A Note on BCQ

Hong-Sheng, Xie
Department of Computer Science
National Yang Ming Chiao Tung University
hongshengxin.cs10@nycu.edu.tw

1 Introduction

For the real-world applications of reinforcement learning, we hope that the agent can learn a great policy from a fixed batch of data, i.e. no further interactions with the environment because the data collection process is costly and time-consuming. To achieve this requirement, Off-policy algorithms, such as DQN Mnih et al. [2015] and DDPG Lillicrap et al. [2015], might be a choice because the agent can reuse the past data to update parameters. These algorithms indeed perform well if dataset is close to true distribution. However, they fail to learn a safe policy if dataset deviated from the true distribution. This inability is caused by extrapolation error, in which unseen state-action pairs are estimated to have unrealistic values. To eliminate extrapolation error in off-policy reinforcement learning, they introduce batch-constrained reinforcement learning, where the agent is trained to maximize the expectation of the cumulative discounted reward while minimizing the extrapolation error.

2 Problem Formulation

2.1 Extrapolation Error

This phenomenon is introduced by the mismatch between the dataset and true state-visitation of the current policy. It makes the value estimate $Q(s, a)$ become inaccurate. The unseen state-action pairs are estimated to have unrealistic values. There are three main reasons cause the extrapolation error:

- **Absent Data**
  For batch reinforcement learning, the experience replay dataset $\mathcal{B}$ is fixed. Because of that, we expected that there would be some state-action pairs that don't contain in batches. Moreover, there are no sufficient data near such state-action pairs. Therefore, It is difficult to approximate the true value for absent data.
- **Model Bias**
  When performing off-policy reinforcement learning with a batch $\mathcal{B}$, the Bellman operator $\mathcal{T}^\pi$ is approximated by sampling transition tuples (s, a, r, s') from $\mathcal{B}$ to estimate the value function $Q(s, a)$ over s'. However, this produces a biased estimate for a stochastic MDP without infinite state-action visitation:

  $$\mathcal{T}^\pi Q(s, a) \approx \mathbb{E}_{s' \sim \mathcal{B}}[r + \gamma Q(s', \pi(s'))]$$

  where the expectation is with respect to transitions in the batch $\mathcal{B}$, rather than the true MDP.
- **Training Mismatch**
  In the training process, transitions are sampled uniformly from the dataset typically. Then, the loss is weighted with respect to the likelihood of data in the batch:

  $$\frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s') \in \mathcal{B}} ||r + \gamma Q_{\theta'}(s', \pi(s')) - Q_\theta(s, a)||^2$$

If the distribution of data in the batch is uncorrelated to the distribution under the current policy, the value function may be a poor estimation of actions selected by the current policy, due to the mismatch in training.

2.2 Batch-Constrained Reinforcement Learning

Most modern off-policy reinforcement learning algorithms fail to eliminate extrapolation error because the agent select action with respect to a learned value estimate without consideration of the accuracy of the estimation. For off-policy reinforcement learning algorithms, the value of the agent can be accurately evaluated in regions of the available data. Therefore, they propose a simple idea: in order to avoid extrapolation error, a policy should make a similar state-action visitation to the batch. To optimize off-policy learning for a fixed batch, batch constrained policies are trained to select actions with respect to three objectives:

- Minimize the distance of selected actions to the data in the batch.
- Lead to states where familiar data can be observed.
- Maximize the value function

3 Theoretical Analysis

In the paper, they show the following three things:

- Batch-constrained policies can eliminate extrapolation error for deterministic MDPs by inducing a data distribution that contained entirely within batch.
- The batch-constrained variant of Q-learning converges to the optimal policy under the same conditions as the standard form of Q-learning.
- For deterministic MDP, batch-constrained Q-learning (BCQL) is guaranteed to match, or outperform, the behavioral policy when starting from any state contained in the batch.

From the true MDP M and initial values $Q(s, a)$, they define the new MDP $M_{\mathcal{B}}$ with the same action space and action space as M. Furthermore, $M_{\mathcal{B}}$ has an additional terminal state s_{init}. The transition probability of $M_{\mathcal{B}}$ is defined by $p_{\mathcal{B}}(s'|s, a) = \frac{N(s,a,s')}{\sum_{\tilde{s}} N(s,a,\tilde{s})}$, where $N(s, a, s')$ is number of times the tuple (s, a, s') is observed in $\mathcal{B}$. If $\sum_{\tilde{s}} N(s, a, s') = 0$, then $p_{\mathcal{B}}(s_{init}|s, a) = 1$ and $r(s, a, s_{init})$ is set to the initialized value of $Q(s, a)$.

