Statistical and Information-Theoretic Considerations in Fitted Q-Iteration

Zhizhou Ren

March 9, 2020

References

- Main Materials
  - CS598 Statistical Reinforcement Learning:
    Notes on Fitted Q-Iteration (Jiang, 2018)
- Advanced Materials
  - Information-Theoretic Considerations in Batch Reinforcement Learning (Chen and Jiang, 2019)
  - Diagnosing Bottlenecks in Deep Q-learning Algorithms (Fu et al., 2019)
  - Off-Policy Deep Reinforcement Learning without Exploration (Fujimoto et al., 2019)

Background: State Abstraction $\Rightarrow$ Generalization

An abstraction ϕ is … if … $\forall\, s^{(1)}, s^{(2)}$ where $\phi(s^{(1)}) = \phi(s^{(2)})$

- π^*-irrelevant: $\exists\, \pi_M{}^*$ s.t. $\pi_M{}^*(s^{(1)}) = \pi_M{}^*(s^{(2)})$
- Q^*-irrelevant: $\forall\, a$, $Q_M{}^*(s^{(1)}, a) = Q_M{}^*(s^{(2)}, a)$
- Model-irrelevant: $\forall\, a \in A$, $R(s^{(1)}, a) = R(s^{(2)}, a)$
  (bisimulation) $\forall\, a \in A,\ x' \in \phi(S)$, $P(x' \mid s^{(1)}, a) = P(x' \mid s^{(2)}, a)$

$$\boxed{\sum_{s' \in \phi^{-1}(x')} P(s' | s^{(1)}, a)}$$

Theorem: Model-irrelevance $\Rightarrow$ Q^*-irrelevance $\Rightarrow$ π^*-irrelevance

Fitted Q-Iteration (FQI)

Let $D = \{(s, a, r, s')\}$ denote a dataset of past transitions.
The value function is updated iteratively,

$$Q_{t+1} = \arg\min_{Q \in \mathcal{Q}} L_D(Q; Q_t) \approx \mathcal{T} Q_t$$

where $L_D(\cdot; \cdot)$ is the empirical Bellman error evaluated by dataset D.

$$L_D(Q; Q_t) = \frac{1}{|D|} \sum_{(s,a,r,s') \in D} \left(r + \gamma \max_{a' \in \mathcal{A}} Q_t(s', a') - Q(s, a) \right)^2$$

FQI is equivalent to value iteration while $\mathcal{Q} = \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$.

Why do we need target values?

Given a function class $\mathcal{Q}$. Assume realizability $Q^* \in \mathcal{Q}$.
Consider an alternative algorithm:

$$Q^\dagger \leftarrow \arg\min_{Q \in \mathcal{Q}} \frac{1}{|D|} \sum_{(s,a,r,s') \in D} \left(r + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a) \right)^2$$

Does this modification help to simply our algorithmic framework?

Why do we need target values?

Given a function class $\mathcal{Q}$. Assume realizability $Q^* \in \mathcal{Q}$.
Consider an alternative algorithm:

$$Q^\dagger \leftarrow \arg\min_{Q \in \mathcal{Q}} \frac{1}{|D|} \sum_{(s,a,r,s') \in D} \left(r + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a) \right)^2$$

Does this modification help to simply our algorithmic framework?

Unfortunately, $Q^* \neq Q^\dagger$ in some cases.

A Problematic Alternative Objective

Assume we have infinite data across all state-action pairs.

