Q-learning with Nearest Neighbors

Devavrat Shah *
Massachusetts Institute of Technology
devavrat@mit.edu

Qiaomin Xie *
Massachusetts Institute of Technology
qxie@mit.edu

【强化学习 81】NNQL

张楚珩
清华大学 交叉信息院博士在读

NNQL 全称如标题所示，即 Nearest Neighbor Q-Learning。

原文传送门

Shah, Devavrat, and Qiaomin Xie. "Q-learning with nearest neighbors." Advances in Neural Information Processing Systems. 2018.

特色

设计了一个算法，使用 nearest neighbor regression 来拟合 Q 函数，做 Q-learning，并对其做了理论分析。该工作有以下几方面特色：

- **Continuous state space:** 处理连续状态空间问题；
- **Unknown transition kernel:** 是一个 model-free 的方法；
- **Single sample path:** 通过对环境的采样得到的样本来学习，采样可以通过任意的（甚至是 non-stationary 的）策略得到，并且不要求环境是可以重置状态的，即样本是若干单一马可夫链上的一堆样本；
- **Online update:** 每步采样之后都可能对价值函数进行更新，是一个完全 online 的算法，不要求需要 batch 地采样之后再一块更新；
- **Non-asymptotic guarantees:** 理论分析上不仅仅是 asymptotic convergence analysis，而是 finite sample analysis。

相比其他工作的总结如下：

Table 1: Summary of relevant work. See Appendix A for details.

Specific work	Method	Continuous state space	Unknown transition Kernel	Single sample path	Online update	Non-asymptotic guarantees
[10], [36], [37]	Finite-state approximation	Yes	No	No	Yes	Yes
[43], [22], [41]	Q-learning	No	Yes	Yes	Yes	No
[20], [3], [18]	Q-learning	No	Yes	Yes	Yes	Yes
[23]	Q-learning	No	Yes	No	Yes	Yes
[42], [28]	Q-learning	Yes	Yes	Yes	Yes	No
[33], [32]	Kernel-based approximation	Yes	Yes	No	No	No
[19]	Value/Policy iteration	No	Yes	No	No	Yes
[44]	Parameterized TD-learning	No	Yes	Yes	Yes	No
[12]	Parameterized TD-learning	No	Yes	No	Yes	Yes
[8]	Parameterized TD-learning	No	Yes	Yes	Yes	Yes
[9]	Non-parametric LP	No	Yes	No	No	Yes
[30]	Fitted value iteration	Yes	Yes	No	No	Yes
[1]	Fitted policy iteration	Yes	Yes	Yes	No	Yes
Our work	Q-learning	Yes	Yes	Yes	Yes	Yes

过程

1. 问题设定

由于是连续状态空间，因此 transition probability 的定义可以写为：

$$\Pr(x_{t+1} \in B | x_t = x, a_t = a) = \int_B p(y|x,a)\lambda(dy) \tag{1}$$

由于需要分析连续状态空间上的问题，因此需要对 MDP 各种动力学在状态空间 $\mathcal{X}$ 上的连续性做一定的假设，即下面讲的 MDP Regularity，它和前面 DeepMDP 里面讲到的 Lipschitz MDP 类似。

Assumption 1 (MDP Regularity). *We assume that: (A1.) The continuous state space $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$; (A2.) $\mathcal{A}$ is a finite set of cardinality $|\mathcal{A}|$; (A3.) The one-stage reward R_t is non-negative and uniformly bounded by $R_{\max}$, i.e., $0 \le R_t \le R_{\max}$ almost surely. For each $a \in \mathcal{A}$, $r(\cdot, a) \in \mathrm{Lip}(\mathcal{X}, M_r)$ for some $M_r > 0$. (A4.) The transition probability kernel p satisfies*

$$|p(y|x,a) - p(y|x',a)| \le W_p(y)\rho(x,x'), \qquad \forall a \in \mathcal{A}, \forall x, x', y \in \mathcal{X},$$

where the function $W_p(\cdot)$ satisfies $\int_{\mathcal{X}} W_p(y)\lambda(dy) \le M_p$.

