上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Institute of Media, Information, and Network

Offline Reinforcement Learning

Nuowen Kan

Shanghai Jiao Tong University

May 2024

Contents at a glance

1. Preliminary
2. Challenges of Offline RL
3. Solutions
    a) Policy constraint methods
    b) Value-regularized method
    c) Implicit constraints/regularization
    d) Model-based method
    e) Conditional imitation method

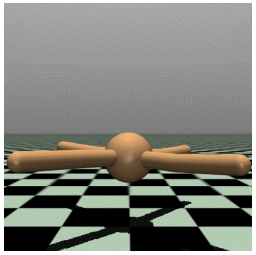
Why offline RL?

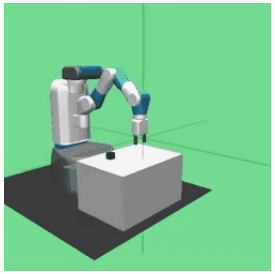
- **Success of modern machine learning**

  Data + huge network

Can we develop data-driven RL methods?

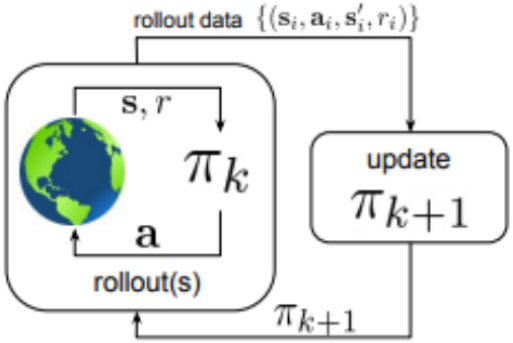
- **Training of RL**



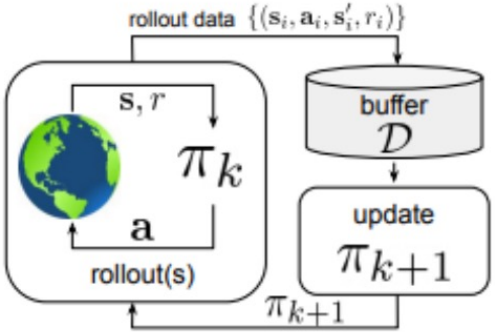



On/off-Policy? On/off-Line?

On-Policy



Off-Policy



Reinforcement Learning with Online Interactions

Online

Offline Reinforcement Learning

Offline

- **Why, or when, might offline RL be more useful?**

  leverage datasets collected by people, existing systems

  online policy collection may be risky, unsafe

  reuse previously collected data rather than recollecting

  (e.g. previous experiments, projects, robots, institutions)

Note: A blend of offline then online RL is also possible!

- **Offline RL terminology**

$\mathcal{D} = \{(s_i, a_i, s'_i, r_i)\}$ Offline dataset sampled from some unknown policy $\pi_\beta(a|s)$

$\pi_\beta(a|s)$ Behavior/sampling policy

$a \sim \pi_\beta(a|s)$

$s \sim d^{\pi_\beta}(s)$

$s' \sim p(s'|s, a)$

$r \leftarrow r(s, a)$

} Sampled trajectories

Objective $\max_\pi \sum_{t=0}^{T} E_{\mathbf{s}_t \sim d^\pi(\mathbf{s}), \mathbf{a}_t \sim \pi(\mathbf{a}|\mathbf{s})}[\gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$

- **Where does the data come from?**

  human collected data

  data from a hand-designed system / controller

  data from previous RL run(s)

  a mixture of sources

Contents at a glance

1. Preliminary
2. Challenges of Offline RL
3. Solutions
    a) Policy constraint methods
    b) Value-regularized method
    c) Implicit constraints/regularization
    d) Model-based method
    e) Conditional imitation method

Challenges in offline RL

- **Exploration is impossible**

- **Counterfactual query during offline training**

  Must act different to learn a better policy

  Against I.I.D assumption

- **Distributional shift**

  Distribution mismatch: In offline training, RL policy is trained under distribution $\mathcal{D}$, while it's evaluated on another.

