🔥OFFLINE RL

Created by	
Created	@March 6, 2022
Tags	OfflineRL

卢宗清老师的笔记：Offline RL/Model-Based RL问题
Introduction
- Motivation
- Offline RL vs. IL(模仿学习)
- Offline RL的主要问题 ⇒ Distributional Shift
- Offline RL方法分类
- Offline RL的研究热点
- Offline RL主要参考文献

Offline RL-Step By Step
- SOTA1 — DT Decision Transformer

Valuable Github Repo
- 1.**Re-implemented Offline Algorithms**

这篇笔记主要是对Offline RL的总结，以及一些关键算法的实现。

卢宗清老师的笔记：Offline RL/Model-Based RL问题

【RLChina 2021】第9课 强化学习前沿（一） 卢宗青_哔哩哔哩_bilibili

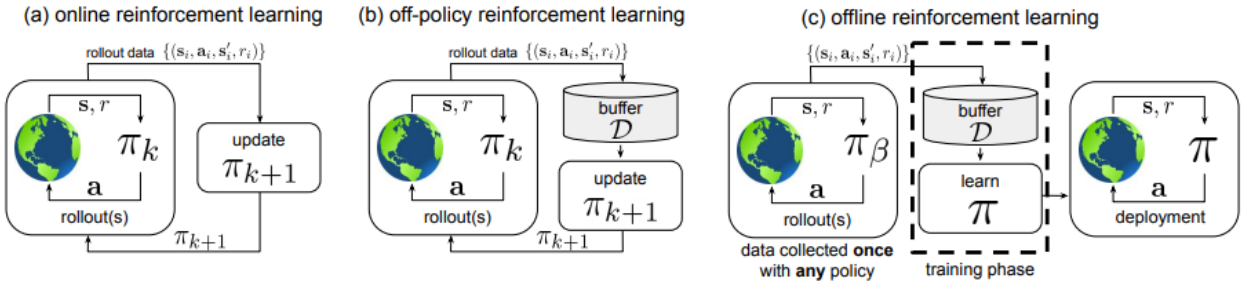
【RLChina 2021】第9课 强化学习前沿（一） 卢宗青

https://www.bilibili.com/video/BV1cQ4y1m7Nn?spm_id_from=333.337.search-card.all.click

Introduction

Offline RL，又称Batch RL，是RL的变体，需要Agent从固定批次的数据中学习，而**不进行探索**。

- 研究如何最大限度利用静态数据集训练RL的Agent.



Motivation

研究Offline RL的动机

- **探索存在成本**：例如，使用机器人/自动驾驶车辆，在真实环境中进行探索可能会有损坏硬件或周围物体/行人的风险。
- **验证算法Exploitation能力**：由于离线强化学习将exploration和exploitation分离开来，它可以提供标准化的比较来验证不同算法的exploitation能力

Offline RL vs. IL(模仿学习)

模仿学习也是从固定的数据集中进行学习，而不进行探索，它们之间有几个关键的区别：

- **现有的Offline RL算法建立在标准的off policy DRL算法之上**，算法倾向于优化某种形式的Bellman方程或TD差分误差。
- 大多数IL问题假设**有一个最优的或一个高性能的专家**来提供数据；而Offline RL可能需要**从次优的数据中进行学习**。
  - Offline RL Can Stitch(缝合) Parts of Good Behaviors
  - 将好的Experience 缝合在一起，学到Return/cumulative rewards最大的behavior
- **大多数IL问题没有reward的概念**；Offline RL考虑reward，方便在事后进行处理和修改。
- **一些IL问题要求数据被标记为专家经验和非专家经验**，Offline RL不做这个假设

Offline RL的主要问题 ⇒ Distributional Shift

- 即Overestimation产生的问题；
- Discrepancy（差异） between ***Behavior Policy*** (行为策略/Data Collection Policy）and ***Learning Policy*** （目标策略）
  - **Behavior Policy是与环境交互的策略** π_β
  - Offline Q Learning algorithms can overestimate the value of unseen actions, and thus be falsely optimistic ...
- Offline RL的overestimation会非常严重：
  - Online RL，会有一些解决overestimation的方法
    - Double DQN
    - 探索操作，执行error correction
    - **Offline RL无法探索**，没法与环境进行交互...

