DEEP LEARNING JP
[DL Papers]

Conservative Q-Learning for Offline Reinforcement Learning

Hiroki Furuta

書誌情報

- **タイトル**: Conservative Q-Learning for Offline Reinforcement Learning
- **著者**: Aviral Kumar[1], Aurick Zhou[1], George Tucker[2], Sergey Levine[12]
- **所属**: [1]UC Berkeley, [2]Google Research, Brain Team
- **URL**: https://arxiv.org/abs/2006.04779
- **概要**: オフライン強化学習で、データセットと学習方策のdistributional shiftによって起こる価値の過大評価を解決するConservative Q-Learning (CQL)を提案した

研究背景

- 最近**Offline RL**(Batch RL, Fully Off-Policy RL)が流行っている
  - ある挙動方策(複数の場合もある)によって集められたデータセットのみから学習方策を最適化、環境との相互作用はなし、実応用向き
    - BCQ[Fujimoto+ 2018], BEAR[Kumar+ 2019], BRAC[Wu+ 2019], AWR[Peng+ 2019], ABM[Siegel+ 2020], QR-DQN[Dabney+ 2018], REM[Agarwal+ 2020], MOReL[Kidambi+ 2020], MOPO[Yu+ 2020], BREMEN[Matsushima+ 2020]
- Offline RLではデータセットと学習方策のdistributional shiftが大きな問題となる
  - 通常のOff-Policyの手法ではデータが増えて緩和される
- 既存手法は学習方策が推定した挙動方策(データセットを集めた方策)から大きく離れないような制約をかけるが、不十分である

準備：問題設定

- 通常のMDP: $(\mathcal{S}, \mathcal{A}, T, r, \gamma)$
- データセットDを集めた挙動方策: $\pi_\beta(\mathbf{a}|\mathbf{s})$
- 挙動方策の元でのdiscounted state-marginal distribution: $d^{\pi_\beta}(\mathbf{s})$
- Q-Learningはベルマン最適作用素を繰り返し適用することでQ関数を学習する手法（actionが高次元の場合はCEMなどでmaxを計算）

$$\mathcal{B}^*Q(\mathbf{s},\mathbf{a}) = r(\mathbf{s},\mathbf{a}) + \gamma \mathbb{E}_{\mathbf{s}'\sim P(\mathbf{s}'|\mathbf{s},\mathbf{a})}[\max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}')]$$

- Actor-Criticでは、 $\mathcal{B}^\pi Q = r + \gamma P^\pi Q$

$$P^\pi Q(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim T(\mathbf{s}'|\mathbf{s},\mathbf{a}),\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}[Q(\mathbf{s}',\mathbf{a}')]$$

Distributional Shift

- 挙動方策の下で集めたデータセットでQ関数と方策を交互に最適化

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}' \sim \mathcal{D}} \left[\left(\left(r(\mathbf{s},\mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}' \sim \hat{\pi}^k(\mathbf{a}'|\mathbf{s}')} [\hat{Q}^k(\mathbf{s}',\mathbf{a}')] \right) - Q(\mathbf{s},\mathbf{a}) \right)^2 \right] \quad \text{(policy evaluation)}$$

$$\hat{\pi}^{k+1} \leftarrow \arg\min_{\pi} \mathbb{E}_{\mathbf{s} \sim \mathcal{D}, \mathbf{a} \sim \pi^k(\mathbf{a}|\mathbf{s})} \left[\hat{Q}^{k+1}(\mathbf{s},\mathbf{a}) \right] \quad \text{(policy improvement)}$$

- Policy evaluationにおいて、学習方策に関する期待値をとってTarget Valueの値を計算
- 学習方策からはデータセットの分布外（OOD）のactionがサンプルされる可能性があり、**OODのactionの価値が過大評価**されるDistributional Shiftの問題に繋がりうる

Conservative Off-Policy Evaluation

- 特定の方策μに関するQ-valueの期待値を最小化する項を追加

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha\, \mathbb{E}_{\mathbf{s}\sim\mathcal{D},\mathbf{a}\sim\mu(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right] + \frac{1}{2}\,\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}(\mathbf{s},\mathbf{a})\right)^2\right]$$

