pi, \Omega} v = T_{\pi} v - \Omega(\pi) = \langle \pi, q \rangle - \Omega(\pi), \quad q = r + \gamma P v$$

注意到这个式子是一个向量式子，它对每一个 $s \in \mathcal{S}$ 成立。

Regularized value function

Regularized value 是 regularized Bellman operator 的不动点，即

Definition 2 (Regularized value function of policy π).
 Noted $v_{\pi, \Omega}$, it is defined as the unique fixed point of the operator $T_{\pi, \Omega}$: $v_{\pi, \Omega} = T_{\pi, \Omega} v_{\pi, \Omega}$. We also define the associated state-action value function $q_{\pi, \Omega}$ as

$$q_{\pi, \Omega}(s, a) = r(s, a) + \gamma \mathbb{E}_{s' | s, a} [v_{\pi, \Omega}(s')] \\
\text{with } v_{\pi, \Omega}(s) = \mathbb{E}_{a \sim \pi(\cdot | s)} [q_{\pi, \Omega}(s, a)] - \Omega(\pi(s)) \text{ 知乎 @张楚珩}$$

Regularized Bellman optimality Operator

$$T_{*,\Omega} : v \in \mathbb{R}^S \rightarrow T_{*,\Omega} v = \max_{\pi \in \Delta_{\mathcal{A}}^S} T_{\pi,\Omega} v = \Omega^*(q) \in \mathbb{R}^S,$$

Regularized optimal value function

Regularized optimal value function 是 Regularized Bellman optimality Operator 的不动点，即

Definition 3 (Regularized optimal value function). *Noted $v_{*,\Omega}$, it is the unique fixed point of the operator $T_{*,\Omega} : v_{*,\Omega} = T_{*,\Omega} v_{*,\Omega}$. We also define the associated state-action value function $q_{*,\Omega}(s, a)$ as*

$$q_{*,\Omega}(s, a) = r(s, a) + \gamma \mathbb{E}_{s'|s,a} [v_{*,\Omega}(s')] \\ \text{with } v_{*,\Omega}(s) = \Omega^*(q_{*,\Omega}(s, \cdot)).$$

知乎 @张楚珩

Greedy policy

$$\pi' = \mathcal{G}_{\Omega}(v) = \nabla \Omega^*(q) \Leftrightarrow T_{\pi',\Omega} v = T_{*,\Omega} v,$$

性质

Proposition 2. *The operator $T_{\pi,\Omega}$ is affine and we have the following properties.*

i *Monotonicity: let $v_1, v_2 \in \mathbb{R}^S$ such that $v_1 \geq v_2$. Then,*

$$T_{\pi,\Omega}v_1 \geq T_{\pi,\Omega}v_2 \text{ and } T_{*,\Omega}v_1 \geq T_{*,\Omega}v_2.$$

ii *Distributivity: for any $c \in \mathbb{R}$, we have that*

$$\begin{aligned} T_{\pi,\Omega}(v + c\mathbf{1}) &= T_{\pi,\Omega}v + \gamma c\mathbf{1} \\ \text{and } T_{*,\Omega}(v + c\mathbf{1}) &= T_{*,\Omega}v + \gamma c\mathbf{1}. \end{aligned}$$

iii *Contraction: both operators are γ -contractions in supremum norm. For any $v_1, v_2 \in \mathbb{R}^S$,*

$$\begin{aligned} \|T_{\pi,\Omega}v_1 - T_{\pi,\Omega}v_2\|_\infty &\leq \gamma \|v_1 - v_2\|_\infty \\ \text{and } \|T_{*,\Omega}v_1 - T_{*,\Omega}v_2\|_\infty &\leq \gamma \|v_1 - v_2\|_\infty. \end{aligned}$$

Theorem 1 (Optimal regularized policy). *The policy $\pi_{*,\Omega} = \mathcal{G}_\Omega(v_{*,\Omega})$ is the unique optimal regularized policy, in the sense that for all $\pi \in \Delta_{\mathcal{A}}^S$, $v_{\pi_{*,\Omega},\Omega} = v_{*,\Omega} \geq v_{\pi,\Omega}$.*