Offline RL方法分类

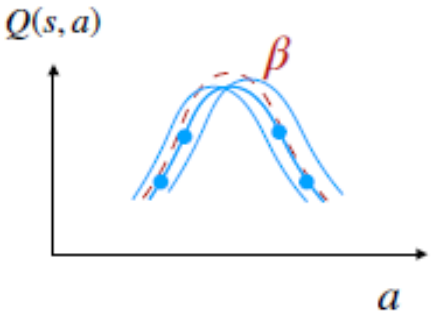
▼ **Policy Constraints （软约束）**

是一种**间接**解决问题的方式（软约束），只有Overestimation才会影响Performance...

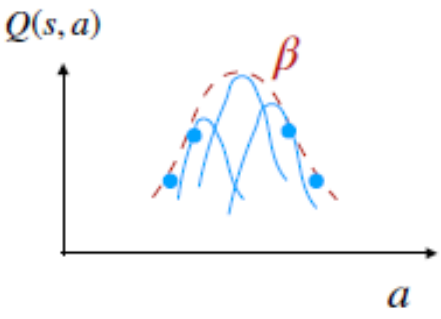
- 优化目标为

  $\max_\pi \sum_{t=0}^{\infty} \mathbb{E}_{s_t \sim d^\pi(s), a_t \sim \pi(a|s)} \left[\gamma^t r\left(s_t, a_t\right)\right] - \alpha D\left(\pi(a \mid s), \pi_\beta(a \mid s)\right)$

  把Behavior Policy与Learning Policy的Divergence 限定在很小的范围内， 避免action出现out-of-distribution的情况



左KL(BRAC),



右MMD(BEAR)；MMD是subset即可

- **BCQ**（选择Best Q的action时，选择Dataset中见过的action ...)

$$\text{BCQ}: Q(s,a) \leftarrow r + \gamma \max_{a' s.(s',a') \in \mathcal{D}} Q\left(s', a'\right)$$

- **BRAC** (KL Divergence)
- **BEAR (** Support Matching)

  ... ...
- 本质上就是不同的带约束问题

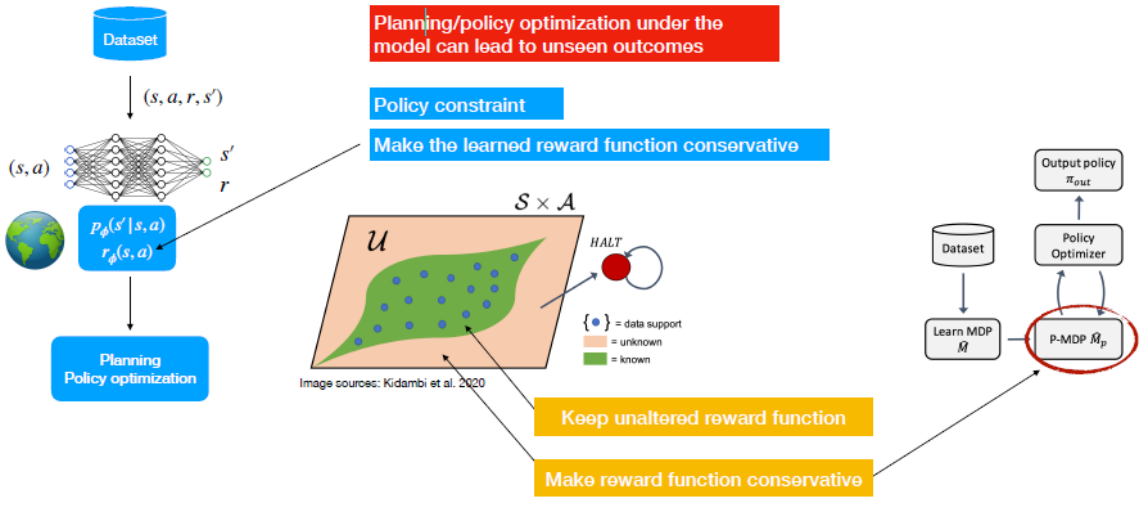
$$\pi_\theta := \arg\max_\theta \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi_\theta(a|s)}[Q(s,a)] \quad \text{s.t.} \quad D\left(\pi_\theta(a \mid s), \pi_\beta(a \mid s)\right) \leq \epsilon$$