其中

$$\mathrm{Lip}(E, M) = \{f \in C(E) \mid |f(x) - f(y)| \le M\rho(x,y),\ \forall x, y \in E\}.$$

表示在 E 空间上 M-Lipschitz 连续的函数集合。

连续状态空间下的 Bellman optimality operator 可以写为

$$(FQ)(x,a) = r(x,a) + \gamma \mathbb{E}\left[\max_{b\in\mathcal{A}} Q(x',b) \mid x,a\right] = r(x,a) + \gamma \int_{\mathcal{X}} p(y|x,a) \max_{b\in\mathcal{A}} Q(y,b)\lambda(dy).$$

一个比较“平滑”的 MDP 产生的 optimal Q 函数也会比较平滑，即

Lemma 1. *Under Assumption 1, the function Q^* satisfies that $\|Q^*\|_\infty \le V_{\max}$ and that $Q^*(\cdot,a) \in \mathrm{Lip}(\mathcal{X}, M_r + \gamma V_{\max} M_p)$ for each $a \in \mathcal{A}$.*

（这个 lemma 我之前自己也得到了，并且对于一个给定 policy 的 Q 函数 Q^π 上述结论也成立）

2. NNQL 算法相关的定义

State space discretization and nearest neighbor regression

由于处理的是连续状态空间，因此需要先找出一系列散布在状态空间上的点，然后根据这些点上的 Q 函数来定义整个状态空间上的 Q 函数。

首先定义一系列散步在状态空间上的点，使得以这些点为球心的小球能够把整个状态空间布满。定义（以及一些其他的符号）如下：

Let $h > 0$ be a pre-specified scalar parameter. Since the state space $\mathcal{X}$ is compact, one can find a finite set $\mathcal{X}_h \triangleq \{c_i\}_{i=1}^{N_h}$ of points in $\mathcal{X}$ such that

$$\min_{i\in[N_h]} \rho(x,c_i) < h, \ \forall x \in \mathcal{X}.$$

The finite grid $\mathcal{X}_h$ is called an h-net of $\mathcal{X}$, and its cardinality $n \equiv N_h$ can be chosen to be the h-covering number of the metric space $(\mathcal{X},\rho)$. Define $\mathcal{Z}_h = \mathcal{X}_h \times \mathcal{A}$. Throughout this paper, we denote by $\mathcal{B}_i$ the ball centered at c_i with radius h; that is, $\mathcal{B}_i \triangleq \{x \in \mathcal{X} : \rho(x,c_i) \le h\}$.

知道这个集合中各个点上 Q 函数 $q = \{q(c_i,a), c_i \in \mathcal{X}, a \in \mathcal{A}\}$ 之后，对于任意一个状态上的 Q 函数都可以用 q 表示出来，即

$$(\Gamma_{\mathrm{NN}} q)(x,a) = \textstyle\sum_{i=1}^n K(x,c_i) q(c_i,a), \qquad \forall x \in \mathcal{X}, a \in \mathcal{A}, \tag{2}$$

Γ_{NN} 算子把一个定义在 $\mathcal{X}_h \times \mathcal{A}$ 上的函数映射为一个定义在 $\mathcal{X} \times \mathcal{A}$ 上的函数。其中 K 是 kernel function，满足 $K(x,c_i) \ge 0, \sum_{i=1}^n K(x,c_i) = 1$ ，并且认为一个点上的价值函数之和其附近一个小邻域内集合中的点的价值函数有关，即，

$$K(x,y) = 0 \ \text{ if } \ \rho(x,y) \ge h, \qquad \forall x \in \mathcal{X}, y \in \mathcal{X}_h, \tag{4}$$

Γ_{NN} 算子是 non-expansive 的，因为该算子求了加权平均，这样两个函数差的最大值在作用该算子之和不会变得更大。

$$\|\Gamma_{\mathrm{NN}} q - \Gamma_{\mathrm{NN}} q'\|_\infty \le \|q - q'\|_\infty, \qquad \forall q, q' \in C(\mathcal{X}_h \times \mathcal{A}). \tag{3}$$