Policy evaluation

➢ 十分大きなαの下で、 $\hat{Q}^{\pi}(\mathbf{s},\mathbf{a}) \leq Q^{\pi}(\mathbf{s},\mathbf{a})$ $\alpha > \frac{C_{r,T}R_{\max}}{1-\gamma} \cdot \max_{\mathbf{s},\mathbf{a}\in\mathcal{D}} \frac{\hat{\pi}_{\beta}(\mathbf{a}|\mathbf{s})}{\mu(\mathbf{a}|\mathbf{s})}$

- データセットの方策に関するQ-valueの期待値を最大化する項を追加

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha \cdot \left(\mathbb{E}_{\mathbf{s}\sim\mathcal{D},\mathbf{a}\sim\mu(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right] - \mathbb{E}_{\mathbf{s}\sim\mathcal{D},\mathbf{a}\sim\hat{\pi}_{\beta}(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right]\right)$$

$$+\frac{1}{2}\,\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}(\mathbf{s},\mathbf{a})\right)^2\right]$$

Q-valueの期待値について
よりtightなバウンドとなる

Policy evaluation

➢ 十分大きなαの下で、 $\hat{V}^{\pi}(\mathbf{s}) \leq V^{\pi}(\mathbf{s})$ $\alpha > \frac{C_{r,T}R_{\max}}{1-\gamma} \cdot \max_{\mathbf{s}\in\mathcal{D}} \left[\sum_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s})\left(\frac{\pi(\mathbf{a}|\mathbf{s})}{\hat{\pi}_{\beta}(\mathbf{a}|\mathbf{s})} - 1\right)\right]^{-1}$

$\hat{V}^{\pi}(\mathbf{s}) = \mathbb{E}_{\pi(\mathbf{a}|\mathbf{s})}[\hat{Q}^{\pi}(\mathbf{s},\mathbf{a})]$

Conservative Q-Learning for Offline RL

- 価値関数に関する最適化問題に加えて、方策に関する最適化も考慮

$$\min_{Q} \max_{\mu} \alpha \left(\mathbb{E}_{\mathbf{s} \sim \mathcal{D}, \mathbf{a} \sim \mu(\mathbf{a}|\mathbf{s})} \left[Q(\mathbf{s}, \mathbf{a}) \right] - \mathbb{E}_{\mathbf{s} \sim \mathcal{D}, \mathbf{a} \sim \hat{\pi}_{\beta}(\mathbf{a}|\mathbf{s})} \left[Q(\mathbf{s}, \mathbf{a}) \right] \right)$$
$$+ \frac{1}{2} \mathbb{E}_{\mathbf{s}, \mathbf{a}, \mathbf{s}' \sim \mathcal{D}} \left[\left(Q(\mathbf{s}, \mathbf{a}) - \hat{\mathcal{B}}^{\pi_k} \hat{Q}^k(\mathbf{s}, \mathbf{a}) \right)^2 \right] + \mathcal{R}(\mu) \quad (\mathrm{CQL}(\mathcal{R}))$$

- 上式によるQ-Learning (or Actor-Critic)をCQL($\mathcal{R}$)と呼ぶ
  - $\mathcal{R}$は方策に関する正則化項で、事前分布 ρ とのKLやエントロピー$\mathcal{H}$を用いる

Variants of CQL

- CQL($\mathcal{H}$)の目的関数: 方策μに関して、

$$\max_{\mu} \ \mathbb{E}_{\mathbf{x}\sim\mu(\mathbf{x})}[f(\mathbf{x})] + \mathcal{H}(\mu) \quad \text{s.t.} \quad \sum_{\mathbf{x}} \mu(\mathbf{x}) = 1,\ \mu(\mathbf{x}) \geq 0\ \forall \mathbf{x},$$

$$\mu^*(\mathbf{x}) = \frac{1}{Z}\exp(f(\mathbf{x}))$$

- $f = Q$として前項の期待値の計算に代入するとCQL($\mathcal{H}$)の目的関数が得られる

$$\min_{Q}\ \alpha\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\log\sum_{\mathbf{a}}\exp(Q(\mathbf{s},\mathbf{a})) - \mathbb{E}_{\mathbf{a}\sim\hat{\pi}_{\beta}(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right]\right] + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q - \hat{\mathcal{B}}^{\pi_k}\hat{Q}^k\right)^2\right]$$