▼ 问题：

1. **无论哪种方法，都需要一种评估形式，去评估Behavior Policy** (需要某种方式来评估Behavior Policy)
   - If the behavior policy is estimated ***wrongly***, policy constraint methods can fail 如果behavior policy 本身就是评估错误的，那策略约束方法就会失败...
2. **太保守 Too Conservative**
   - 并不一定非要限制behavior policy与我们要学习的policy非常接近：Consider at a state, all actions have reward 0, then **policy constraint is not necessary**

▼ Model-Based Offline RL

在some cases中，不是所有**out-of-dist**都是坏的... 只有当他们无法产生很好的回报值时，才是真正的overestimation... 所以讨论如何直接解决overestimation问题。

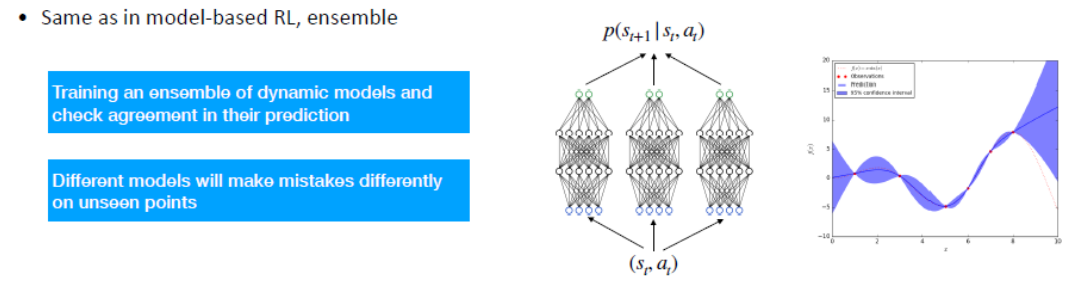


从Dataset中学Transition P_ϕ或者 Reward Func R_ϕ → 进行Planning & Policy Optimization

1. **Conservative Model-Based Offline RL**

   学习State Trans Model $P(s'|s,a)$或者$r(s,a)$...

   - **ENSEMBLE**方法



     - 训练 an ensemble of dynamic models 检查他们预测的一致性
     - 不确定性Uncertainty越大，说明是未见过的action...
     - 不确定性Uncertainty越小，说明是Dataset中的action...

2. **Model-Based Offline RL Methods — MOReL**

   Disagreement in an ensemble of dynamic models　*MOReL (Kidambi et al. 2020)*

   建模不确定度，并给予一个惩罚...

   - $\tilde{r}(s,a) = r(s,a) - \lambda u(s,a)$
   - $dis(s,a) = max_{i,j}||f_{\phi_i}(s,a) - f_{\phi_j}(s,a)||_2$
     - 若$dis(s,a)$过大，大于Threshold，则给予r惩罚

3. **MOPO**

   Covariance matrix of an ensemble of dynamics models, *MOPO (Yu et al. 2020)* , *Dyna*

   $p_i(s' \mid s,a) = \mathcal{N}(\mu_i(s,a), \Sigma_i(s,a))$　　$u(s,a) = \max_j \|\Sigma_j(s,a)\|_F$

   学了一个高斯分布的Transition

   - 每一个Dynamic Model都会预测一个mean μ和协方差矩阵 ...
   - 将Uncertanty设为Vector Norm最大的哪个值，作为惩罚函数u ...
   - Less Conservative in Many Cases!

▼ Value Function Regularization

直接对Q/V进行惩罚...