Joint Bellman-NN operator

有了上述定义之后，给定集合 $\mathcal{Z}_h := \mathcal{X}_h \times \mathcal{A}$ 上的函数值 $q = \{q(c_i, a), c_i \in \mathcal{X}, a \in \mathcal{A}\}$ 之后，就能够确定所有的 Q 函数了，即

$$\tilde{Q}(x, a) = (\Gamma_{\mathrm{NN}} q)(x, a), \quad \forall (x, a) \in \mathcal{Z}.$$

Q-learning 的目标就是迭代地学习到 optimal Q function。在这里，即为迭代地学习到集合 $\mathcal{Z}_h$ 上的 optimal Q function。类似地，需要一个定义在集合 $\mathcal{Z}_h$ 上的 Q-learning 更新公式。

$$(Gq)(c_i, a) \triangleq (F\Gamma_{\mathrm{NN}} q)(c_i, a) = (F\tilde{Q})(c_i, a) = r(c_i, a) + \gamma \mathbb{E}\left[\max_{b \in \mathcal{A}} (\Gamma_{\mathrm{NN}} q)(x', b) \mid c_i, a\right]. \quad (5)$$

从该公式可以看出，要更新 (c_i, a) 上的函数值，需要至少一个 transition 样本 (x, a, x', r)，其中 $x \in \mathcal{B}_i$。如果要整体更新一轮，需要集合 $\mathcal{Z}_h$ 中所有元素都被至少采样到一次之后才能更新。那么需要多长时间集合 $\mathcal{Z}_h$ 中所有元素才能都被采样到呢？这就是下面要研究的问题。

Covering time of discretized MDP

这里定义的 covering time $\tau_{\pi,h}(x,t)$ 就描述了在时间 t 时刻，从状态 x 出发，需要经过多少步，才能够使得集合 $\mathcal{Z}_h$ 中所有元素都被采样到一次。注意到，这里的采样方式是使用一个任意的 policy π 来采样，并且该 policy 可能是 non-stationary 的，因此 covering time 中含有当前的时刻 t。

Definition 1 (Covering time of discretized MDP). *For each $1 \le i \le n = N_h$ and $a \in \mathcal{A}$, a ball-action pair $(\mathcal{B}_i, a)$ is said to be visited at time t if $x_t \in \mathcal{B}_i$ and $a_t = a$. The discretized state-action space $\mathcal{Z}_h$ is covered by the policy π if all the ball-action pairs are visited at least once under the policy π. Define $\tau_{\pi,h}(x,t)$, the covering time of the MDP under the policy π, as the minimum number of steps required to visit all ball-action pairs starting from state $x \in \mathcal{X}$ at time-step $t \ge 0$. Formally, $\tau_{\pi,h}(x,t)$ is defined as*

$$\min\left\{s \ge 0 : x_t = x,\ \forall i \le N_h, a \in \mathcal{A},\ \exists t_{i,a} \in [t, t+s],\ \text{such that } x_{t_{i,a}} \in B_i \text{ and } a_{t_{i,a}} = a,\ \text{under } \pi\right\},$$

with notation that minimum over empty set is ∞.

关于 covering time，论文假设它的期望是有限的。即，要解决的 MDP 或者产生样本所用的策略不能够使得状态空间有些区域访问不到，或者某些行动从不采用。

Assumption 2. *There exists an integer $L_h < \infty$ such that $\mathbb{E}[\tau_{\pi,h}(x,t)] \le L_h, \forall x \in \mathcal{X}, t > 0$. Here the expectation is defined with respect to randomness introduced by Markov kernel of MDP as well as the policy π.*

下面两个命题说明了，如果 MDP 满足一定程度上的遍历性条件的话，ϵ-greedy 的策略就可以使得 covering time 的期望有限。其中第一个命题对于 MDP 的遍历性假设更为严格，要求从某个状态出发，***对于任何的策略***，经过一些步，都能有一定的概率到达每个 $\mathcal{B}_i$。相应的 covering time 期望的上界更紧。第二个命题对于遍历性的假设更松，只要求从某个状态出发，***存在一个行动序列***，经过一些步，能有一定的概率到达每个 $\mathcal{B}_i$。相应的 covering time 的期望上界更松。

