

# Provably Efficient Reinforcement Learning with Linear Function Approximation

Chi Jin

University of California, Berkeley  
chijin@cs.berkeley.edu

Zhuoran Yang

Princeton University  
zy6@princeton.edu

Zhaoran Wang

Northwestern University  
zhaoranwang@gmail.com

Michael I. Jordan

University of California, Berkeley  
jordan@cs.berkeley.edu

## 【强化学习 97】Linear MDP



张楚琦

清华大学 交叉信息院博士在读

19 人赞同了该文章

对于线性 MDP 模型，可以做线性函数拟合。

### 原文传送门

Jin, Chi, et al. "Provably efficient reinforcement learning with linear function approximation." arXiv preprint arXiv:1907.05388 (2019).

### 特色

考虑对价值函数做函数拟合（function approximation）。当函数拟合使用的函数 capacity 大的时候，容易遇到 sparsity 的问题，即所遇到的大多数状态的附近都没有其他样本，则很难估计出一个准确的价值函数（variance 大）；当函数拟合使用的函数 capacity 小的时候，容易遇到 misspecification 的问题，即所使用的函数族不足以表示真实的价值函数（bias 大）。为此，文章提出了线性 MDP 模型，在线性 MDP 模型下，线性的函数拟合不会产生 bias，同时也定量分析了如果稍微偏离线性 MDP 模型，线性函数拟合会产生的 bias 大小。值得注意的是，文章提出的线性 MDP 模型能够概括 tabular case。

同时，要设计一个有效的（probably efficient）算法，还需要做探索（或称作 exploration-exploitation tradeoff）。探索问题可以想象成如何有效地估计到 transition function / matrix。

- 有些算法假设能有一个 simulator，这样能够设置成任意的状态（比如，最简单的那种 tabular Q-learning），这等于是能够直接把你带到 transition matrix 的任意行中。
- 有些算法假设稍微不那么强，它们假设有一个 restart distribution（比如专栏里面讲到的 Kakade 02 和 19 年的两篇工作），这样能够从一个在状态空间中分布比较均匀的初始分布出发来采样，这等于是从一定程度上保证了能比较均匀地访问到 transition matrix 的所有行。
- 还有些算法假设了比较简单的 transition function，比如 transition function 是（比较）确定性的或者自由度较低的（本文引用的若干文献），这等于是对 transition function 加了一些先验，从而用更少的样本能够估计到 transition function。
- 本文使用 UCB 方法来解决探索问题，在初始状态甚至可以是对手故意选取的情况下，能够达到  $\tilde{O}(\sqrt{dH^2T})$ 。其中 d 是线性化特征的维度（后面会详讲）

- 为什么在这种情形下，**regret**会有一个上界呢？可以这么理解，如果对手挑选了一个从来没有见过的状态，我没什么信息都没有，容易产生一个较大的 regret，但通过遭受这个 regret 我们获取了更多的信息；最多在所有的状态上都通过遭受 regret 来学习，因此总是有一个 regret 的上限的。
- 为什么 **regret** 上界和状态空间的大小没关系，而只和  $d$  有关呢？因为这里不是直接对 transition matrix 估计，而是估计它朝  $d$  维的『基底』上的投影；只要对每一维度上的投影估计准确即能得到最优策略。

## 过程

### 一、Linear MDP Model

**Assumption A** (Linear MDP [12, 29]). MDP( $\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r$ ) is a *linear MDP* with a feature map  $\phi: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$ , if for any  $h \in [H]$ , there exist  $d$  unknown (signed) measures  $\mu_h = (\mu_h^{(1)}, \dots, \mu_h^{(d)})$  over  $\mathcal{S}$  and an unknown vector  $\theta_h \in \mathbb{R}^d$ , such that for any  $(x, a) \in \mathcal{S} \times \mathcal{A}$ , we have

$$\mathbb{P}_h(\cdot | x, a) = \langle \phi(x, a), \mu_h(\cdot) \rangle, \quad r_h(x, a) = \langle \phi(x, a), \theta_h \rangle. \quad (3)$$

Without loss of generality, we assume  $\|\phi(x, a)\| \leq 1$  for all  $(x, a) \in \mathcal{S} \times \mathcal{A}$ , and  $\max_{i \in [d]} \|\mu_h^{(i)}\| \leq \sqrt{d}$  for all  $h \in [H]$ .