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(s,a,r,s')\sim D}\left[\left(r+\gamma\max_{a'\in\mathcal{A}}Q(s',a')-Q(s,a)\right)^2\right] \\
&= \mathop{\mathbb{E}}_{(s,a)\sim D}\left[\left(\mathop{\mathbb{E}}_{(r,s')\sim D_{s,a}}\left[r+\gamma\max_{a'\in\mathcal{A}}Q(s',a')\right]-Q(s,a)\right)^2\right] && (1) \\
&\quad+\mathop{\mathbb{E}}_{(s,a)\sim D}\left[\mathop{\mathrm{Var}}_{r,s'}\left[r+\gamma\max_{a'\in\mathcal{A}}Q(s',a')\right]\right] && (2)
\end{aligned}
$$

The term (1) is what we want, i.e., $\|(\mathcal{T}Q)(s,a)-Q(s,a)\|_{2,D}$.
The term (2) incorrectly penalizes the variance w.r.t. random transitions.

Workaround #1: Double Sampling Trick

Adopting two independent samples of r and s' (indexed by A and B).

$$\left(\mathbb{E}_{(r,s')\sim D_{s,a}} \left[r + \gamma \max_{a'\in\mathcal{A}} Q(s',a') \right] - Q(s,a) \right)^2$$

$$= \mathbb{E}_{\substack{(r_A,s'_A)\sim D_{s,a}\\(r_B,s'_B)\sim D_{s,a}}} \left[\left(r_A + \gamma \max_{a'\in\mathcal{A}} Q(s'_A,a') - Q(s,a) \right) \left(r_B + \gamma \max_{a'\in\mathcal{A}} Q(s'_B,a') - Q(s,a) \right) \right]$$

It requires a strong assumption on simulator.

Workaround #2: Estimating the Second Term

Adopting another function class $\mathcal{G}$ to estimate the second term.

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(s,a)\sim D}\left[\mathop{\mathrm{Var}}_{r,s'}\left[r+\gamma\max_{a'\in\mathcal{A}} Q(s',a')\right]\right] \\
&= \mathop{\mathbb{E}}_{(s,a,r,s')\sim D}\left[\left(r+\gamma\max_{a'\in\mathcal{A}} Q(s',a') - (\mathcal{T}Q)(s,a)\right)^2\right] \\
&\approx \inf_{g\in\mathcal{G}} \mathop{\mathbb{E}}_{(s,a,r,s')\sim D}\left[\left(r+\gamma\max_{a'\in\mathcal{A}} Q(s',a') - g(s,a)\right)^2\right] \qquad (3)
\end{aligned}
$$

Subtracting the term (3) from the original objective (Antos et al., 2008; Dai et al., 2018).

Return to Fitted Q-Iteration

Q and g can be optimized iteratively.

$$
\begin{aligned}
Q_t = \arg\min_{Q \in \mathcal{Q}} & \mathop{\mathbb{E}}_{(s,a,r,s') \sim D} \left[\left(r + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a) \right)^2 \right] \\
& - \cancel{\inf_{g \in \mathcal{G}}} \mathop{\mathbb{E}}_{(s,a,r,s') \sim D} \left[\left(r + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - g_{t-1}(s, a) \right)^2 \right] \\
g_t = \arg\min_{g \in \mathcal{G}} & \mathop{\mathbb{E}}_{(s,a,r,s') \sim D} \left[\left(r + \gamma \max_{a' \in \mathcal{A}} Q_t(s', a') - g(s, a) \right)^2 \right] \qquad (4)
\end{aligned}
$$

Notice that $g_t = \mathcal{T} Q_t$ is the optimal solution of Eq. (4).
Another kind of target values: g_t is a backup of $\mathcal{T} Q_t$.

Return to Fitted Q-Iteration

More clearly, let $\hat{Q}$ denote a frozen copy of value function.

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(s,a,r,s')\sim D}\left[\left(r + \gamma \max_{a'\in\mathcal{A}} \hat{Q}(s',a') - Q(s,a)\right)^2\right] \\
&= \mathop{\mathbb{E}}_{(s,a)\sim D}\left[\left(\mathop{\mathbb{E}}_{(r,s')\sim D_{s,a}}\left[r + \gamma \max_{a'\in\mathcal{A}} \hat{Q}(s',a')\right] - Q(s,a)\right)^2\right] \\
&\quad + \mathop{\mathbb{E}}_{(s,a)\sim D}\left[\operatorname*{Var}_{r,s'}\left[r + \gamma \max_{a'\in\mathcal{A}} \hat{Q}(s',a')\right]\right] \qquad (5)
\end{aligned}
$$

The term (5) is independent from the selection of Q.

