Non-delusional Q-learning and value iteration

Tyler Lu
Google AI
tylerlu@google.com

Dale Schuurmans
Google AI
schuurmans@google.com

Craig Boutilier
Google AI
cboutilier@google.com

【强化学习算法 37】PCVI/PCQL

张楚珩
清华大学 交叉信息院博士在读

PCVI/PCQL分别是Policy-Class Value Iteration和Policy-class Q-learning的简称，是最新出炉的NIPS 2018 best paper。

原文传送门

Lu, Tyler, Dale Schuurmans, and Craig Boutilier. "Non-delusional Q-learning and value-iteration." Advances in Neural Information Processing Systems. 2018.

特色

这篇NIPS best paper提出了value-based强化学习算法里面都普遍存在的妄想偏差（delusional bias），并且说明了妄想偏差不仅仅可能导致求得的策略不是最优，而且可能导致算法发散（divergence）、算法振荡（cyclic behaviour）以及使用不同dicount rate训练效果不同的现象（discounting paradox）。为了完全避免妄想偏差，文章中分别提出了model-based的算法PCVI和model-free的算法PCQL，这两个算法只能针对tabular case并且复杂度较高。为了得到更合实际的算法，文章也提出了几种可能的途径。

过程

- Delusional bias
- Impact of delusional bias
  - Divergence
  - Cyclic behavior
  - Discounting paradox
- Algorithm to fully resolve the delusion problem
  - PCVI
  - PCQL
- Towards practical algorithms

1. **妄想偏差如何产生？**

这里提出的妄想偏差，个人认为其实解释了Sutton书里面提到的the deadly triad。

11.3 The Deadly Triad

Our discussion so far can be summarized by saying that the danger of instability and divergence arises whenever we combine all of the following three elements, making up what we call *the deadly triad*:

Function approximation A powerful, scalable way of generalizing from a state space much larger than the memory and computational resources (e.g., linear function approximation or artificial neural networks).

Bootstrapping Update targets that include existing estimates (as in dynamic programming or TD methods) rather than relying exclusively on actual rewards and complete returns (as in MC methods).

Off-policy training Training on a distribution of transitions other than that produced by the target policy. Sweeping through the state space and updating all states uniformly, as in dynamic programming, does not respect the target policy and is an example of off-policy training.

考虑Q-learning的更新公式

$$\theta \leftarrow \theta + \alpha\Big(r + \gamma \max_{a' \in A} Q_\theta(s', a') - Q_\theta(s, a)\Big)\nabla_\theta Q_\theta(s, a).$$

注意到在function approximation下，价值函数的表示能力是有限的

$$\mathcal{F} = \{f_\theta : S \times A \to \mathbb{R} | \theta \in \Theta\}$$

与之对应的greedy策略的表示能力也是有限的

$$G(\Theta) = \Big\{\pi_\theta \;\Big|\; \pi_\theta(s) = \operatorname*{argmax}_{a \in A} f_\theta(s, a), \theta \in \Theta\Big\}.$$

妄想偏差在考虑某一个state-action对的时候会基于”下面状态的所有行动都是可以取到“的假设的，当前后相互冲突的时候就会产生妄想偏差。

下面通过一个实际的例子来展示妄想偏差。上面显示所构造的MDP，左下角显示的是state-action对的特征抽取，右下角显示了不考虑linear function approximation情况下的最优策略，和考虑该限制下的最优策略。

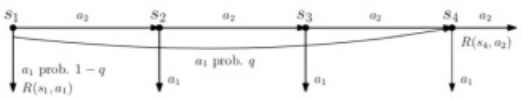
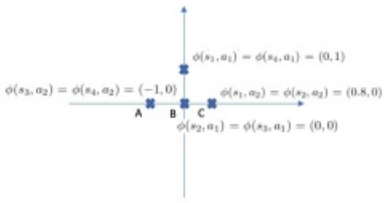


Figure 1: A simple MDP that illustrates delusional bias (see text for details).