Singh et al. [2000] prove the convergence of Q-learning rely on Lemma 1. I skip the proof of Lemma 1 in this report, but it is important for us to prove the following theorem.

Lemma 1. *Consider a stochastic process* $(\zeta_t, \Delta_t, F_t), t \geq 0$ *where* $\zeta_t, \Delta_t, F_t : X \to \mathbb{R}$ *satisfy the equation:*

$$\delta_{t+1}(x_t) = (1 - \zeta_t(x_t))\Delta_t(x_t) + \zeta_t(x_t)F_t(x_t)$$

where $x_t \in X$ *and* $t = 0, 1, 2, ...$ *Let* P_t *be a sequence of increasing* σ*-field such that* ζ_0 *and* Δ_0 *are* P_0*-measurable and* ζ_t, Δ_t *and* F_{t-1} *are* P_t*-measurable, t=1, 2, ... Assume that the following hold:*

1. *The set X is finite.*
2. $\zeta_t(x_t) \in [0, 1], \sum_t \zeta_t(x_t) = \infty, \sum_t(\zeta_t(x_t))^2 < \infty$ *with probability 1 and* $\forall x \neq x_t : \zeta(x) = 0$.
3. $||\mathbb{E}[F_t|P_t]|| \leq \kappa||\Delta_t|| + c_t$ *where* $\kappa \in [0, 1)$ *and* c_t *converges to 0 with probability 1.*
4. $Var[F_t(x_t)|P_t] \leq K(1 + \kappa||\Delta_t||)^2$*, where K is some constant.*

Where $\|\cdot\|$ *denotes the maximum norm. Then* Δ_t *converges to 0 with probability 1.*

Theorem 1. *Performing Q-learning by sampling from a batch* $\mathcal{B}$ *converges to the optimal value function under the MDP* $M_{\mathcal{B}}$.

Proof. For any given MDP Q-learning converges to the optimal value function given infinite state-action visitation and some assumptions (see Lemma 1). Sampling under a batch $\mathcal{B}$ with uniform probability satisfies the infinite state-action visitation assumptions of the MDP $M_{\mathcal{B}}$, where given (s, a), the probability of sampling (s, a, s') corresponds to $p_{\mathcal{B}}(s'|s, a) = \frac{N(s,a,s')}{\sum_{\tilde{s}} N(s,a,\tilde{s})}$ in the limit. We remark that for $(s, a) \notin \mathcal{B}, Q(s, a)$ will never be updated, and will return the initialized value, which corresponds to the terminal transition s_{init}. It follows that sampling from $\mathcal{B}$ is equivalent to sampling from the MDP $M_{\mathcal{B}}$, and Q-learning converges to the optimal value function under $M_{\mathcal{B}}$.

Now we define ϵ_{MDP} as the tabular extrapolation error, which is use to measure the discrepancy between the value function $Q^{\pi}_{\mathcal{B}}$ computed with the batch $\mathcal{B}$ and the value function Q^{π} computed with the true MDP M:

$$\epsilon_{MDP}(s, a) = Q^{\pi}(s, a) - Q^{\pi}_{\mathcal{B}}(s, a) \tag{1}$$

For any policy π, we can express the exact form of $\epsilon_{MDP}(s, a)$ though a Bellman-like equation:

$$\begin{aligned}
\epsilon_{MDP}(s, a) &= Q^{\pi}(s, a) - Q^{\pi}_{\mathcal{B}}(s, a) \\
&= \sum_{s'} (p_M(s'|s, a) - p_{\mathcal{B}}(s'|s, a))(r(s, a, s') + \gamma \sum_{a'} \pi(a'|s') Q^{\pi}_{\mathcal{B}}(s', a')) \\
&+ p_M(s'|s, a)\gamma \sum_{a'} \pi(a'|s') \epsilon_{MDP}(s', a')
\end{aligned} \tag{2}$$