  Causes: new policy is trained by maximizing the expected return, which changes in visited states during testing.

https://research.google/blog/tackling-open-challenges-in-offline-reinforcement-learning/

- **Can we just use off-policy algorithms?**

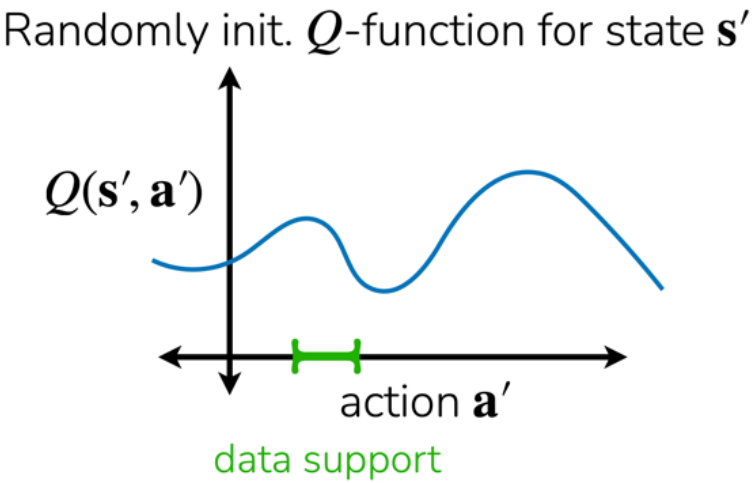
Q-learning objective

$$\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \| Q(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}')) \|^2$$

What happens if you optimize this using a static dataset?

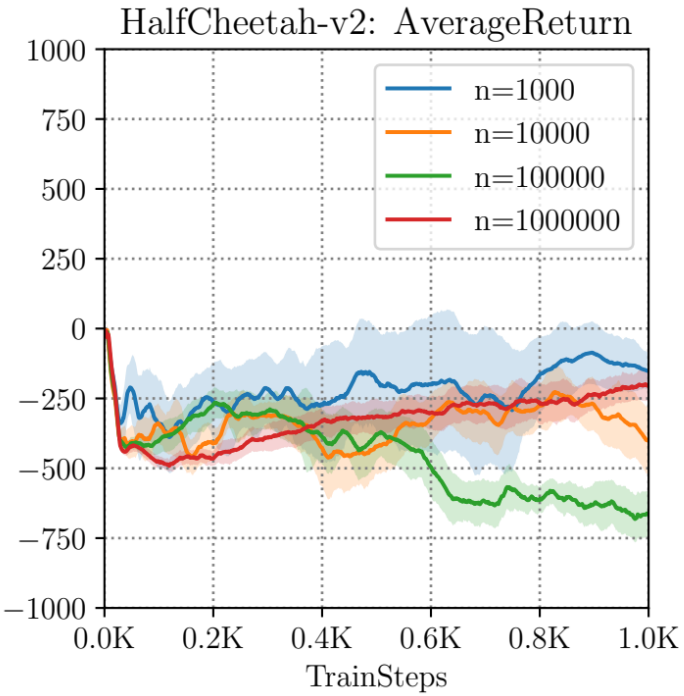
Evaluating Q on actions $\boldsymbol{a}'$ not in the dataset