Gap Expanding

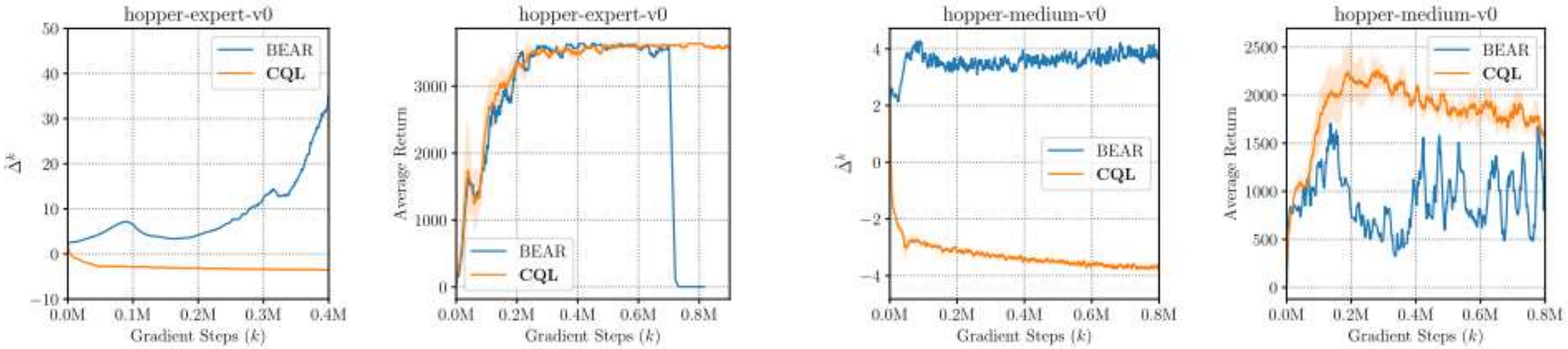
- 十分に大きいαの下で、

$$\mathbb{E}_{\pi_\beta(\mathbf{a}|\mathbf{s})}[\hat{Q}^k(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mu_k(\mathbf{a}|\mathbf{s})}[\hat{Q}^k(\mathbf{s},\mathbf{a})] > \mathbb{E}_{\pi_\beta(\mathbf{a}|\mathbf{s})}[Q^k(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mu_k(\mathbf{a}|\mathbf{s})}[Q^k(\mathbf{s},\mathbf{a})]$$

- CQLではデータセットの分布内の方策によるQ-valueの期待値と分布外の方策によるQ-valueの期待値の差が、真の価値関数による値の差よりも大きくなる
- 分布外のQ-valueが低めに評価されるので、相対的に正確な分布内のQ-valueに基づいて方策を学習できる

CQL vs BEAR(既存手法)

- OODのQ関数の値とデータセット内のQ関数の値の差



(a) hopper-expert-v0　　(b) hopper-medium-v0

$$\hat{\Delta}^k = \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[\max_{\mathbf{a}'_1,\cdots,\mathbf{a}'_N\sim\text{Unif}(\mathbf{a}')}[\hat{Q}^k(\mathbf{s},\mathbf{a}')] - \hat{Q}^k(\mathbf{s},\mathbf{a})\right]$$

分布外の方策によるQ-valueの期待値が分布内の方策によるQ-valueの期待値より小さくなっている

アルゴリズム

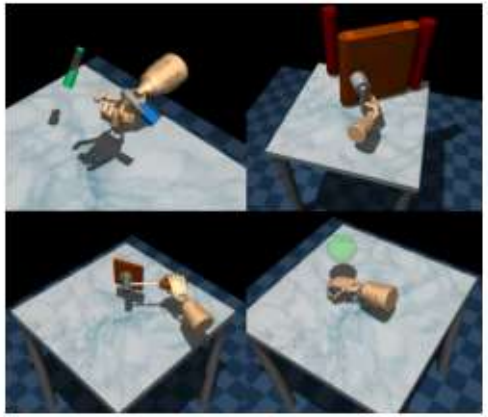
- SACなど既存のQ-LearningやActor-Criticのコードに20行弱加えるだけの簡潔な修正で実現できる