Dataset type	Environment	BC	COMBO (ours)	MOPO	CQL	SAC-off	BEAR	BRAC-p	BRAC-v
random	halfcheetah	2.1	**38.8**	35.4	35.4	30.5	25.1	24.1	31.2
random	hopper	1.6	**17.9**	11.7	10.8	11.3	11.4	11.0	12.2
random	walker2d	9.8	7.0	**13.6**	7.0	4.1	7.3	-0.2	1.9
medium	halfcheetah	36.1	**54.2**	42.3	44.4	-4.3	41.7	43.8	46.3
medium	hopper	29.0	**94.9**	28.0	86.6	0.8	52.1	32.7	31.1
medium	walker2d	6.6	75.5	17.8	74.5	0.9	59.1	77.5	**81.1**
medium-replay	halfcheetah	38.4	**55.1**	53.1	46.2	-2.4	38.6	45.4	47.7
medium-replay	hopper	11.8	**73.1**	67.5	48.6	3.5	33.7	0.6	0.6
medium-replay	walker2d	11.3	**56.0**	39.0	32.6	1.9	19.2	-0.3	0.9
med-expert	halfcheetah	35.8	**90.0**	63.3	62.4	1.8	53.4	44.2	41.9
med-expert	hopper	111.9	**111.1**	23.7	111.0	1.6	96.3	1.9	0.8
med-expert	walker2d	6.4	96.1	44.6	**98.7**	-0.1	40.1	76.9	81.6

Less conservative than CQL

COMBO learns the Q-function over a richer set of states beyond the states in offline dataset

Value Function Regular各个方法的对比

1. 比如，**按照policy constraint添加Penalty** ...

   $$Q(s,a) \leftarrow r(s,a) + \gamma \mathbb{E}_{a' \sim \pi_\theta(a|s)} \left[Q\left(s', a'\right) \right] - \alpha D\left(\pi_\theta, \pi_\beta\right) \text{ (BRAC)}$$

2. **Conservative Q-Learning ... CQL** (*Kumar et al. 2019*)

   Change the objective to make penalizing automatically ... 自动化惩罚机制

$$\min_Q \max_\mu \mathbb{E}_{s\sim\mathcal{D}} \mathbb{E}_{a\sim\mu(a|s)}[Q(s,a)] + \frac{1}{2\alpha} \mathbb{E}_{s,a,r,s'\sim\mathcal{D}}[(Q(s,a) - (r + \gamma \mathbb{E}_{a'\sim\pi_\theta(a|s)}[\bar{Q}(s',a')]))^2]$$

Minimize big Q-values

Standard Bellman error

What this does?

Push up values on seen state-action pairs, while minimizing others

   - 希望， Dataset见过的 s, a pairs 的 Q 值更高，其他的尽量最小化unseen pairs...
   - 原文可以证明，这个Q^π_{CQL}是真实Q的一个下界...
   - $a \sim \mu(a|s)$可以从任意的Policy中sample ...
   - Too Conservative

3. **CQL-$\mathcal{H}$ Improvement**

$$Q^\pi_{\text{CQL}}(s,a) := \min_Q \max_\mu (\mathbb{E}_{s\sim\mathcal{D}} \mathbb{E}_{a\sim\mu(a|s)}[Q(s,a)] - \mathbb{E}_{s,a\sim\mathcal{D}}[Q(s,a)]) + \frac{1}{2\alpha} \mathbb{E}_{s,a,r,s'\sim\mathcal{D}}[(Q(s,a) - (r + \gamma \mathbb{E}_{a'\sim\pi_\theta(a|s)}[\bar{Q}(s',a')]))^2]$$

Minimize big Q-values

Maximize data Q-values

Standard Bellman error

- V的Lower Bound

$$Q^{\pi}_{\text{CQL}}(s,a) \le Q(s,a) \quad \forall s \in \mathcal{D}, a \quad \Rightarrow \quad V^{\pi}_{\text{CQL}}(s) := \mathbb{E}_{a\sim\pi_\theta(a|s)}[Q^{\pi}_{\text{CQL}}(s,a)] \le V^{\pi}(s) \quad \forall s \in \mathcal{D}$$