需要注意几件事情：

- 虽然特征向量只有  $d$  维，但是 transition function 中  $\mu_h(\cdot)$  可以是任意在状态空间上的函数，因此在该模型下 transition function 的自由度仍然为无穷维；显然，reward function 的自由度是  $d$  维。
- Linear MDP 看起来做了比较强的假设，要求这些过程都是线性的。但是只要  $d$  足够大，该模型可以完全概括 tabular MDP 的情形。参见下图的 Example 2.1。
- 要使得 transition function 是一个合法的函数，在特征向量  $\phi(x, a)$  给定的情况下， $\mu_h(\cdot)$  的选取需要有一定的限制。下图的 Example 2.2 给出了一个合法的例子。下图的 Proposition A.1. 给出了具体的限制条件。
- 最后这件事情最为重要：**Linear MDP 和 linear function approximation 之间具有必然联系！** 具体来说，
  - 如果模型为线性的，那么任意策略的价值函数就可以用线性函数拟合来准确表示（Proposition 2.3.）；
  - 如果价值函数用线性函数拟合来表示，要希望任意策略的 Bellman 算子作用到某个线性价值函数上之后还是线性价值函数，那么 MDP 模型必须为线性（Proposition 5.1.）。

**Example 2.1** (Tabular MDP). For the scenario with finitely many states and actions, letting  $d = |\mathcal{S}| \times |\mathcal{A}|$ , then each coordinate can be indexed by state-action pair  $(x, a) \in \mathcal{S} \times \mathcal{A}$ . Let  $\phi(x, a) = \mathbf{e}_{(x,a)}$  be the canonical basis in  $\mathbb{R}^d$ . Then if we set  $\mathbf{e}_{(x,a)}^\top \mu_h(\cdot) = \mathbb{P}_h(\cdot | x, a)$  and  $\mathbf{e}_{(x,a)}^\top \theta_h = r_h(x, a)$  for any  $h \in [H]$ , we recover the tabular MDP.

**Example 2.2** (Simplex Feature Space). When the feature space,  $\{\phi(x, a): (x, a) \in \mathcal{S} \times \mathcal{A}\}$ , is a subset of the  $d$ -dimensional simplex,  $\{\psi | \sum_{i=1}^d \psi_i = 1 \text{ and } \psi_i \geq 0 \text{ for all } i\}$ , a linear MDP can be instantiated by choosing  $\mathbf{e}_i^\top \mu_h$  to be an arbitrary probability measure over  $\mathcal{S}$  and letting  $\theta_h$  be any vector such that  $\|\theta_h\|_\infty \leq 1$ .

**Proposition A.1.** For a linear MDP, for any  $(x, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$ , we have

$$\phi(x, a)^\top \mu_h(\mathcal{S}) = 1, \quad \phi(x, a)^\top \mu_h(\mathcal{B}) \geq 0, \quad \forall \text{ measurable } \mathcal{B} \subseteq \mathcal{S}. \quad (12)$$

**Proposition 2.3.** For a linear MDP, for any policy  $\pi$ , there exist weights  $\{\mathbf{w}_h^\pi\}_{h \in [H]}$  such that for any  $(x, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$ , we have  $Q_h^\pi(x, a) = \langle \phi(x, a), \mathbf{w}_h^\pi \rangle$ .

*Proof.* The linearity of the action-value functions directly follows from the Bellman equation in Eq. (1):

$$Q_h^\pi(x, a) = r(x, a) + (\mathbb{P}_h V_{h+1}^\pi)(x, a) = \langle \phi(x, a), \boldsymbol{\theta}_h \rangle + \int_{\mathcal{S}} V_{h+1}^\pi(x') \cdot \langle \phi(x, a), d\mu_h(x') \rangle.$$

Therefore, we have  $Q_h^\pi(x, a) = \langle \phi(x, a), \mathbf{w}_h^\pi \rangle$  where  $\mathbf{w}_h^\pi$  is given by  $\mathbf{w}_h^\pi = \boldsymbol{\theta}_h + \int_{\mathcal{S}} V_{h+1}^\pi(x') d\mu_h(x')$ .