If the policy is chosen carefully, the mismatch between value functions can be minimized by visiting regions where the transition distributions are similar, i.e. $p_M(s'|s, a) \approx p_{\mathcal{B}}(s'|s, a)$. For simplicity, we denote

$$\epsilon^{\pi}_{MDP} = \sum_{s} \mu_{\pi}(s) \sum_{a} \pi(a|s) |\epsilon_{MDP}(s, a)| \tag{3}$$

Lemma 2. *For all reward functions, $\epsilon^{\pi}_{MDP} = 0$ if and only if $p_{\mathcal{B}}(s'|s, a) = p_M(s'|s, a)$ for all $s' \in \mathcal{S}$ and (s, a) such that $\mu_{\pi}(s) > 0$ and $\pi(a|s) > 0$.*

Proof.
($\Rightarrow$) By $\epsilon^{\pi}_{MDP} = 0$ and the definition of ϵ^{π}_{MDP}, we know that $\epsilon_{MDP}(s, a) = 0$ for all state-action pairs (s, a). Based on this condition, we know that $\sum_{s'} p_M(s'|s, a)\gamma \sum_{a'} \pi(a'|s')\epsilon_{MDP}(s', a') = 0$. By the equation 2 and $\epsilon_{MDP}(s, a) = 0$ for all state-action pairs (s, a), we get the following equation:

$$\epsilon_{MDP}(s, a) = \sum_{s'} (p_M(s'|s, a) - p_{\mathcal{B}}(s'|s, a))(r(s, a, s') + \gamma \sum_{a'} \pi(a'|s') Q^{\pi}_{\mathcal{B}}(s', a')) \tag{4}$$

$$\begin{gathered}
\because \epsilon_{MDP}(s, a) = 0 \text{ for all state-action pairs and the equation 4.} \\
\therefore p_M(s'|s, a) = p_{\mathcal{B}}(s'|s, a) \text{ for all } s' \in \mathcal{S} \text{ and } (s, a) \text{ such that } \mu_{\pi}(s) > 0 \text{ and } \pi(a|s) > 0.
\end{gathered}$$

($\Leftarrow$)

$$\begin{gathered}
\because p_{\mathcal{B}}(s'|s, a) = p_M(s'|s, a) \text{ for all } s' \in \mathcal{S} \text{ and } (s, a) \text{ such that } \mu_{\pi}(s) > 0 \text{ and } \pi(a|s) > 0. \\
\therefore \epsilon_{MDP}(s, a) = 0 + \sum_{s'} p_M(s'|s, a)\gamma \sum_{a'} \pi(a'|s')\epsilon_{MDP}(s', a') \\
= 0 + \gamma \sum_{s'} \sum_{a'} p_M(s'|s, a)\pi(a|s)\epsilon_{MDP}(s', a')
\end{gathered}$$

We can expand $\epsilon_{MDP}(s, a)$ recursively. Then, we can get $\epsilon_{MDP}(s, a) = 0 + \gamma 0 + \gamma^2 0 + ... = 0$. Hence, $\epsilon^{\pi}_{MDP} = 0$.

Lemma 2 shows that if $M_{\mathcal{B}}$ and M exhibit the same transition probabilities, the policy can be accurately evaluated, i.e. the extrapolation error is eliminated.

Definition 1. A policy $\pi \in \Pi_{\mathcal{B}}$ is batch-constrained if $\pi(a|s) > 0$ for all $(s, a) \in \{(s, a)|\mu_{\pi}(s) > 0 \; and \; \pi(a|s) > 0\}$ then $(s, a) \in \mathcal{B}$.

Definition 2. A batch $\mathcal{B}$ is coherent if $(s, a, s') \in \mathcal{B}$ then $s' \in \mathcal{B}$ unless s' is a terminal state.
(Note: $s \in \mathcal{B}$ means that there exists a transition tuple containing s in the batch $\mathcal{B}$, i.e $(s, a, s') \in \mathcal{B}$)

Theorem 2. *For a deterministic MDP and all reward functions, $\epsilon^{\pi}_{MDP} = 0$ if and only if the policy π is batch-constrained. Furthermore, if $\mathcal{B}$ is coherent, then such a policy must exist if the start $s_0 \in \mathcal{B}$.*