$q = 0.1$
$R(s_1, a_1) = 0.3$
$R(s_4, a_2) = 2$
$R(s, a) = 0$ for other s and a
Linear function approximator
$Q_\theta(\phi(s,a)) = \theta_1\phi_1 + \theta_2\phi_2$



- Unconstrained optimal policy
  - Take a_2 in every state, $V(s_1) = 2$
- Constraint from linear function approximator
  - Monotonicity of A, B, C
  - If $\theta_1 < 0$, A>B>C, $\pi(s_3) = a_2, \pi(s_2) = a_1$
  - If $\theta_1 > 0$, A<B<C, $\pi(s_3) = a_1, \pi(s_2) = a_2$
  - $\pi(s_3) = a_2, \pi(s_2) = a_2$ is unreachable
- Constrained optimal policy
  - Take a_1 on s_1 and take a_2 on s_4, $V(s_1) = 0.5$

通过计算可以发现，Q-learning不能够收敛到这个限制下的最优策略，而是收敛到一个更差的策略。

- Consider Q-learning with ϵ-greedy exploration
- Conclusion: Q-learning does not converge to the optimal policy but to a second best policy
- Proof sketch:
  - Assume optimal policy is a fixed point
  - Solve for θ by condition of fixed points
  - Whether the solution contradict the assumption

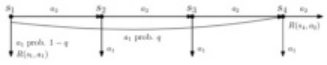


Figure 1: A simple MDP that illustrates delusional bias (see text for details).

- Constrained optimal policy
  - Take a_1 on s_1 and take a_2 on s_4, $V(s_1) = 0.5$
- The second best policy
  - Take a_1 on s_1 and take a_1 on s_4, $V(s_1) = 0.03$

2. 妄想偏差产生的影响

妄想偏差可能导致算法发散

Delusional Bias Causes Divergence

- Consider $S = \{s_1, s_2\}$ and A= $\{a_1, a_2\}$.
- Reward: zero everywhere
- State-action representation

  $\phi(s_1, a_1) = (1+\eta)/Z$; $\phi(s_1, a_2) = 1/Z$; $\phi(s_2, a_1) = 1/Z$; and $\phi(s_2, a_2) = 3/Z$, $\quad Z = \sqrt{12 + 2\eta + \eta^2}$ for $\eta > 0$
- Linear function approximation $Q(s, a) = \boldsymbol{\phi}(s, a)\theta$
- Dynamics: a_1 leads to s_1, a_2 leads to s_2
- For $\theta > 0$, the greedy policy stays where it is; for $\theta < 0$, the greedy policy always chooses to switch
- Conclusion: Starting at $\theta_0 = 1$, θ grows positively without bound (Q-learning with ϵ-greedy)

- Proof.
  - Visitation frequency: $\mu(s_1, a_1) = (1-\varepsilon)/2$; $\mu(s_1, a_2) = \varepsilon/2$; $\mu(s_2, a_1) = \varepsilon/2$; and $\mu(s_2, a_2) = (1-\varepsilon)/2$.
  - Expected update: When $\gamma > (5 - 4\epsilon)/(5 - 3\epsilon)$, the update is positive

    $$\mathbb{E}[\Delta\theta] = \sum_{s,a} \mu(s,a)\phi(s,a)\delta(s,a), \quad \mathbb{E}[\Delta\theta] = \left(\frac{(5-3\varepsilon)\theta}{Z}\right)\gamma - \left(\frac{(5-4\varepsilon)\theta}{Z}\right),$$
  - Divergence w.p. 1

    $$\begin{aligned} \mathbb{E}[\theta_k] &= \mathbb{E}[\theta_{k-1} + \alpha\,\mathbb{E}[\Delta\theta_{k-1}]] \\ &= \mathbb{E}[\theta_{k-1} + \alpha\theta_{k-1}\,\mathbb{E}[\Delta\theta_0]] \\ &= (1 + \alpha\,\mathbb{E}[\Delta\theta_0])\,\mathbb{E}[\theta_{k-1}], \end{aligned}$$

- What's wrong?