**Proposition 5.1.** Let  $\mathcal{Q} = \{Q | Q(\cdot, \cdot) = \phi(\cdot, \cdot)^\top \mathbf{w}, \mathbf{w} \in \mathbb{R}^d\}$  be the family of linear action-value functions. Suppose that  $\mathcal{S}$  is a finite set, and for any  $x \in \mathcal{S}$ , there exist two actions  $a, \bar{a} \in \mathcal{A}$  such that  $\phi(x, a) \neq \phi(x, \bar{a})$ . Then,  $\mathbb{T}_h^\pi \mathcal{Q} \subset \mathcal{Q}$  for all  $\pi$  only if the Markov transition measures  $\mathbb{P}_h$  are linear in  $\phi$ .

## 二、算法

算法使用 Least-Square Value Iteration (LSVI) + Upper-Confidence Bound (UCB)。每一轮迭代分为两个步骤：

- 第一个步骤是基于历史上所有的样本来估计 Q 函数的参数，该 Q 函数是增加了 UCB bonus 的 Q 函数置信上界；
- 第二个步骤是基于估计到的 Q 函数，直接采用 argmax 来做 rollout（注意到在标准的 Q-learning 中，这一步需要加  $\epsilon$ -greedy 来做探索，但是这里由于使用了更有效的 UCB 因此代替了  $\epsilon$ -greedy）。

先给出算法框图，然后再详细讲解。

---

### Algorithm 1 Least-Squares Value Iteration with UCB (LSVI-UCB)

---

<pre> 1: for episode <math>k = 1, \dots, K</math> do 2:   Receive the initial state <math>x_1^k</math>. 3:   for step <math>h = H, \dots, 1</math> do 4:     <math>\Lambda_h \leftarrow \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) \phi(x_h^\tau, a_h^\tau)^\top + \lambda \cdot \mathbf{I}</math>. 5:     <math>\mathbf{w}_h \leftarrow \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) [r_h(x_h^\tau, a_h^\tau) + \max_a Q_{h+1}(x_{h+1}^\tau, a)]</math>. 6:     <math>Q_h(\cdot, \cdot) \leftarrow \min\{\mathbf{w}_h^\top \phi(\cdot, \cdot) + \beta[\phi(\cdot, \cdot)^\top \Lambda_h^{-1} \phi(\cdot, \cdot)]^{1/2}, H\}</math>. 7:   for step <math>h = 1, \dots, H</math> do 8:     Take action <math>a_h^k \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} Q_h(x_h^k, a)</math>, and observe <math>x_{h+1}^k</math>. </pre>	<p>Initialize:</p> <p><math>\Lambda_h \leftarrow \lambda \mathbf{I}</math></p> <p><math>\mathbf{w}_h \leftarrow \mathbf{0}</math></p>
--	---

---

知乎 @张楚珩

## Least-Square Value Iteration (LSVI)

假设 transition function 已知，那么要求解 optimal Q function，只需要依照如下更新方式按  $h = H, \dots, 1$  撸一遍就好了（动态规划）。

$$Q_h^*(x, a) \leftarrow [r_h + \mathbb{P}_h \max_{a' \in \mathcal{A}} Q_{h+1}^*(\cdot, a')](x, a), \quad \forall (x, a) \in \mathcal{S} \times \mathcal{A}.$$

但是实际上 transition function 是不知道的，因此只能在估计的样本上最小化上式左边和右边之间的 MSE；同时，当状态空间、动作空间很大时，很难对于每一个  $(\mathbf{s}, \mathbf{a})$  pair 都有数据，因此把它们往  $d$  维空间上做一个投影  $\phi(\mathbf{s}, \mathbf{a})$ ，并限定线性模型。得到以下优化问题

$$\mathbf{w}_h \leftarrow \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \sum_{\tau=1}^{k-1} [r_h(x_h^\tau, a_h^\tau) + \max_{a \in \mathcal{A}} Q_{h+1}(x_{h+1}^\tau, a) - \mathbf{w}^\top \phi(x_h^\tau, a_h^\tau)]^2 + \lambda \|\mathbf{w}\|^2.$$