Proposition 3. *Assume that $L_\Omega \leq \Omega \leq U_\Omega$. Let π be any policy. We have that $v_\pi - \frac{U_\Omega}{1-\gamma}\mathbf{1} \leq v_{\pi,\Omega} \leq v_\pi - \frac{L_\Omega}{1-\gamma}\mathbf{1}$ and $v_* - \frac{U_\Omega}{1-\gamma}\mathbf{1} \leq v_{*,\Omega} \leq v_* - \frac{L_\Omega}{1-\gamma}\mathbf{1}$.*

Theorem 2. Assume that $L_\Omega \leq \Omega \leq U_\Omega$. We have that

$$v_* - \frac{U_\Omega - L_\Omega}{1 - \gamma} \leq v_{\pi_*, \Omega} \leq v_*. \quad \text{知乎 @张楚珩}$$

3. Regularized Modified Policy Iteration

Policy iteration 和 value iteration 可以被统一写成如下形式。

$$\begin{cases} \pi_{k+1} = \mathcal{G}_\Omega(v_k) \\ v_{k+1} = (T_{\pi_{k+1}, \Omega})^m v_k \end{cases} \quad (1)$$

当 $m=1$ 时，为 value iteration，上下两个方程合并之后就是 $v_{k+1} = T_{\pi, \Omega} v_k$ ；当 $m=\infty$ 时，为 policy iteration，即分为 policy improvement 和 policy evaluation 两步。

Value iteration

先讨论 $m=1$ 的情形，即 value iteration，这里参数化的是 Q 函数。在 unregularized 的情况下，这种情形就是 Q-learning；相应地，regularized 时为

$$\begin{aligned} J(\theta) &= \hat{\mathbb{E}} \left[(\hat{q}_i - q_\theta(s_i, a_i))^2 \right] \\ \text{with } \hat{q}_i &= r_i + \gamma \Omega^*(q_{\bar{\theta}}(s'_i, \cdot)). \end{aligned} \quad (2)$$

注意到，区别在于这里用 Ω^* 代替了原来的 $\max_{\pi} q(s', a)$ ，其实道理是一样的，因为本身其定义就是 maximum over all policies。

考虑正则项为 negative entropy $\Omega(\pi_s) = -\sum_a \pi_s(a) \ln \pi_s(a)$ ，可以解得 $\Omega^*(q_s) = \ln \sum_a \exp q_s(a)$ 。这样，上述算法就对应的是 soft Q-learning。

Policy iteration

还可以做 policy iteration，一般用 actor-critic 方法，参数化策略和价值函数。价值函数的更新和前面类似，只不过不是做 $T_{\pi, \Omega}$ 而是 $T_{\pi, \rho}$ ，即

$$\begin{aligned} J(\theta) &= \hat{\mathbb{E}}[(\hat{q}_i - q_\theta(s_i, a_i))^2] \\ \text{with } \hat{q}_i &= r_i + \gamma(\mathbb{E}_{a \sim \pi(\cdot | s'_i)}[q_\theta(s'_i, a)] - \Omega(\pi(\cdot, s'_i))). \end{aligned} \quad (4)$$

策略的更新有两种方式：在能解析地写出 greedy policy 的形式 $\pi^* = \nabla \Omega^*(q)$ 的时候，策略的更新可以直接最小化参数化策略和 greedy policy 之间的距离（SAC 和 MPO 就是这样做的）

$$J(w) = \hat{\mathbb{E}}[\text{KL}(\pi_w(\cdot|s_i) || \nabla \Omega^*(q_k(s_i, \cdot)))]. \quad (3)$$

在不能解析地写出的时候，可以直接策略梯度去优化如下目标（TRPO 就是把下面的目标转化为了 hard constraint 来解）

$$J(w) = \hat{\mathbb{E}} [\mathbb{E}_{a \sim \pi_w(\cdot|s_i)} [q_k(s_i, a)] - \Omega(\pi_w(\cdot|s_i))]. \quad (5)$$