  $$\delta(s,a) = \gamma \sum_{s'} p(s'|s,a) \max_{a'} \phi(s',a')\theta - \phi(s,a)\theta \qquad \mathbb{E}[\Delta\theta] = \sum_{s,a} \mu(s,a)\phi(s,a)\delta(s,a),$$

  $$\delta(s_1, a_1) = -(1-\gamma)\theta/Z, \quad \delta(s_1, a_2) = (3\gamma - 1)\theta/Z, \quad \delta(s_2, a_1) = -(1-\gamma)\theta/Z; \text{ and } \delta(s_2, a_2) = -3(1-\gamma)\theta/Z,$$
  - State (s_1, a_2) is *deluded* by the wrong value from $\max_{a'} Q(s_2, a')$, leading to a increase of θ
  - Unfortunately magnitude of update incurred by (s_1, a_2) overwhelms the others

妄想偏差会产生无尽的振荡

Delusional Bias Causes Cyclic Behaviour

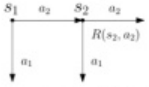
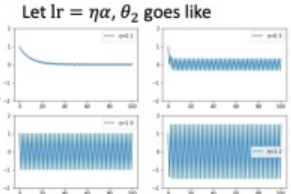


Figure 4: This two state MDP has $\phi(s_1, a_2) = (0, 1/\sqrt{\alpha})$, $\phi(s_2, a_2) = (0, -1/\sqrt{\alpha})$, $\phi(s_1, a_1) = \phi(s_2, a_1) = (0, 0)$. There is a single non-zero reward $R(s_2, a_2) = 1/\sqrt{\alpha}$. All transitions are deterministic and only one is non-terminal: $s_1, a_2 \to s_2$.

Let $\mathrm{lr} = \eta\alpha$, θ_2 goes like



- Backup at (s_1, a_1) and (s_2, a_1) dose not change weight
- Backup at (s_2, a_2), assuming $\theta^{(k-1)} = (b, c)$

  $$\theta^{(k)} = (b,c) + \alpha \left(0, -\frac{1}{\sqrt{\alpha}}\right) \cdot \left(\frac{1}{\sqrt{\alpha}} + \gamma \cdot 0 - \left(-\frac{c}{\sqrt{\alpha}}\right)\right) = (b, -1)$$
- Next, backup at (s_2, a_2), (s_1, a_1) and (s_2, a_1) dose not change weight
- Backup at (s_1, a_2), assuming $\theta^{(k-1)} = (b, -1)$

  $$\theta^{(k')} = (b,-1) + \alpha \left(0, \frac{1}{\sqrt{\alpha}}\right) \cdot \left(0 + \gamma \cdot \frac{1}{\sqrt{\alpha}} - \left(-\frac{1}{\sqrt{\alpha}}\right)\right) = (b, 1)\,.$$

- Conclusion: this causes cyclic behavior as long as learning rate has a lower bound

妄想偏差会导致即使测试的时候使用某个discount rate，但是训练的时候可能使用不同的discount rate会产生更好的效果（discounting paradox）

Delusional Bias Causes Discounting Paradox

- Policy evaluated using $\gamma = 1$
- Linear approximator $Q_\theta(s,a) = \theta_1\phi(s) + \theta_2\phi(a) + \theta_3$,
- Feature embedding
  $\phi(s_1) = 2$; $\phi(s_2) = 1$; $\phi(s_2') = 0$; $\phi(a_1) = -1$; and $\phi(a_2) = 1$.
- Dynamics:
  - when $\theta_2 > 0$ always a_2, policy π_{a_2}
  - when $\theta_2 < 0$ always a_1, policy π_{a_1}
- Optimal policy is π_{a_2}, since
  $V^{\pi_{a_2}}_{\gamma=1} = 2 > 2 - \delta = V^{\pi_{a_1}}_{\gamma=1}$

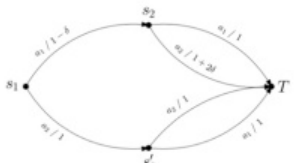


Figure 5: A deterministic MDP starting at state s_1 and terminating at T, $Q_\theta(T,a) = 0$. Directed edges are transitions of the form a/r where a is the action taken and r the reward. Parameter $\delta > 0$.