其中，L2 正则项可以避免 over-fitting（我记得至少 PRML 书里面讲过其原理）。对上述待优化函数写下来，求导，并且令导数为零，可以直接写出闭式解（见算法框图的第 4-5 行）。

## Upper-Confidence Bound (UCB)

首先， $m(\mathbf{s}, \mathbf{a}) := (\phi^\top(\mathbf{s}, \mathbf{a}) \Delta_h^{-1} \phi(\mathbf{s}, \mathbf{a}))^{-1}$ （参考算法框图第 6 行）表示了  $(\mathbf{s}, \mathbf{a})$  被等效地访问了多少次。考虑 Example 2.1. 中的 tabular MDP 的情形。 $\Delta_h$  是一个  $S \times S \times A$  的对角矩阵，对角元上的元素为该  $(\mathbf{s}, \mathbf{a})$  pair 在历史数据中被访问了多少次（为了防止某些  $(\mathbf{s}, \mathbf{a})$  pair 一次都没有访问到，会不能取逆，因此加了正则）。而接下来的操作则提取出来具体的某个  $(\mathbf{s}, \mathbf{a})$  访问的次数。在非 tabular MDP 的情形下，该计数等于是把历史数据的特征向量朝  $\phi(\mathbf{s}, \mathbf{a})$  做投影并且计数。

最后，考虑到 UCB bonus 的公式为  $\beta/\sqrt{m}$ ，由此可以写出算法框图第 6 行的公式。

## 三、理论结果

如果实际 MDP 满足前述 Linear MDP 的要求，则有如下 regret bound。

**Theorem 3.1.** Under Assumption A, there exists an absolute constant  $c > 0$  such that, for any fixed  $p \in (0, 1)$ , if we set  $\lambda = 1$  and  $\beta = c \cdot dH\sqrt{\epsilon}$  in Algorithm 1 with  $\epsilon := \log(2dT/p)$ , then with probability  $1 - p$ , the total regret of LSVI-UCB (Algorithm 1) is at most  $\mathcal{O}(\sqrt{d^3 H^3 T \epsilon^2})$ , where  $\mathcal{O}(\cdot)$  hides only absolute constants.

- 该 regret bound 可以被转化为 PAC 的结果：考虑固定初始状态  $\mathbf{s}_1$ （而不再是每轮对手选定），根据 regret 的定义有  $\sum_{k=1}^K (V^*(\mathbf{s}_1) - V^{a^k}(\mathbf{s}_1)) = \mathcal{O}(\sqrt{d^3 H^3 T}) = \mathcal{O}(\sqrt{d^3 H^4 K})$ ，取  $K = \mathcal{O}(d^3 H^4 / \epsilon^2)$ ，并且随机取一个  $k \in [K]$ ，则有  $K(V^*(\mathbf{s}_1) - V^{a^k}(\mathbf{s}_1)) \approx \sum_{k=1}^K (V^*(\mathbf{s}_1) - V^{a^k}(\mathbf{s}_1)) = \mathcal{O}(d^3 H^4 / \epsilon)$ ，其中约等号表示随机选取可以大概率取到低于均值的情形，在这种情况下有  $V^*(\mathbf{s}_1) - V^{a^k}(\mathbf{s}_1) \approx \epsilon$ ，即需要  $KH = \mathcal{O}(d^3 H^4 / \epsilon^2)$  的样本以拿到一个  $\epsilon$ -optimal 的策略。（跟文中算出来的差了一个  $H$ 。）
- 空间复杂度为  $\mathcal{O}(d^2 H + dAT)$ ：储存  $\Delta_h$  需要  $d^2 H$ ，储存  $\mathbf{w}_h$  需要  $dH$ ，储存所有的 reward 需要  $T$ ，储存所有遇到的  $\phi(\mathbf{s}, \mathbf{a})$  pair 以及所遇到状态下所有其他行动的  $\phi(\mathbf{s}, \mathbf{a}')$ ，它们需要  $dAT$ 。
- 计算复杂度为  $\mathcal{O}(d^2 AKT)$ ：有点没算出来，跟文中的差了一个  $d$ 。