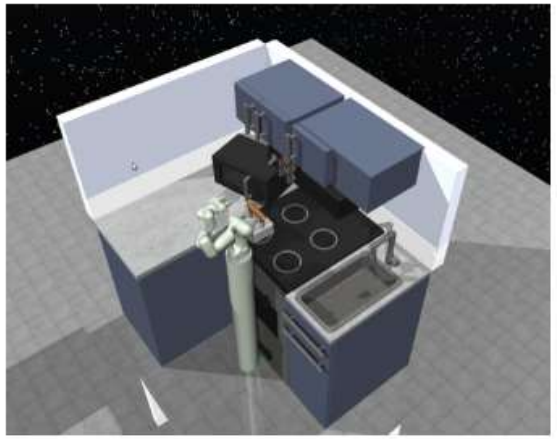
Algorithm 1 Conservative Q-Learning (both variants)

1: Initialize Q-function, Q_θ, and optionally a policy, π_ϕ.
2: **for** step t in $\{1, \ldots, \mathrm{N}\}$ **do**
3: Train the Q-function using G_Q gradient steps on objective from Equation 4
$\theta_t := \theta_{t-1} - \eta_Q \nabla_\theta \mathrm{CQL}(\mathcal{R})(\theta)$
(Use $\mathcal{B}^*$ for Q-learning, $\mathcal{B}^{\pi_{\phi_t}}$ for actor-critic)
4: (only with actor-critic) Improve policy π_ϕ via G_π gradient steps on ϕ with SAC-style entropy regularization:
$\phi_t := \phi_{t-1} + \eta_\pi \mathbb{E}_{\mathbf{s}\sim\mathcal{D}, \mathbf{a}\sim\pi_\phi(\cdot|\mathbf{s})}[Q_\theta(\mathbf{s}, \mathbf{a}) - \log \pi_\phi(\mathbf{a}|\mathbf{s})]$
5: **end for**

評価実験: 環境など

- D4RL[Fu+ 2020] のベンチマーク環境で評価
  - **MuJoCo Gym**: HalfCheetah, Hopper, Walker2d
  - **AntMaze**: MuJoCoのAntで迷路を解くタスク
  - **Adoit**: 24-DoFのハンドを制御、ペン回し、釘打ち、ドア開け、ボールのpick & place
  - **Kitchen**: 9-DoFのマニピュレーターで複数物体のマニピュレーション





結果: MuJoCo Gym

- Expertのパフォーマンスを100に正規化したスコア
- 様々なデータセットで既存手法を上回る成績

Task Name	SAC	BC	BEAR	BRAC-p	BRAC-v	CQL($\mathcal{H}$)
halfcheetah-random	2.1	30.5	25.5	23.5	28.1	**35.4**
hopper-random	9.8	**11.3**	9.5	**11.1**	**12.0**	**10.8**
walker2d-random	1.6	4.1	**6.7**	0.8	0.5	**7.0**
halfcheetah-medium	-4.3	36.1	38.6	**44.0**	**45.5**	**44.4**
walker2d-medium	0.9	6.6	33.2	72.7	**81.3**	79.2
hopper-medium	0.8	29.0	47.6	31.2	32.3	**58.0**
halfcheetah-expert	-1.9	**107.0**	**108.2**	3.8	-1.1	104.8
hopper-expert	0.7	**109.0**	**110.3**	6.6	3.7	**109.9**
walker2d-expert	-0.3	125.7	106.1	-0.2	-0.0	**153.9**
halfcheetah-medium-expert	1.8	35.8	51.7	43.8	45.3	**62.4**
walker2d-medium-expert	1.9	11.3	10.8	-0.3	0.9	**98.7**
hopper-medium-expert	1.6	**111.9**	4.0	1.1	0.8	**111.0**
halfcheetah-random-expert	53.0	1.3	24.6	30.2	2.2	**92.5**
walker2d-random-expert	0.8	0.7	1.9	0.2	2.7	**91.1**
hopper-random-expert	5.6	10.1	10.1	5.8	11.1	**110.5**
halfcheetah-mixed	-2.4	38.4	36.2	**45.6**	**45.9**	**46.2**
hopper-mixed	3.5	11.8	25.3	0.7	0.8	**48.6**
walker2d-mixed	1.9	11.3	10.8	-0.3	0.9	**26.7**

結果: D4RL

- Expertのパフォーマンスを100に正規化したスコア
- AdoitではKLによる正則化の方が良い(CQL(ρ))