Convergence?

Bellman operator $\mathcal{T}$ is γ-contraction

$$\forall (Q_1, Q_2) \in \mathcal{Q}^2, \quad \|\mathcal{T}Q_1 - \mathcal{T}Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

which provides convergence guarantee for value iteration.
How about Fitted Q-Iteration?

Convergence?

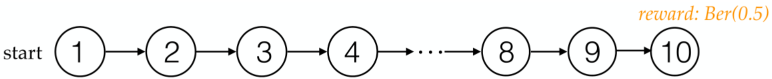
Bellman operator $\mathcal{T}$ is γ-contraction

$$\forall (Q_1, Q_2) \in \mathcal{Q}^2, \quad \|\mathcal{T}Q_1 - \mathcal{T}Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

which provides convergence guarantee for value iteration.
How about Fitted Q-Iteration?

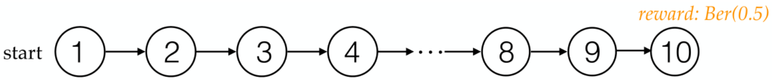
Lots of counterexamples have been proposed (Baird, 1995; Gordon, 1995; Tsitsiklis and Van Roy, 1996).

A Simple Example (Finite Horizon, $\gamma = 1$)



- Dataset $D = \{(s, r, s')\}$ looks like (action omitted):
  $\{(①, 0, ②), (②, 0, ③), \ldots, (⑩, 1, end), \ldots, (⑩, 0, end)\}$

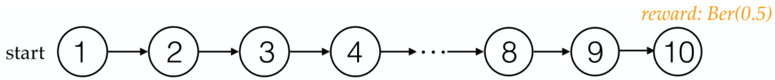
A Simple Example (Finite Horizon, $\gamma = 1$)



FQI Iter #1: Data: (⑩, 1, *end*), …, (⑩, 0, *end*) ⇨ 0.501

- Dataset $D = \{(s, r, s')\}$ looks like (action omitted):
  {(①, 0, ②), (②, 0,③), …, (⑩, 1, *end*), …, (⑩, 0, *end*)}

A Simple Example (Finite Horizon, $\gamma = 1$)

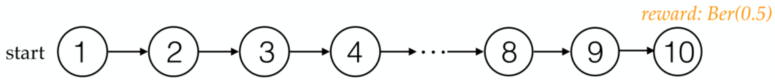


FQI Iter #1: Data: (⑩, 1, *end*), …, (⑩, 0, *end*) ⇨ 0.501

Iter #2: Data: (⑨, 0, ⑩) ⇨ (⑨, 0+0.501) ⇨ 0.501 0.501

- Dataset $D = \{(s, r, s')\}$ looks like (action omitted):
  {(①, 0, ②), (②, 0,③), …, (⑩, 1, *end*), …, (⑩, 0, *end*)}

A Simple Example (Finite Horizon, $\gamma = 1$)



FQI Iter #1: Data: (⑩, 1, *end*), …, (⑩, 0, *end*) ⇨ 0.501

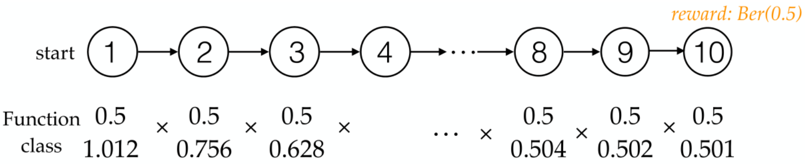
Iter #2: Data: (⑨, 0, ⑩) ⇨ (⑨, 0+0.501) ⇨ 0.501 0.501