Proof.
Claim: For a deterministic policy π, if $(s, a) \in \mathcal{B}$ then $p_{\mathcal{B}}(s'|s, a) = p_M(s'|s, a)$ for all $s' \in \mathcal{S}$.

$$\because \text{both } \pi \text{ and MDP are deterministic, and } (s, a) \in \mathcal{B}$$

$$\therefore p_{\mathcal{B}}(s'|s, a) = \frac{N(s, a, s')}{\sum_{\tilde{s}} N(s, a, \tilde{s})} = \begin{cases} 1 & if\ p_M(s'|s, a) = 1 \\ 0 & otherwise \end{cases}$$

$$\text{Hence, } p_{\mathcal{B}}(s'|s, a) = p_M(s'|s, a) \text{ for all } s' \in \mathcal{S}.$$

The first part of the theorem follows from Lemma 2. ($\because$ For a deterministic policy π, if $(s, a) \in \mathcal{B}$ then we must have $p_{\mathcal{B}}(s'|s, a) = p_M(s'|s, a)$ for all $s' \in \mathcal{S}$.) For the second part of the theorem, we can construct the batch constrained policy by selecting a in the state $s \in \mathcal{B}$, such that $(s, a) \in \mathcal{B}$. Because the MDP is deterministic and the batch is coherent, when starting from s_0, we must be able to follow at least one trajectory until termination.

Theorem 2 shows that Batch-constrained policies can eliminate extrapolation error for deterministic MDPs. Based on this nice property, they apply the concept of batch-constrained policy to the standard Q-learning. Then, they propose batch-constrained Q-learning (BCQL) which follows the standard tabular Q-learning update while constraining the possible actions with respect to the batch:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a' s.t.(s', a') \in \mathcal{B}} Q(s', a')) \tag{5}$$

Theorem 3. *Given the Robbins-Monro stochastic convergence conditions on the learning rate α, and standard sampling requirements from the environment, BCQL converges to the optimal value function Q^*.*

Proof. Similar to the proof of convergence of Q-learning in Singh et al. [2000].

Theorem 3 shows that the batch-constrained variant of Q-learning converges to the optimal policy under the same conditions as the standard form of Q-learning.

Definition 3. $\pi^* \in \Pi_{\mathcal{B}}$ is optimal batch-constrained policy if π^* satisfies $Q^{\pi^*}(s, a) \geq Q^{\pi}(s, a)$ for all $\pi \in \Pi_{\mathcal{B}}$ and $(s, a) \in \mathcal{B}$.

Theorem 4. *Given a deterministic MDP and coherent batch $\mathcal{B}$, along with the Robbins-Monro stochastic convergence conditions on learning rate α and standard sampling requirements on the batch $\mathcal{B}$, BCQL converges to $Q^{\pi}_{\mathcal{B}}(s, a)$ where $\pi^*(s) = argmax_{a\ s.t.(s,a)\in\mathcal{B}} Q^{\pi}_{\mathcal{B}}(s, a)$ is the optimal batch-constrained policy.*

Proof. By Theorem 1, we know that performing Q-learning by sampling from a batch $\mathcal{B}$ converges to the optimal value function under the MDP $M_{\mathcal{B}}$. Because batch-constrained policies operate only on $\mathcal{B}$, π^* is the optimal batch-constrained policy from the optimality of Q-learning.

Theorem 4 shows that batch-constrained Q-learning (BCQL) is guaranteed to match, or outperform, any behavioral policy when starting from any state contained in the batch.

Then, they propose batch-constrained deep Q-learning (BCQ), which is applied to the same idea as BCQL. Specifically, they use conditional variational autoencoder to generate candidate actions with high similarity to the batch and then select the highest valued action by a learned Q-network. I think that above analysis is enough for us to understand the main idea of this paper. Because they just extend BCQL to deep reinforcement learning, I skip the remaining details about BCQ.

4 Conclusion

They raise an important issue, extrapolation error, in off-policy reinforcement learning with fixed batch setting. This issue cause standard off-policy algorithm can't approximate the true value function preciously. In their work, they introduce the concept of batch-constrained policy to eliminate extrapolation error. Therefore, they propose batch-constrained deep Q-learning (BCQ) where the agent acts close to on-policy with respect to the available data.

References

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Satinder Singh, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvari. Convergence results for single-step on-policy reinforcement-learning algorithms. In *Machine Learning*, 2000.