Domain	Task Name	BC	SAC	BEAR	BRAC-p	BRAC-v	CQL($\mathcal{H}$)	CQL(ρ)
AntMaze	antmaze-umaze	65.0	0.0	**73.0**	50.0	70.0	**74.0**	**73.5**
	antmaze-umaze-diverse	55.0	0.0	61.0	40.0	70.0	**84.0**	61.0
	antmaze-medium-play	0.0	0.0	0.0	0.0	0.0	**61.2**	4.6
	antmaze-medium-diverse	0.0	0.0	8.0	0.0	0.0	**53.7**	5.1
	antmaze-large-play	0.0	0.0	0.0	0.0	0.0	**15.8**	3.2
	antmaze-large-diverse	0.0	0.0	0.0	0.0	0.0	**14.9**	2.3
Adroit	pen-human	34.4	6.3	-1.0	8.1	0.6	37.5	**55.8**
	hammer-human	1.5	0.5	0.3	0.3	0.2	**4.4**	2.1
	door-human	0.5	3.9	-0.3	-0.3	-0.3	**9.9**	**9.1**
	relocate-human	0.0	0.0	-0.3	-0.3	-0.3	0.20	**0.35**
	pen-cloned	**56.9**	23.5	26.5	1.6	-2.5	39.2	40.3
	hammer-cloned	0.8	0.2	0.3	0.3	0.3	2.1	**5.7**
	door-cloned	-0.1	0.0	-0.1	-0.1	-0.1	0.4	**3.5**
	relocate-cloned	**-0.1**	-0.2	-0.3	-0.3	-0.3	**-0.1**	**-0.1**
Kitchen	kitchen-complete	33.8	15.0	0.0	0.0	0.0	**43.8**	31.3
	kitchen-partial	33.8	0.0	13.1	0.0	0.0	**49.8**	**50.1**
	kitchen-undirected	47.5	2.5	47.2	0.0	0.0	**51.0**	**52.4**

結果：Discrete Action

- 離散actionのAtariのゲーム環境にCQLを適用
- online DQN agentが集めた最初の1%(top)/10%(bottom)のデータ

Task Name	QR-DQN	REM	CQL($\mathcal{H}$)
Pong (1%)	-13.8	-6.9	**19.3**
Breakout	7.9	11.0	**61.1**
Q*bert	383.6	343.4	**14012.0**
Seaquest	672.9	499.8	**779.4**
Asterix*	166.3	386.5	**592.4**
Pong (10%)	15.1	8.9	**18.5**
Breakout	151.2	86.7	**269.3**
Q*bert	7091.3	8624.3	**13855.6**
Seaquest	2984.8	**3936.6**	3674.1
Asterix*	**189.2**	75.1	156.3

Analysis of CQL

- (学習したQ関数の期待値) − (真の価値の期待値)
- CQL($\mathcal{H}$), CQL(データセットの方策による価値の最大化なし), Q関数のアンサンブル, BEAR(Offlineの既存手法)で比較
- 学習されるQ関数はtightな下界になっている

Task Name	CQL($\mathcal{H}$)	CQL (Eqn. 1)	Ensemble(2)	Ens.(4)	Ens.(10)	Ens.(20)	BEAR
hopper-medium-expert	**-43.20**	-151.36	3.71e6	2.93e6	0.32e6	24.05e3	65.93
hopper-mixed	**-10.93**	-22.87	15.00e6	59.93e3	8.92e3	2.47e3	1399.46
hopper-medium	**-7.48**	-156.70	26.03e12	437.57e6	1.12e12	885e3	4.32

参考: CQL(データセットの方策による価値の最大化なし)

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \; \alpha \, \mathbb{E}_{\mathbf{s}\sim\mathcal{D},\mathbf{a}\sim\mu(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right] + \frac{1}{2}\,\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}(\mathbf{s},\mathbf{a})\right)^2\right]$$

まとめ

- オフライン強化学習で、データセットと学習方策のdistributional shiftによって起こる価値の過大評価を解決するConservative Q-Learning (CQL)を提案
- 真の価値関数の値のtightな下界を与えるQ関数を学習できる
- データセットの分布内のactionのQ関数の期待値と分布外のactionのQ関数の期待値の差が、真の価値関数による値の差よりも大きくなる性質によってdistributional shiftの問題を解決