…

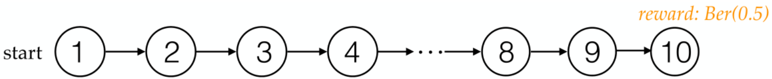
Iter #10: 0.501 0.501 0.501 0.501 … 0.501 0.501 0.501

- Dataset $D = \{(s, r, s')\}$ looks like (action omitted):
  {(①, 0, ②), (②, 0,③), …, (⑩, 1, *end*), …, (⑩, 0, *end*)}

How Things Goes Wrong



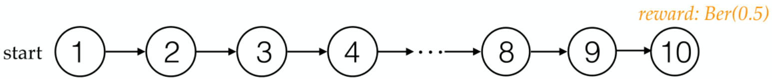
How Things Goes Wrong



Function class $\begin{matrix}0.5\\1.012\end{matrix} \times \begin{matrix}0.5\\0.756\end{matrix} \times \begin{matrix}0.5\\0.628\end{matrix} \times \cdots \times \begin{matrix}0.5\\0.504\end{matrix} \times \begin{matrix}0.5\\0.502\end{matrix} \times \begin{matrix}0.5\\0.501\end{matrix}$

FQI Iter #1: Data: (⑩, 1, *end*), …, (⑩, 0, *end*) ⇨ 0.501

How Things Goes Wrong

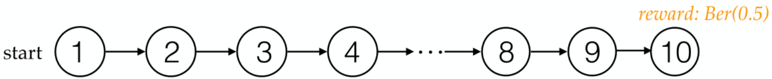


Function class	0.5 / 1.012	×	0.5 / 0.756	×	0.5 / 0.628	×	…	×	0.5 / 0.504	×	0.5 / 0.502	× 0.5 / 0.501

FQI Iter #1: Data: (⑩, 1, *end*), …, (⑩, 0, *end*) ⇨ 0.501

Iter #2: Data: (⑨, 0, ⑩) ⇨ (⑨, 0+0.501) ⇨ 0.502 0.501

How Things Goes Wrong



Function class	0.5 1.012	×	0.5 0.756	×	0.5 0.628	×	⋯	×	0.5 0.504	×	0.5 0.502	×	0.5 0.501

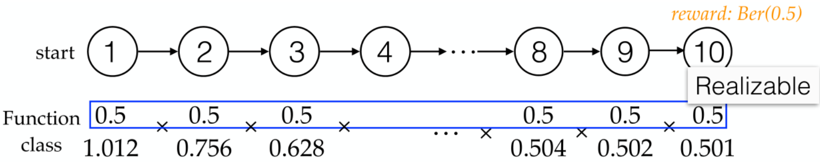
FQI Iter #1: Data: (⑩, 1, *end*), …, (⑩, 0, *end*) ⇨ 0.501