- Consider train with different discounts

$$\hat{\theta}^{\gamma=0} = \text{QL}(\gamma = 0, \theta^{(0)}, \varepsilon\text{Greedy}), \qquad \hat{\pi}_0 = \text{Greedy}(Q_{\hat{\theta}^{\gamma=0}}),$$
$$\hat{\theta}^{\gamma=1} = \text{QL}(\gamma = 1, \theta'^{(0)}, \varepsilon\text{Greedy}), \qquad \hat{\pi}_1 = \text{Greedy}(Q_{\hat{\theta}^{\gamma=1}}).$$

- The myopic setting ($\gamma = 0$) finds the optimal policy, whereas non-myopic setting ($\gamma = 1$) does not.
- Key: non-myopic setting considers (s_1, a_1) $r = 1 - \delta$ followed by (s_2, a_2) $r = 1 + 2\delta$ which cannot be realized

3. 构造完全消除妄想偏差的算法

其主要思想就是维护一个信息集（information set），对于每一个Q值，都记录下来得到这个Q值对于策略参数又怎样的限制，然后每次backup都在这些信息集上分别运算。先来看一些定义。

- Main idea: avoids delusion by using *information sets* to track the "dependencies" contained in all Q-values
- Information set $X \subseteq \Theta$, each Q-value is associated with a information set (X, q)
- Finite partition of $\mathcal{X}$: set of non-empty subsets $P = \{X_1, \cdots, X_k\}$ such that $X_1 \cup \cdots \cup X_k = \mathcal{X}$ and $X_i \cap X_j = \emptyset$; call any $X_i \in P$ a cell; denote the set of all finite partitions $\mathcal{P}(\mathcal{X})$
- Refinement: a partition P' is a refinement of P if for all $X' \in P'$ there exists a $X \in P$ such that $X' \subseteq X$
- Function of partition / set of all functions of partitions / refinement of function of partitions
- Intersection sum

Definition 5. *Let $P \in \mathcal{P}(\mathcal{X})$. A* mapping $h : P \to \mathbb{R}$ *is called a* function of partition P. *Let $\mathcal{H} = \{h : P \to \mathbb{R} \mid P \in \mathcal{P}(\mathcal{X})\}$ be the set of all such functions of partitions. Let $h_1, h_2 \in \mathcal{H}$, an* intersection sum *is a binary operator $h = h_1 \oplus h_2$ defined by*

$$h(X_1 \cap X_2) = h_1(X_1) + h_2(X_2), \quad \forall X_1 \in \mathrm{dom}(h_1), X_2 \in \mathrm{dom}(h_2), X_1 \cap X_2 \neq \emptyset$$

where $\mathrm{dom}(\cdot)$ is the domain of a function (in this case a partition of $\mathcal{X}$). We say h_1 is a refinement *of h_2 if partition $\mathrm{dom}(h_1)$ is a refinement of $\mathrm{dom}(h_2)$.*

- Information set constrained by a specified step of greedy policy

$$[s \mapsto a] = \{\theta \in \Theta \mid \pi_\theta(s) = a\}$$

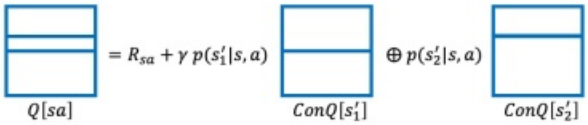
- Assume access to an oracle Witness: given a set of state-action constraints $\{(s, a)\} \subseteq S \times A$, whether there exists a $\pi_\theta(s) = 1$ for all pairs; if there exists, return a such θ

4. 一个model-based的算法（PCVI）

PCVI

- $dom(Q[sa]) \in \mathcal{P}(\Theta)$
- $dom(ConQ[sa]) \in \mathcal{P}([s \mapsto a])$
- $dom(ConQ[s]) \in \mathcal{P}(\Theta)$

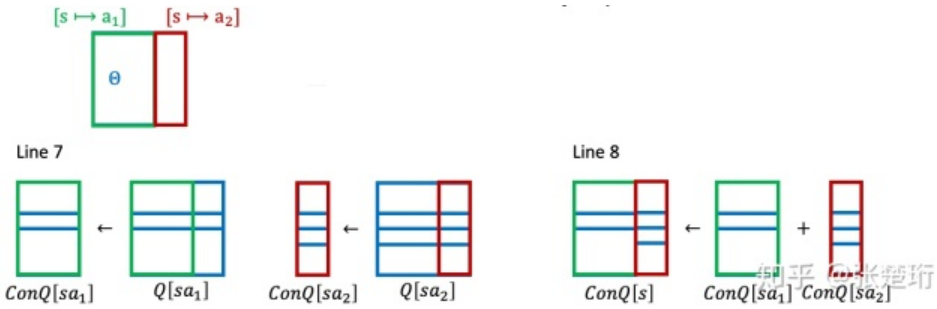
Line 6



Algorithm 1 Policy-Class Value Iteration (PCVI)