In practice $\min_Q \alpha \mathbb{E}_{s\sim\mathcal{D}}\left[\log \sum_a \exp(Q(s,a) - \mathbb{E}_{a\sim\hat{\pi}_\beta(a|s)}[Q(s,a)]\right] + \frac{1}{2}\mathbb{E}_{s,a,r,s'\sim\mathcal{D}}[(Q(s,a) - \bar{\mathcal{B}}^{\pi}\bar{Q}(s,a))^2]$

Update Q_ϕ to decrease the loss above
Update π_θ to increase $\mathbb{E}_{s\sim d^{\pi_\theta}(s), a\sim\pi_\theta(a|s)}[Q_\phi(s,a)]$

Kumar et al. 2020

4. **<u>Value-function Regularization in Model-based Offline RL (COMBO) (Yu et al. 2021)</u>**

$$Q(s,a) := \arg\min_Q \beta\left(\mathbb{E}_{s,a\sim\rho(s,a)}[Q(s,a)] - \mathbb{E}_{s,a\sim\mathcal{D}}[Q(s,a)]\right) + \frac{1}{2}\mathbb{E}_{s,a,r,s'\sim d_f}\left[\left(Q(s,a) - \bar{\mathcal{B}}^{\pi}\bar{Q}(s,a)\right)^2\right]$$

Minimize Q-values of state-action pairs from model rollout

Maximize Q-values of state-action pairs from dataset

Standard Bellman error for mixed samples from dataset and model rollout

$\rho(s,a) = d^{\pi}_{\hat{\mathcal{M}}}(s)\pi(a|s)$

$d_f := fd(s,a) + (1-f)d^{\mu}_{\hat{\mathcal{M}}}(s,a)$

μ can be π or uniform

$\mathbb{E}_{s\sim\mu_0,a\sim\pi(a|s)}[\hat{Q}^{\pi}(s,a)] \le \mathbb{E}_{s\sim\mu_0,a\sim\pi(a|s)}[Q^{\pi}(s,a)]$

- ρ 是在根据当前的policy π，以及学出来的Model去**<u>Roll out</u>**得到的分布
- 最大化Dataset的(s,a)的Q, 最小化其他的Q ...
- **Standard Bellman error for mixed samples** from **dataset and model rollout:**
  - 真实数据dsa和基于model去做rollout的数据组合（rollout可以是π可以是random policy ...) 来update bellman error ...
- COMBO的Q仍然是真实Q的Low Bound

$$\mathbb{E}_{s\sim\mu_0,a\sim\pi(a|s)}\left[\hat{Q}^{\pi}(s,a)\right] \le \mathbb{E}_{s\sim\mu_0,a\sim\pi(a|s)}\left[Q^{\pi}(s,a)\right]$$

▼ **Uncertainty-Based Offline RL Methods**

❖ Model-based offline RL

- This has been covered

$\tilde{r}(s,a) = r(s,a) - \lambda u(s,a)$

❖ Model-free offline RL

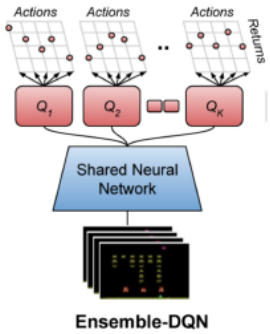
- Directly model uncertainty of Q-function
- $P_{\mathcal{D}}(Q^{\pi})$, uncertainty set of Q-function given the dataset
- Bootstrap ensembles of Q-function

N Q-functions $Q_{\theta_1}, Q_{\theta_2}, \ldots, Q_{\theta_K}$

Train each Q-function independently

$P_{\mathcal{D}}(Q^{\pi}) \approx \frac{1}{K}\sum_{j=1}^{K} \delta[Q^{\pi} = Q_{\theta_j}]$

Can have very little diversity in an ensemble and incorrect uncertainty (Fujimoto et al. 2019)