Iter #2: Data: (⑨, 0, ⑩) ⇨ (⑨, 0+0.501) ⇨ 0.502 0.501

⋯

Iter #10: 1.012 0.756 0.628 ⋯ 0.502 0.501

How Things Goes Wrong



FQI Iter #1: Data: (⑩, 1, *end*), …, (⑩, 0, *end*) ⇨ 0.501

Iter #2: Data: (⑨, 0, ⑩) ⇨ (⑨, 0+0.501) ⇨ 0.502 0.501

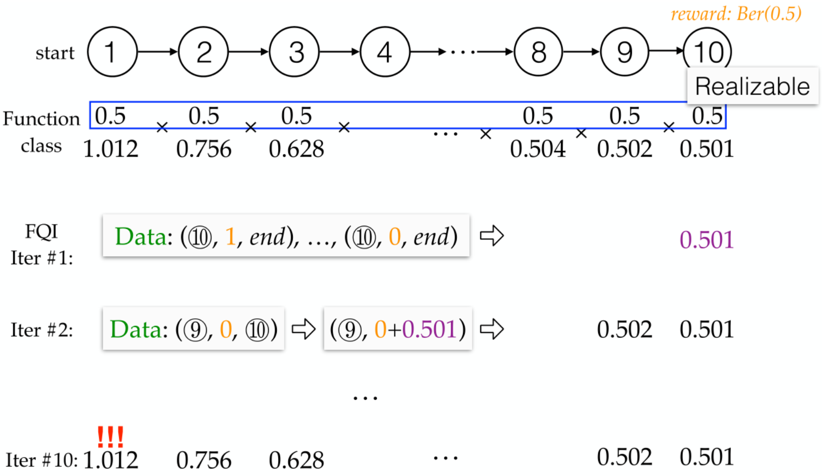
…

!!!

Iter #10: 1.012 0.756 0.628 … 0.502 0.501

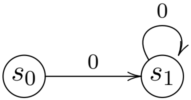
What is the cause of divergence?

Empirical Error? Projection Error?



Another Simple Example with Linear Function Approximation

An MDP with two state s_0, s_1 and features $f(s_0) = 1, f(s_1) = 2$.
Linear Function Approximation $V(s_i) = \theta f(s_i)$.



FQI diverges in this MDP if $\gamma > \frac{5}{6}$ (Tsitsiklis and Van Roy, 1996).

$$
\begin{aligned}
\theta_k &= \arg\min_{\theta} \left[(\theta f(s_0) - \gamma \theta_{k-1} f(s_1))^2 + (\theta f(s_1) - \gamma \theta_{k-1} f(s_1))^2 \right] \\
&= \arg\min_{\theta} \left[(\theta - 2\gamma\theta_{k-1})^2 + (2\theta - 2\gamma\theta_{k-1})^2 \right] \\
&= \frac{6}{5} \gamma \theta_{k-1}
\end{aligned}
$$

Representation Condition of Function Class

Realizability

The optimal value function Q is realizable, i.e. $Q^* \in \mathcal{Q}$.

Completeness

$\mathcal{Q}$ is closed under $\mathcal{T}$, i.e. $\forall Q \in \mathcal{Q}$, $\mathcal{T}Q \in \mathcal{Q}$.
In an approximated view, the violation is measured by

$$\epsilon_{\mathcal{T}} = \sup_{Q \in \mathcal{Q}} \inf_{\hat{Q} \in \mathcal{Q}} \|\hat{Q} - \mathcal{T}Q\|_2^2$$

If $\mathcal{Q}$ is finite, $\epsilon_{\mathcal{T}} = 0$ implies *Realizability*.
If $\epsilon_{\mathcal{T}} = 0$ and data is adequate, FQI is equivalent to value iteration.

One Last Assumption: Data

With *Realizability* and *Completeness* assumptions, FQI works pretty well while data is adequate.

How about the situation with finite samples?

Sample Complexity in Supervised Learning

- $D \in \Delta(\mathcal{X} \times \{0,1\})$ denotes a data distribution, $S \sim D^m$ is a set of samples.
- $\mathcal{H}$ is a finite set of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.
- $R(h)$ and $\hat{R}(h)$ denote the overall error and the empirical error of $h \in \mathcal{H}$.

$$R(h) = \mathop{\mathbb{P}}_{(x,y)\sim D}[h(x) \neq y] \qquad \hat{R}(h) = \frac{1}{m}\sum_{(x,y)\in S} \mathbb{I}[h(x) \neq y]$$

Learning Bound

For a finite hypothesis class $\mathcal{H}$, $\forall \delta > 0$, with probability $1-\delta$, $\forall h \in \mathcal{H}$,

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2m}}$$

To make $R(h) \leq \hat{R}(h) + \epsilon$, we need $m = O\left(\frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{\epsilon^2}\right)$.