Input: $S, A, p(s' \mid s, a), R, \gamma, \Theta$, initial state s_0

1: $\mathtt{Q}[sa] \leftarrow$ initialize to mapping $\Theta \mapsto 0$ for all s, a
2: $\mathtt{ConQ}[sa] \leftarrow$ initialize to mapping $[s \mapsto a] \mapsto 0$ for all s, a
3: Update $\mathtt{ConQ}[s]$ for all s (i.e., combine all table entries in $\mathtt{ConQ}[sa_1], \ldots, \mathtt{ConQ}[sa_m]$)
4: **repeat**
5: **for** all s, a **do**
6: $\mathtt{Q}[sa] \leftarrow R_{sa} + \gamma \bigoplus_{s'} p(s' \mid s, a)\mathtt{ConQ}[s']$
7: $\mathtt{ConQ}[sa](Z) \leftarrow \mathtt{Q}[sa](X)$ for all X such that $Z = X \cap [s \mapsto a]$ is non-empty
8: Update $\mathtt{ConQ}[s]$ by combining table entries of $\mathtt{ConQ}[sa']$ for all a'
9: **end for**
10: **until** Q converges: $\mathrm{dom}(\mathtt{Q}(sa))$ and $\mathtt{Q}(sa)(X)$ does not change for all s, a, X
11: /* Then recover an optimal policy */
12: $X^* \leftarrow \mathrm{argmax}_X\, \mathtt{ConQ}[s_0](X)$
13: $q^* \leftarrow \mathtt{ConQ}[s_0](X^*)$
14: $\theta^* \leftarrow \mathrm{Witness}(X^*)$
15: return π_{θ^*} and q^*.



该算法能够正确收敛，并且能够保证最优，缺点就是计算复杂度比较高。

Theorem 1. *PCVI (Alg. 1) has the following guarantees:*

(a) *(Convergence and correctness) The* Q *function converges and, for each* $s \in S, a \in A$, *and any* $\theta \in \Theta$: *there is a unique* $X \in \mathsf{dom}(\mathtt{Q}[sa])$ *s.t.* $\theta \in X$ *and*

$$Q^{\pi_\theta}(s,a) = \mathtt{Q}[sa](X). \tag{9}$$

(b) *(Optimality and Non-delusion) Given initial state* s_0, π_{θ^*} *is an optimal policy within* $G(\Theta)$ *and* q^* *is its value.*

(c) *(Runtime bound) Assume* $\oplus$ *and non-emptiness checks (lines 6 and 7) have access to* Witness. *Let*

$$\mathcal{G} = \{g_\theta(s,a,a') := \mathbf{1}[f_\theta(s,a) - f_\theta(s,a') > 0], \forall s, a \neq a' \mid \theta \in \Theta\}, \tag{10}$$

where $\mathbf{1}[\cdot]$ *is the indicator function. Then each iteration of Alg. 1 runs in time* $O(nm \cdot [\binom{m}{2}n]^{2^{\mathsf{VCDim}(\mathcal{G})}}(m-1)w)$ *where* $\mathsf{VCDim}(\cdot)$ *is the VC-dimension [31] of a set of boolean-valued functions, and* w *is the worst-case running time of the oracle called on at most* nm *state-action constraints. Combined with Part (a), if* $\mathsf{VCDim}(\mathcal{G})$ *is finite, then* Q *converges in time polynomial in* n, m, w.