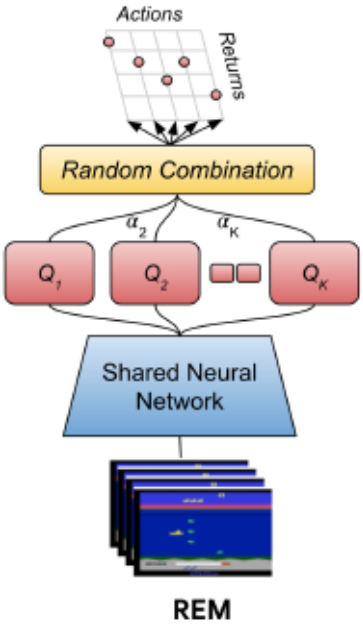
侧重点在model-free Offline RL

- 直接对Q-function的不确定性$P_{\mathcal{D}}(Q^{\pi})$进行建模...

Ensemble-DQN就是某种Q-function Uncertainty...

- 直接取多头Q的均值，不够准确...

1. **Random Ensemble Mixture (REM)** *Agarwal et al. 2021*



- 对每一个Mini Batch，加上随机的Categorical Distribution ..
- 把加权之后的值做Loss...

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s' \sim \mathcal{D}} \left[\mathbb{E}_{\alpha_1,\ldots,\alpha_K \sim \mathrm{P}_\Delta} \left[l_\delta \Big(\sum_k \alpha_k Q_\theta^k(s,a) - r - \gamma \max_{a'} \sum_k \alpha_k Q_{\theta'}^k(s',a') \Big) \right] \right]$$

random categorical distribution for each mini-batch

$\mathcal{L}(\theta)$ can be seen as a finite set of constraints

$$Q(s,a) = \frac{1}{K} \sum_{i=1}^{K} Q_\theta^i(s,a)$$

2. **<u>UWAC Uncertainty weighted actor-critic</u>** (UWAC) *Wu et al. 2021*

   通过Monte Carlo Dropout去做不确定性估计...

- The uncertainty is captured by the predictive variance w.r.t $\hat{Q}$ for T stochastic forward passes

$$\mathrm{Var}[Q(s,a)] \approx \sigma^2 + \frac{1}{T} \sum_{t=1}^{T} \hat{Q}_t(s,a)^\top \hat{Q}_t(s,a) - \mathbb{E}[\hat{Q}(s,a)]^\top \mathbb{E}[\hat{Q}(s,a)]$$

inherent noise in data

Model uncertainty

- Uncertainty-weighted policy distribution π'

$$\pi'(a \mid s) = \frac{\beta}{\mathrm{Var}\left[Q_\theta^\pi(s,a)\right]} \pi(a \mid s) / Z(s)$$

$$\mathcal{L}(Q_\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \mathbb{E}_{a' \sim \pi(\cdot \mid s')} \left[\frac{\beta}{\mathrm{Var}\left[Q_{\theta'}(s',a')\right]} \left(Q_\theta(s,a) - r(s,a) - \gamma Q_{\theta'}(s',a') \right)^2 \right]$$

$$\mathcal{L}(\pi) = -\mathbb{E}_{a \sim \pi'(\cdot \mid s)} \left[Q_\theta(s,a) \right] = -\mathbb{E}_{a \sim \pi(\cdot \mid s)} \left[\frac{\beta}{\mathrm{Var}\left[Q_\theta(s,a)\right]} Q_\theta(s,a) \right]$$