Sample Complexity in Supervised Learning

Learning Bound

For a finite hypothesis class $\mathcal{H}$, $\forall \delta > 0$, $\forall h \in \mathcal{H}$, with probability $1 - \delta$,

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log \frac{2}{\delta}}{2m}}$$

To make $R(h) \leq \hat{R}(h) + \epsilon$, we need $m = O\left(\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{\epsilon^2}\right)$.

The dependence $O(\log |\mathcal{H}|)$ comes from *Boole's inequality* (a.k.a. *union bound*).

- For infinite function class, it can be extend to other measures.
- e.g., VC dimension, growth function, covering number.

Sample Complexity in Supervised Learning

Learning Bound

For a finite hypothesis class $\mathcal{H}$, $\forall \delta > 0$, $\forall h \in \mathcal{H}$, with probability $1 - \delta$,

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log \frac{2}{\delta}}{2m}}$$

To make $R(h) \leq \hat{R}(h) + \epsilon$, we need $m = O\left(\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{\epsilon^2}\right)$.

The dependence $O(\log |\mathcal{H}|)$ comes from *Boole's inequality* (a.k.a. *union bound*).

- For infinite function class, it can be extend to other measures.
- e.g., VC dimension, growth function, covering number.

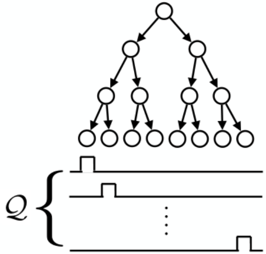
Is it possible to guarantee $\mathrm{Poly}(|\mathcal{A}|, H, \log |\mathcal{Q}|, \frac{1}{\epsilon}, \frac{1}{\delta})$ sample complexity in FQI?

Is it possible to guarantee Poly$(|\mathcal{A}|, H, \log|\mathcal{Q}|, \frac{1}{\epsilon}, \frac{1}{\delta})$ sample complexity?

Notice that unbounded numbers of states are allowed.
Construct a depth-H complete tree to emulate a multi-arm bandit with $|\mathcal{A}|^H$ arms.

Let $\mathcal{Q}$ contain all possible optimal functions.
Then $O(\log|\mathcal{Q}|) = O(H\log|\mathcal{A}|)$ is tractable.

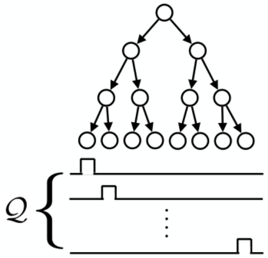
However, the lower bound of sample complexity of this MDP is $\Omega(|\mathcal{A}|^H)$.



Issues on Data Distribution

What causes the exponential sample complexity

- All paths are symmetric.
- Training data should be uniform.



How to define the term "uniform"?
What kind of data distribution is uniform?

Additional Assumption on Data Distribution

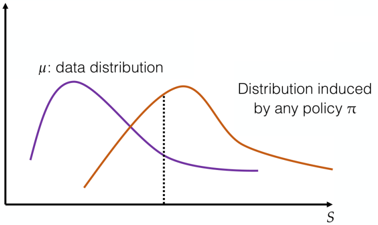
Concentratability

Let $\mu(s)$ denote the data distribution of states, i.e. $D_s \sim \mu$.
There exists a constant $C < \infty$,

$$\forall \nu, \forall s \in \mathcal{S}, \quad \frac{\nu(s)}{\mu(s)} \leq C$$

where ν is generated by a policy π (can be non-stationary and stochastic).

- Sample complexity can have polynomial dependence on C. (Munos, 2003)
- Implicitly assume C is small.