如果实际 MDP 和前述 Linear MDP 的假设有一定的差距（misspecification），差距定义为

**Assumption B** ( $\zeta$ -Approximate Linear MDP). For any  $\zeta \leq 1$ , we say that  $\text{MDP}(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$  is a  $\zeta$ -approximate linear MDP with a feature map  $\phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$ , if for any  $h \in [H]$ , there exist  $d$  unknown (signed) measures  $\mu_h = (\mu_h^{(1)}, \dots, \mu_h^{(d)})$  over  $\mathcal{S}$  and an unknown vector  $\theta_h \in \mathbb{R}^d$  such that for any  $(x, a) \in \mathcal{S} \times \mathcal{A}$ , we have

$$\|\mathbb{P}_h(\cdot | x, a) - \langle \phi(x, a), \mu_h(\cdot) \rangle\|_{\text{TV}} \leq \zeta, \quad |r_h(x, a) - \langle \phi(x, a), \theta_h \rangle| \leq \zeta. \quad (4)$$

Without loss of generality, we assume that  $\|\phi(x, a)\| \leq 1$  for all  $(x, a) \in \mathcal{S} \times \mathcal{A}$ , and  $\max_{h \in [H]} \|\theta_h\| \leq \sqrt{d}$  for all  $h \in [H]$ .

则有如下 regret bound。

**Theorem 3.2.** Under Assumption B, there exists an absolute constant  $c > 0$  such that, for any fixed  $p \in (0, 1)$ , if we set  $\lambda = 1$  and  $\beta_k = c \cdot (d\sqrt{\iota} + \zeta\sqrt{kd})H$  in Algorithm 1 with  $\iota := \log(2dT/p)$ , then with probability  $1 - p$ , the total regret of LSVI-UCB (Algorithm 1) is at most  $\mathcal{O}(\sqrt{d^3 H^3 T \iota^2} + \zeta d H T \sqrt{\iota})$ .

不难理解，但存在 misspecification 的时候，最后学习到的模型一定存在 bias，因此 regret 中肯定会有关于  $T$  线性的项。

#### 四、算法机制

算法中有两点最为关键

- 探索的实质就是说明要如何估计 transition function，因此这里证明的重点之一就是说明 LSVI 是如何估计到 transition function 的。
- UCB 的作用机理可以概括为：加上 UCB bonus 之后，估计的价值函数 > 最优价值函数；通过采样样本的增多，估计的价值函数  $\rightarrow$  实际的策略性能；而最优价值函数 > 实际的策略性能。最后的结果是实际的策略性能最后能够逼近最优价值函数。需要注意的一点是由于价值函数的更新是 bootstrapped，因此分析上也是后一层的误差会传递到前一层。
  - 关于 UCB 的机理，可以具体参考该作者前面的一篇工作 [UCB+Q-learning](#)。注意到前面的算法是 online 的，因此 Q 值的更新设置了一个 learning rate  $\alpha_n$ 。这里比较粗暴，每次都对于所有的样本一锅端，直接计算 optimal Bellman operator 作用之后的闭式解。

这里主要来讲第一点。即，算法中使用样本估计 transition function 的方法能够逼近一个完美的 optimal Bellman operation。为了简化分析，可以 1) 扔掉 UCB 项；2) 扔掉 smoothing 的项（即  $\lambda=0$ ）。在第二个简化下，有

$$\Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) \phi(x_h^\tau, a_h^\tau)^\top \approx \mathbf{I}.$$

一个 optimal Bellman operation，对于所有的  $(x, a)$  的 Q 值都做如下更新：

$$Q_h^*(x, a) \leftarrow [r_h + \mathbb{P}_h \max_{a' \in \mathcal{A}} Q_{h+1}^*(\cdot, a')](x, a), \quad \forall (x, a) \in \mathcal{S} \times \mathcal{A}.$$