Task Name	UWAC (OURS)	MOPO	MOReL	BEAR	BRACv	AWR	BCQ	BC	CQL	REM
halfcheetah-random	14.5 ± 3.3	35.4 ± 2.5	25.6	25.1	31.2	2.5	2.2	2.1	**35.4**	-2.6
walker2d-random	15.5 ± 11.7	13.6 ± 2.6	**37.3**	7.3	1.9	1.5	4.9	1.6	7	-0.3
hopper-random	22.4 ± 12.1	11.7 ± 0.4	**53.6**	11.4	12.2	10.2	10.6	9.8	10.8	0.7
halfcheetah-medium	**46.5** ± 2.5	42.3 ± 1.6	42.1	41.7	46.3	37.4	40.7	36.1	44.4	-2.6
walker2d-medium	57.5 ± 7.8	17.8 ± 19.3	77.8	59.1	**81.1**	17.4	53.1	6.6	79.2	-0.2
hopper-medium	88.9 ± 12.2	28.0 ± 12.4	**95.4**	52.1	31.1	35.9	54.5	29.0	58	0.6
halfcheetah-med-replay	46.8 ± 3.0	**53.1** ± 2.0	40.2	38.6	47.7	40.3	38.2	38.4	46.2	-3.0
walker2d-med-replay	27.0 ± 6.3	39.0 ± 9.6	**49.8**	19.2	0.9	15.5	15.0	11.3	26.7	-0.2
hopper-med-replay	39.4 ± 6.1	67.5 ± 24.7	**93.6**	33.7	0.6	28.4	33.1	11.8	48.6	0.8
halfcheetah-med-expert	**127.4** ± 3.7	63.3 ± 38.0	53.3	53.4	41.9	52.7	64.7	35.8	62.4	-2.6
walker2d-med-expert	**99.7** ± 12.2	44.6 ± 12.9	95.6	40.1	81.6	53.8	57.5	6.4	98.7	-0.2
hopper-med-expert	**134.7** ± 21.2	23.7 ± 6.0	108.7	96.3	0.8	27.1	110.9	111.9	111	0.7
halfcheetah-expert	**128.6** ± 2.9	-	-	108.2	-1.1	-	-	107	104.8	-
walker2d-expert	121.1 ± 22.4	-	-	106.1	0	-	-	125.7	**153.9**	-
hopper-expert	**135.0** ± 14.1	-	-	110.3	3.7	-	-	109	109.9	-

Work quite well in med-expert and expert
Generally better than CQL

Not as good as model-based methods in random, medium, medium replay

Offline RL的研究热点

- 现阶段，Offline RL主要还是集中在Overestimation本身
- Offline MARL并没有吸引太多的注意
- Decision Tranformer非常的惊艳...

Offline RL主要参考文献

1. Levine, Kumar, Tucker, Fu (2020). Offline Reinforcement Learning: Tutorial, Survey and Perspectives on Open Problems
2. Kumar et al. (2019) Stabilizing Off-Policy Reinforcement Learning via Bootstrapping Error Reduction. **NeurIPS 2019**
3. Fujimoto et al. (2019) Off-Policy Reinforcement Learning without Exploration. **ICML2019**
4. Wu et al. (2019). Behavior Regularized Offline Reinforcement Learning
5. Peng et al. (2019). Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning
6. Nahum and Dai (2019) Reinforcement Learning via Fenchel-Rockafeller Duality
7. Wang et al. (2020). Critic-regularized Regression. **NeurIPS 2020**
8. Kidambi et al. (2020) MOReL: Model-Based Offline Reinforcement Learning. **NeurIPS** 2020.
9. Yu et al. (2020) MOPO: Model-based Offline Policy Optimization. **NeurIPS 2020**.
10. Kumar et al. (2020) Conservative Q-Learning for Offline RL. **NeurIPS 2020**.
11. Yu et al. (2021) COMBO: Conservative Offline Model-Based Policy Optimization.
12. Agarwal et al. (2020) An Optimistic Perspective on Offline Reinforcement Learning, **ICML2020**.
13. Wu et al. (2021) Uncertainty Weighted Actor-Critic for Offline Reinforcement Learning, **ICML2021**.
14. Fu et al. D4RL: Datasets for Deep Data-Driven RL.
15. Jiang and Lu (2021), Offline Decentralized Multi-Agent Reinforcement Learning.

Offline RL-Step By Step

SOTA1 — DT Decision Transformer

🔥 Decision Transformer: Reinforcement Learning via Sequence Modeling 通过序列建模的强化学习

Valuable Github Repo

1.Re-implemented Offline Algorithms

https://github.com/sparkmxy/my-offlinerl