An Ideal Data Distribution

Construct a dataset

- $s \sim \mu$, $a \sim \mathrm{Unif}(\mathcal{A})$, $r \sim R(\cdot|s,a)$, $s' \sim P(\cdot|s,a)$
- A uniform bound on norms, $\forall \nu$, $\forall \pi$,

$$\|\cdot\|_{2,\nu\times\pi} \leq \sqrt{C|\mathcal{A}|}\|\cdot\|_{2,\mu\times\mathrm{Unif}(\mathcal{A})}$$

Under *Completeness* assumption

- To achieve $V^* - V^\pi \leq \epsilon \cdot \frac{R_{\max}}{1-\gamma}$, we need

$$|D| = O\left(\frac{C|\mathcal{A}|\log\frac{|\mathcal{Q}|}{\delta}}{\epsilon^2(1-\gamma)^4}\right)$$

- An error bound with approximated *Completeness* ($\epsilon_{\mathcal{T}} \neq 0$) refers to Chen and Jiang (2019).

The Magnitude of Concentratability Constant C

How large can the constant C be?

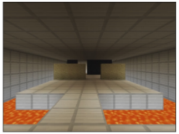
- In the worst case, $C = O(|\mathcal{S}|)$.
- We have not gotten rid of the dependence on state space.

In some specific classes of problems, C is small.

MDPs with Rich Observation (ROMDP)

- a finite hidden state space $\mathcal{Z}$
- an arbitrarily large observation space $\mathcal{S}$
- hidden state dynamics $\Gamma : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$
- emission process $\Psi : \mathcal{Z} \to \Delta(\mathcal{S})$
- $\forall z_1 \neq z_2, \forall s \in \mathcal{S}, \ \Psi(s|z_1) \cdot \Psi(s|z_2) = 0.$
  In other words, this MDP is Markovian w.r.t. $\mathcal{S}$.

hidden state



Markovian high-dimensional observation

Result: In ROMDPs, $C = O(|\mathcal{Z}|)$.

Rethinking Learning State Representation

In ROMDPs, the sample complexity to achieve $V^* - V^\pi \leq \epsilon \cdot \frac{R_{\max}}{1-\gamma}$ is

$$|D| = O\left(\frac{|\mathcal{Z}||\mathcal{A}| \log \frac{|\mathcal{Q}|}{\delta}}{\epsilon^2 (1-\gamma)^4}\right)$$

In an information-theoretic view, the algorithm can learn to generalize by itself.

Rethink: Is it principal to learn a state abstraction explicitly?

Rethinking Learning State Representation

In ROMDPs, the sample complexity to achieve $V^* - V^\pi \le \epsilon \cdot \frac{R_{\max}}{1-\gamma}$ is

$$|D| = O\left(\frac{|\mathcal{Z}||\mathcal{A}| \log \frac{|\mathcal{Q}|}{\delta}}{\epsilon^2 (1-\gamma)^4}\right)$$

In an information-theoretic view, the algorithm can learn to generalize by itself.

Rethink: Is it principal to learn a state abstraction explicitly?

- Deep Q-Learning = FQI + Online Exploration

Rethinking Learning State Representation

In ROMDPs, the sample complexity to achieve $V^* - V^\pi \le \epsilon \cdot \frac{R_{\max}}{1-\gamma}$ is

$$|D| = O\left(\frac{|\mathcal{Z}||\mathcal{A}| \log \frac{|\mathcal{Q}|}{\delta}}{\epsilon^2 (1-\gamma)^4}\right)$$

In an information-theoretic view, the algorithm can learn to generalize by itself.

Rethink: Is it principal to learn a state abstraction explicitly?

- Deep Q-Learning = FQI + Online Exploration
- Potentially pruning function class $\mathcal{Q}$

Rethinking Learning State Representation

In ROMDPs, the sample complexity to achieve $V^* - V^\pi \leq \epsilon \cdot \frac{R_{\max}}{1-\gamma}$ is

$$|D| = O\left(\frac{|\mathcal{Z}||\mathcal{A}| \log \frac{|\mathcal{Q}|}{\delta}}{\epsilon^2 (1-\gamma)^4}\right)$$

In an information-theoretic view, the algorithm can learn to generalize by itself.

Rethink: Is it principal to learn a state abstraction explicitly?