Proposition 1. *Suppose that the MDP satisfies the following: there exists a probability measure ν on $\mathcal{X}$, a number $\varphi > 0$ and an integer $m \geq 1$ such that for all $x \in \mathcal{X}$, all $t \geq 0$ and all policies μ,*

$$\Pr_{\mu}\left(x_{m+t} \in \cdot \,|\, x_t = x\right) \geq \varphi\nu(\cdot). \tag{6}$$

Let $\nu_{\min} \triangleq \min_{i\in[n]} \nu(\mathcal{B}_i)$, where we recall that $n \equiv N_h = |\mathcal{X}_h|$ is the cardinality of the discretized state space. Then the expected covering time of ε-greedy is upper bounded by $L_h = O\Big(\frac{m|\mathcal{A}|}{\varepsilon\varphi\nu_{\min}} \log(n|\mathcal{A}|)\Big)$.

Proposition 2. *Suppose that the MDP satisfies the following: there exists a probability measure ν on $\mathcal{X}$, a number $\varphi > 0$ and an integer $m \geq 1$ such that for all $x \in \mathcal{X}$, all $t \geq 0$, there exists a sequence of actions $\hat{\mathbf{a}}(x) = (\hat{a}_1, \ldots, \hat{a}_m) \in \mathcal{A}^m$,*

$$\Pr\left(x_{m+t} \in \cdot \,|\, x_t = x, a_t = \hat{a}_1, \ldots, a_{t+m-1} = \hat{a}_m\right) \geq \varphi\nu(\cdot). \tag{7}$$

Let $\nu_{\min} \triangleq \min_{i\in[n]} \nu(\mathcal{B}_i)$, where we recall that $n \equiv N_h = |\mathcal{X}_h|$ is the cardinality of the discretized state space. Then the expected covering time of ε-greedy is upper bounded by $L_h = O\Big(\frac{m|\mathcal{A}|^{m+1}}{\varepsilon^{m+1}\varphi\nu_{\min}} \log(n|\mathcal{A}|)\Big)$.

（证明思路如下：由于计算 covering time 上界，可以认为各个 $\mathcal{B}_i$ 是不相交的。先考虑已经访问了 $\mathcal{Z}_h$ 中的一部分元素，策略再走一步，能够访问到新元素的概率。然后再把它表示为需要经过多少步才能够访问到一个新元素，求和，即可得到把 $\mathcal{Z}_h$ 中每个元素访问一遍需要的时间）

3. NNQL 算法

算法如下图所示

Policy 1 Nearest-Neighbor Q-learning

Input: Exploration policy π, discount factor γ, number of steps T, bandwidth parameter h, and initial state Y_0.
Construct discretized state space $\mathcal{X}_h$; initialize $t = k = 0,\ \alpha_0 = 1,\ q^0 \equiv 0$;
Foreach $(c_i, a) \in \mathcal{Z}_h$, **set** $N_0(c_i, a) = 0$; **end**
repeat
  Draw action $a_t \sim \pi(\cdot|Y_t)$ and observe reward R_t; generate the next state $Y_{t+1} \sim p(\cdot|Y_t, a_t)$;
  Foreach i **such that** $Y_t \in \mathcal{B}_i$ **do**
    $\eta_N = \frac{1}{N_k(c_i,a_t)+1}$;
    if $N_k(c_i, a_t) > 0$ **then**
      $(G^k q^k)(c_i, a_t) = (1 - \eta_N)(G^k q^k)(c_i, a_t) + \eta_N \left(R_t + \gamma \max_{b\in\mathcal{A}}(\Gamma_{\mathrm{NN}} q^k)(Y_{t+1}, b)\right)$;
    else $(G^k q^k)(c_i, a_t) = R_t + \gamma \max_{b\in\mathcal{A}}(\Gamma_{\mathrm{NN}} q^k)(Y_{t+1}, b)$;
    end
    $N_k(c_i, a_t) = N_k(c_i, a_t) + 1$
  end
  if $\min_{(c_i,a)\in\mathcal{Z}_h} N_k(c_i, a) > 0$ **then**
    Foreach $(c_i, a) \in \mathcal{Z}_h$ **do**
      $q^{k+1}(c_i, a) = (1 - \alpha_k) q^k(c_i, a) + \alpha_k (G^k q^k)(c_i, a)$;
    end
    $k = k + 1; \alpha_k = \frac{\beta}{\beta+k}$;
    Foreach $(c_i, a) \in \mathcal{Z}_h$ **do** $N_k(c_i, a) = 0$; **end**
  end
  $t = t + 1$;
until $t \geq T$;
return $\hat{q} = q^k$