Error propagation

这一块其实没太弄明白。总体来说，想研究的问题是，如果 policy improvement 和 policy evaluation 这两个步骤都有一定的误差，那么误差会如何传播，以至于影响最后找到的策略的性能。即考虑带误差的 modified policy iteration:

$$\begin{cases} \pi_{k+1} = \mathcal{G}_{\Omega}^{\epsilon'_{k+1}}(v_k) \\ v_{k+1} = (T_{\pi_{k+1}, \Omega})^m v_k + \epsilon_{k+1} \end{cases}, \quad (6)$$

其中，policy evaluation 步的误差项比较好理解，就是估计的不准产生的误差；policy improvement 步的误差项就是说不存在另一个策略使得 $T_{\pi, \Omega} v_k \leq T_{\pi_{k+1}, \Omega} v_k + \epsilon'_{k+1}$ 。分析的是 k 步之后所找到策略的性能相比于不动点的差距 $v_{k, \Omega} - v_{\pi^*, \Omega}$ 。其差距当然和这两个误差项有关，其中还和一个 concentrability coefficient 有关，这个系数衡量了策略的探索/环境的随机程度，一般加了正则之后的策略会更随机，探索会强一些，从而产生更小一些的 concentrability。（这一项和专栏前面讲的 PG Theory 里面的 distribution mismatch coefficient 类似）

4. Mirror Descent Modified Policy Iteration

前面考虑的 convex regularization Ω 是一个固定的凸函数，这里考虑它每次迭代变化。每次做 policy improvement 的时候，限定得到的策略 π 需要和前一轮的策略差距不太大。这个其实就是 conservative policy iteration，一样的道理。即

$$\pi_{k+1} = \mathcal{G}_{\Omega_{\pi_k}}(v_k), \text{ that is } \pi_{k+1} = \operatorname{argmax}_{\pi} \langle q_k, \pi \rangle - D_{\Omega}(\pi || \pi_k)$$

其中

$$\Omega_{\pi'}(\pi) = D_{\Omega}(\pi||\pi') = \Omega(\pi) - \Omega(\pi') - \langle \nabla \Omega(\pi'), \pi - \pi' \rangle.$$

注意到，如果 π 是 negative entropy 的话， $\Omega_{\pi'}(\pi) = KL(\pi||\pi')$ 就是 KL divergence，对应的 $\Omega_{\pi'}(g_a) = \ln \sum_a \pi_a(a) \exp g_a(a)$ ，对应的 greedy policy 为 $\nabla \Omega_{\pi'}(g_a) = \frac{\pi_a \exp g_a}{\sum_a \pi_a(a) \exp g_a(a)}$ 。

文章给出了两种对应的算法：

$$\begin{cases} \pi_{k+1} = \mathcal{G}_{\Omega_{\pi_k}}(v_k) \\ v_{k+1} = (T_{\pi_{k+1}, \Omega_{\pi_k}})^m v_k \end{cases}, \begin{cases} \pi_{k+1} = \mathcal{G}_{\Omega_{\pi_k}}(v_k) \\ v_{k+1} = (T_{\pi_{k+1}})^m v_k \end{cases}$$

注意到 $T_{\pi_{k+1}, \Omega_{\pi_{k+1}}} = T_{\pi_{k+1}}$ 。

前一种除了在 policy improvement 步中考虑正则，在 policy evaluation 的时候也考虑一样的正则，这种情况下，当 $m=1$ 时，两步就可以合并为 $v_{k+1} = T_{\pi_{k+1}} v_k$ 。DPP 就是这一种 $m=1$ 的情形。

后一种情况中，policy evaluation 步中估计的是正常的函数。TRPO、MPO 属于这种情形。

后面的 error propagation 实在没看懂，就不写了。

发布于 2019-10-16

强化学习 (Reinforcement Learning)

算法

机器学习

赞同 26

▼

1 条评论

分享

喜欢

收藏

...

文章被以下专栏收录



强化学习前沿
读呀读paper

进入专栏