5. 一个model-free的算法（PCQL）

注意到PCVI需要知道环境的dynamics因此是一个model-based的算法，同时它还需要每次遍历所有的state-action对。这里提出类似于Q-learning的算法PCQL，它是一个model-free的算法，它能够通过采样得到的各种数据来更新参数。

- From model-based, iterating over all state-action pair to model-free, training from batch
- Initialize and calculate whenever necessary
- Gradually towards the backed-up value (line 4)

Algorithm 2 Policy-Class Q-learning (PCQL)

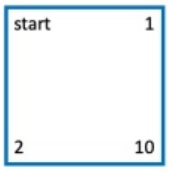
Input: Batch $B = \{(s_t, a_t, r_t, s'_t)\}_{t=1}^{T}$, γ, Θ, scalars α_t^{sa}.

1: **for** $(s, a, r, s') \in B$, t is iteration counter **do**
2: For all a', if $s'a' \notin \texttt{ConQ}$ then initialize $\texttt{ConQ}[s'a'] \leftarrow ([s' \mapsto a'] \mapsto 0)$.
3: Update $\texttt{ConQ}[s']$ by combining $\texttt{ConQ}[s'a'](X)$, for all a', $X \in \text{dom}(\texttt{ConQ}[s'a'])$
4: $\texttt{Q}[sa] \leftarrow (1 - \alpha_t^{sa})\texttt{Q}[sa] \oplus \alpha_t^{sa}(r + \gamma \texttt{ConQ}[s'])$
5: $\texttt{ConQ}[sa](Z) \leftarrow \texttt{Q}[sa](X)$ for all X such that $Z = X \cap [s \mapsto a]$ is non-empty
6: **end for**
7: Return ConQ, Q

实验

这里做实验主要表明该算法能够消除妄想偏差。

PCQL: Experiment



- Grid world
- Each state-action pair with random generated independent standard normal values
- Compare the "deluded value" (dark) and the actual achieved value (light)

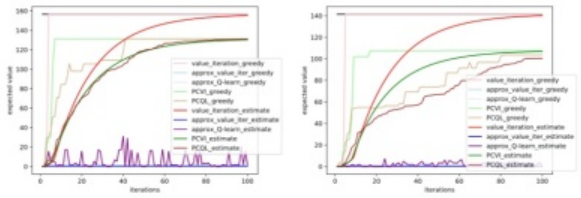


Figure 2: Planning and learning in a grid world with random feature representations. (Left: 4×4 grid using 4 features; Right: 5×5 grid using 5 features.) Here "iterations" means a full sweep over state-action pairs, except for Q-learning and PCQL, where an iteration is an episode of length $3/(1-\gamma) = 60$ using εGreedy exploration with $\varepsilon = 0.7$. Dark lines: estimated maximum achievable expected value. Light lines: actual expected value achieved by greedy policy.

更为可操作的算法

注意到上述算法复杂度非常高，要想准确地消除妄想偏差需要付出高昂的代价。同时注意到PCVI和PCQL都是tabular case的算法，还没有涉及到真正的function approximator的表示。在实际中，状态和行动空间都很大，肯定需要函数近似拟合。这里考虑一些更为实际的算法，付出一定的代价消除一部分妄想偏差。

文章提出了两种可能的框架，一种是维护一个全局的信息集，每次只用一致的（consistent）样本更新该信息集内部的function approximator；另一种则是在每个batch中构建一个临时的信息集，用来确保更新在这个batch内是一致的。

- Multiple regressors
  - PCVI and PCQL are tabular case
  - Information set $I = (X, \omega)$ = regressor that predicts consistent Q-values in the cell ω + constraints defining the cell X
  - A cell's Q-regressor gets updated only if [s ⟼ a] is consistent with constraint of the cell X
  - When new state s is encountered, Information set $I = (X, \omega)$ gets refined by [s ⟼ a] for every a
  - When size of information set gets too large, merge or delete the set by heuristics
    - Merging causes over-relaxed: allow some delusion to creep into Q-values
    - Deleting causes hyper-vigilant: exists some sample contributes to no cell
    - Heuristics include encouraging high Q-values, less constraints or diversity of cells
- Q-learning with locally consistent data
  - Only ensures consistency within one batch
  - No need to maintain a global information set

由于要讲组会就偷个懒直接把非理论部分的PPT贴出来了。ヽ(ゞ▽ゝ)ノ

发布于 2018-12-11

强化学习 (Reinforcement Learning)

文章被以下专栏收录

强化学习前沿
读呀读paper

进入专栏