算法中的基于样本的更新如下，即对于  $w_h$  的所有维度都做如下更新：

$$Q_h(x, a) \approx \phi(x, a)^\top w_h = \phi(x, a)^\top \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) [r_h(x_h^\tau, a_h^\tau) + V_{h+1}(x_{h+1}^\tau)],$$

其中记  $V(s_{h+1}) = \max_a Q(s_{h+1}, a)$ 。

## 1. Reward Function Term

根据 Linear MDP 的假设，奖励函数是确定性的，因此很容易得到

$$\begin{aligned} & \phi(s, a)^T \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(s_h^\tau, a_h^\tau) r(s_h^\tau, a_h^\tau) \\ &= \phi(s, a)^T \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(s_h^\tau, a_h^\tau) \phi(s_h^\tau, a_h^\tau)^T \theta_h \\ &= \phi(s, a)^T \theta_h \\ &= r(s, a) \end{aligned}$$

## 2. Transition Function Term

定义以下两个算子

$$\hat{\mathbb{P}}_h(\cdot|x, a) := \phi(x, a)^T \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) \delta(\cdot, x_{h+1}^\tau),$$

$$\bar{\mathbb{P}}_h(\cdot|x, a) := \phi(x, a)^T \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) \mathbb{P}_h(\cdot|x_h^\tau, a_h^\tau). \quad (5)$$

不难看出，算法中的 transition function term 为  $\hat{\mathbb{P}}_h V_{h+1}$ ，而在线性模型下  $\hat{\mathbb{P}}_h V_{h+1} = \mathbb{P}_h V_{h+1}$ ，而  $\mathbb{P}_h V_{h+1}$  就是 optimal Bellman operation 中对应的项。这一点容易看出：

$$\bar{\mathbb{P}}_h(\cdot|x, a) = \phi(x, a)^T \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) \phi(x_h^\tau, a_h^\tau)^T \mu_h(\cdot) \approx \phi(x, a)^T \mu_h(\cdot) = \mathbb{P}_h(\cdot|x, a).$$

下面的重点就是

**Prove  $\hat{\mathbb{P}}_h V_{h+1}(x, a) \approx \bar{\mathbb{P}}_h V_{h+1}(x, a)$  via Value-Aware Uniform Concentration.**

注意到

$$(\hat{\mathbb{P}}_h - \bar{\mathbb{P}}_h) V_{h+1}(x, a) = \phi(x, a)^T \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau) [V_{h+1}(x_{h+1}^\tau) - \mathbb{P}_h V_{h+1}(x_h^\tau, a_h^\tau)].$$

由于  $s_{h+1}^\tau \sim \mathbb{P}_h(s_{h+1}^\tau, a_h^\tau)$ ，因此可以使用 concentration inequality 来 bound。但是这里面有一些难处理的地方，这就是  $V_{h+1}$  每一轮是在变化的，但是这个变化的过程很难去 track，因此文章直接在如下函数族上做 concentration。

$$\mathcal{V} = \left\{ V(\cdot) | V(\cdot) = \min_{a \in \mathcal{A}} \left\{ \max_{a \in \mathcal{A}} \phi(\cdot, a)^T \mathbf{w} + \beta \sqrt{\phi(\cdot, a)^T \Lambda^{-1} \phi(\cdot, a)} \right\}, \mathbf{w} \in \mathbb{R}^d, \beta \in \mathbb{R}, \Lambda \in \mathbb{R}^{d \times d} \right\}, \quad (6)$$

这一个部分给我们的启示是，虽然要准确估计 transition function 很难，但是通过把它往线性基底上面投影，可以使用更少的样本来对其有准确估计。

UCB 的探索奖励是 Hoeffding type，如果改用 Bernstein type 的 regret 可能可以去掉一个  $\frac{1}{\epsilon}$ 。

机器学习

强化学习 (Reinforcement Learning)

深度学习 (Deep Learning)

▲ 赞同 19



💬 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