每得到一个采样，都会把它暂时更新到 Bellman backup $(Gq)(s_t,a)$ 中存起来（主循环中的第一个 Foreach），注意到 η_N 的设置使得每次更新完毕， $(Gq)(s_t,a)$ 样本上 Bellman backup 的 sample mean。一旦 $\mathcal{Z}_h$ 中的每个元素都被访问过一遍之后，就会做一步更新，让 $q(s_t,a)$ 朝着 $(Gq)(s_t,a)$ 更新一步，更新是 soft 的。

4. Finite sample analysis

以下定理为文章的主要结论，它说明了上述算法要达到足够精度的 optimal Q function 所需要的样本数目。

Theorem 1. *Suppose that Assumptions 1 and 2 hold. With notation* $\beta = 1/(1-\gamma)$ *and* $C = M_r + \gamma V_{\max} M_p$*, for a given* $\varepsilon \in (0, 4V_{\max}\beta)$*, define* $h^* \equiv h^*(\varepsilon) = \frac{\varepsilon}{4\beta C}$*. Let* N_{h^*} *be the* h^**-covering number of the metric space* $(\mathcal{X}, \rho)$*. For a universal constant* $C_0 > 0$*, after at most*

$$T = C_0 \frac{L_{h^*} V_{\max}^3 \beta^4}{\varepsilon^3} \log\left(\frac{2}{\delta}\right) \log\left(\frac{N_{h^*} |\mathcal{A}| V_{\max}^2 \beta^4}{\delta \varepsilon^2}\right)$$

steps, with probability at least $1-\delta$*, we have* $\left\|Q_{h^*}^T - Q^*\right\|_\infty \le \varepsilon$.

- 离散程度：想要达到精度为 ε 的拟合，需要做多精细的离散化呢？文章说明离散程度需要为 h^*，通过公式可以看出：如果要求的精度越高，离散化也需要越精；同时，如果函数空间越不平滑，那么也会要求更高的离散化程度。
- Covering number：在离散状态空间的分析中会有 $|\mathcal{X}|$，在连续状态空间的分析中相应地有 covering number N_{h^*}，它说明大致上需要多少个半径为 h^* 的球能够覆盖整个空间。
- Sample complexity：上述 T 的表达式说的就是 sample complexity。注意到，它和 covering time 呈正比，即如果使用更好的策略去做探索，能够成比例地减小 sample complexity。
- Space complexity：由于本文算法中，每来一个样本，经过处理之后就会被扔掉，因此算法具有较好的 space complexity。基本上只需要存储所有的 $q(s_t,a)$ 和 $(Gq)(s_t,a)$，因此复杂度为 $O(N_{h^*} \times |\mathcal{A}|)$。
- Computational complexity：每一步最坏情况下，需要对每个 $\mathcal{Z}_h$ 中每个元素都做 update，复杂度为 $O(N_{h^*} \times |\mathcal{A}|)$，因此总体复杂度为 $O(T \times N_{h^*} \times |\mathcal{A}|)$。

证明分为三步，第一步分析了一个包含随机性的更新，需要多少步才能够收敛到接近不动点的位置（stochastic approximation）；由于第一步里面对于随机性的更新做了一定的假设，在第二步中需要证明 NNQL 算法满足相应的假设（properties of NNQL）；最后把前两步的结论拼起来即可得到最后的结论。

第一步的结论很有意思，对于一个随机性的更新

$$\theta^{t+1} = \theta^t + \alpha_t \left(F(\theta^t) - \theta^t + w^{t+1} \right), \tag{13}$$