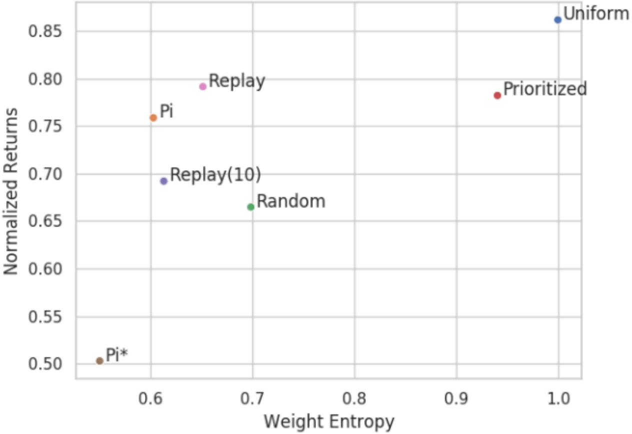
- Deep Q-Learning = FQI + Online Exploration
- Potentially pruning function class $\mathcal{Q}$
- Optimization matters in terms of $\epsilon_{\mathcal{T}}$

Reviewing Prioritized Experience Replay

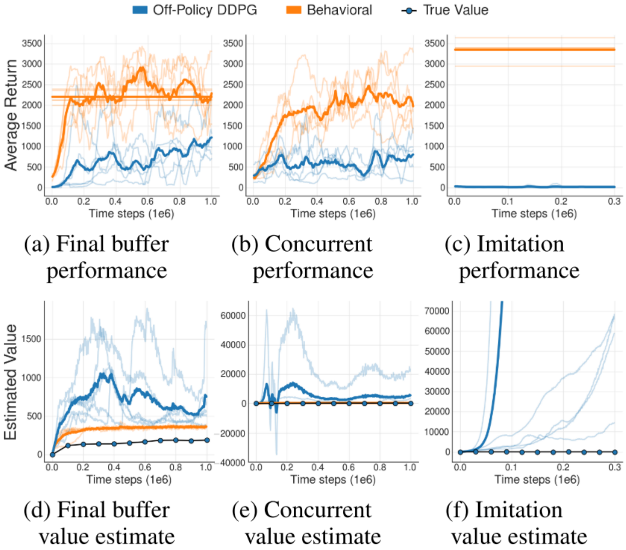
Prioritized Experience Replay (PER)

- manipulating data distribution
- using heuristics to reduce C
- prioritizing by Bellman error (Schaul et al., 2016)
- prioritizing by energy cost (Zhao and Tresp, 2018)

Experiments from Fu et al. (2019)



Experiments from Fujimoto et al. (2019)



Summary

- Assumptions to make FQI provably work
  - Realizability
  - Completeness
  - Concentratability
- Sample complexity of FQI under certain assumptions
- Connecting with empirical results

References I

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. Machine Learning, 71(1):89–129, 2008.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In Machine Learning Proceedings 1995, pages 30–37. Elsevier, 1995.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In International Conference on Machine Learning, pages 1042–1051, 2019.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In International Conference on Machine Learning, pages 1125–1134, 2018.

Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. Diagnosing bottlenecks in deep q-learning algorithms. In International Conference on Machine Learning, pages 2021–2030, 2019.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In International Conference on Machine Learning, pages 2052–2062, 2019.

Geoffrey J Gordon. Stable function approximation in dynamic programming. In Machine Learning Proceedings 1995, pages 261–268. Elsevier, 1995.

Nan Jiang. Notes on fitted q-iteration. 2018.

Rémi Munos. Error bounds for approximate policy iteration. In Proceedings of the Twentieth International Conference on International Conference on Machine Learning, pages 560–567, 2003.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In International Conference on Learning Representations, 2016.

John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. Machine Learning, 22(1-3):59–94, 1996.

Rui Zhao and Volker Tresp. Energy-based hindsight experience prioritization. In Conference on Robot Learning, pages 113–122, 2018.