其中随机性体现在噪声 w 项中。如果噪声满足一定的条件，那么经过若干步，它能够近似地收敛到不动点附近。

Theorem 3. *Suppose that the mapping $F : \mathbb{R}^d \to \mathbb{R}^d$ has a unique fixed point θ^* with $\|\theta^*\|_\infty \le V$, and is a γ-contraction with respect to the ℓ_∞ norm in the sense that*

$$\|F(\theta) - F(\theta')\|_\infty \le \gamma \|\theta - \theta'\|_\infty$$

for all $\theta, \theta' \in \mathbb{R}^d$, where $0 < \gamma < 1$. Let $\{\mathcal{F}^t\}$ be an increasing sequence of σ-fields so that α_t and w^t are $\mathcal{F}^t$-measurable random variables, and θ^t be updated as per (13). Let δ_1, δ_2, M, V be non-negative deterministic constants. Suppose that the following hold with probability 1:

1. *The bias $\Delta^{t+1} = \mathbb{E}\left[w^{t+1} \mid \mathcal{F}^t\right]$ satisfies $\left\|\Delta^{t+1}\right\|_\infty \le \delta_1 + \delta_2 \|\theta^t\|_\infty$, for all $t \ge 0$;*
2. *$\left\|w^{t+1} - \Delta^{t+1}\right\|_\infty \le M$, for all $t \ge 0$;*
3. *$\|\theta^t\|_\infty \le V$, for all $t \ge 0$.*

Further, we choose

$$\alpha_t = \frac{\beta}{\beta + t}, \tag{14}$$

where $\beta = \frac{1}{1-\gamma}$. Then for each $0 < \varepsilon < \min\{2V\beta, 2M\beta^2\}$, after

$$T = \frac{48VM^2\beta^4}{\varepsilon^3} \log\left(\frac{32dM^2\beta^4}{\delta\varepsilon^2}\right) + \frac{6V(\beta - 1)}{\varepsilon}$$

iterations of (13), with probability at least $1 - \delta$, we have

$$\left\|\theta^T - \theta^*\right\|_\infty \le \beta(\delta_1 + \delta_2 V) + \varepsilon.$$

（证明分为两部分，先证明 noise 的积累有上界，再证明均值能够收敛到不动点附近，中间需要用到 Azuma-Hoeffding 不等式）

5. Lower bound

文章最后借用 non-parametric regression 所需样本复杂度 lower bound 的结论证明了任意算法的 lower bound。

Theorem 2. *For any reinforcement learning algorithm $\hat{Q}_T$ and any number $\delta \in (0,1)$, there exists an MDP problem and some number $T_\delta > 0$ such that*

$$\Pr\left[\left\|\hat{Q}_T - Q^*\right\|_\infty \geq C\left(\frac{\log T}{T}\right)^{\frac{1}{2+d}}\right] \geq \delta, \qquad \text{for all } T \geq T_\delta,$$

where $C > 0$ is a constant. Consequently, for any reinforcement learning algorithm $\hat{Q}_T$ and any sufficiently small $\varepsilon > 0$, there exists an MDP problem such that in order to achieve

$$\Pr\left[\left\|\hat{Q}_T - Q^*\right\|_\infty < \varepsilon\right] \geq 1 - \delta,$$

one must have

$$T \geq C' d \left(\frac{1}{\varepsilon}\right)^{2+d} \log\left(\frac{1}{\varepsilon}\right),$$

where $C' > 0$ is a constant.

评价

读了一下审稿人意见，大家认为该工作主要的缺点在于算法复杂度还是过高，以至于 impractical。

- 由于不对探索的策略做过多的假设，而探索策略的 covering time 可能很高；
- 有很多实际的 MDP，可能采取任何的行动都达到不了状态空间的某些状态（controllability），这样会产生巨大的 covering time，即 proposition 1&2 中的假设可能不成立；
- 更重要的是它对于整个状态空间都做了离散化，当状态空间维度比较高的时候，需要维护的节点数目会十分大，从而不切实际。

之后可以考虑：

- 探索策略也随着学到的 Q 更新，应该能有更好的 covering time 的上界（instead of proposition 1&2）；
- 对状态空间不做均匀的离散化，而是针对局部不同的 Lipschitz 连续性（考虑定理 1 中的 C）和 visitation frequency 来做不同精度的离散化；
- 不需要对整个 $\mathcal{Z}_h$ 都有样本了才更新，而是局部有数据了就更新，这样可能产生更多的 bias，但是 sample complexity 应该会更好。

发布于 2019-07-15

强化学习 (Reinforcement Learning)

文章被以下专栏收录