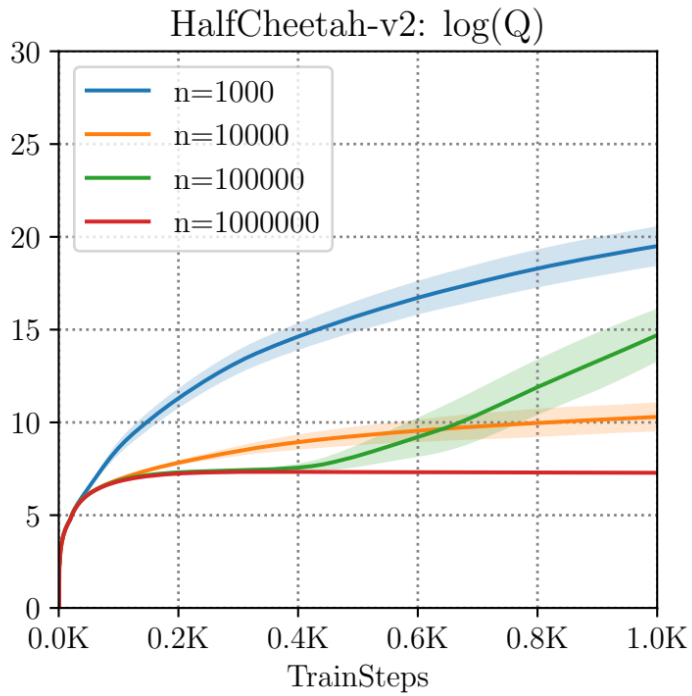
- **Suffer from distribution shift**



- Q-function will be unreliable on Out-of-distributional (OOD) actions
- **max** will seek out actions where Q-function is over-optimistic
- After values propagate, Q-values will become substantially over-estimated

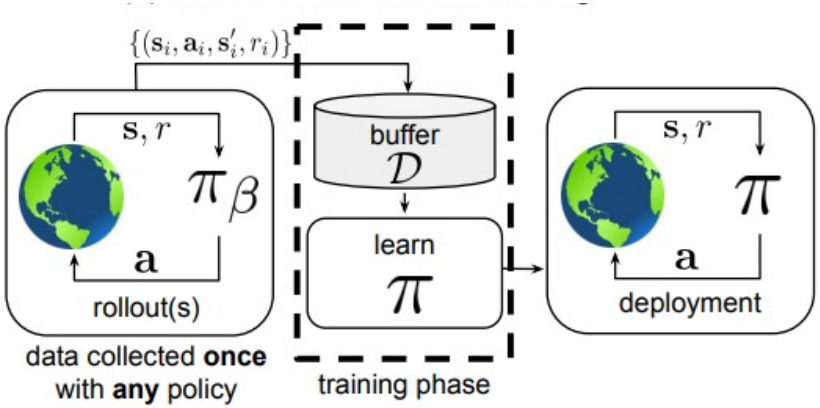
- **Suffer from distribution shift**





How to mitigate overestimation in offline RL

- **Suffer from distribution shift**



- Estimate the expectation of a different distribution

Important Sampling

$$\mathbb{E}_{X\sim P}[f(X)] = \sum P(X)f(X)$$

$$= \sum Q(X)\frac{P(X)}{Q(X)}f(X)$$

$$= \mathbb{E}_{X\sim Q}\left[\frac{P(X)}{Q(X)}f(X)\right]$$

Unbiased estimated but large variation

Contents at a glance

1. Preliminary
2. Challenges of Offline RL
3. Solutions
   a) Policy constraint methods
   b) Value-regularized method
   c) Implicit constraints/regularization
   d) Model-based method
   e) Conditional imitation method

Policy constraint

- **Explicit policy constraint methods**

$$\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \| Q(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}')) \|^2$$

An intuitive idea: let $\mathbf{a}'$ stay close to behavior policy

New objective:

$$\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \| Q(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + \gamma E_{\mathbf{a}'\sim\pi_{new}(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}')) \|^2$$

$$\pi_{new}(\mathbf{a} \mid \mathbf{s}) = \arg\max_{\pi} E_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})] \quad \text{s.t. } D_{\mathrm{KL}}(\pi \| \pi_\beta) \le \epsilon$$

Policy constraint

$$\pi_{new}(\mathbf{a} \mid \mathbf{s}) = \arg\max_{\pi} E_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})] \quad \text{s.t. } D_{\mathrm{KL}}(\pi \parallel \pi_\beta) \leq \epsilon$$

Issue: we don't know what π_β is

How do we implement constraints in RL?

- fit a policy to the data,
- *i.e.*, learn a proxy for π_β through imitation

Forms of policy constraints?

1. support constraint: $\pi(\mathbf{a}\,|\,\mathbf{s}) > 0$ only if $\pi_\beta(\mathbf{a}\,|\,\mathbf{s}) \geq \epsilon$

   + close to what we want - challenging to implement in practice

2. KL divergence: $D_{KL}(\pi \| \pi_\beta)$

   + easy to implement - not necessarily what we want

Policy constraint

Unconstrained optimization objective:

$$\theta \leftarrow \arg\max_{\theta} E_{\mathbf{s}\sim D}\left[E_{\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a}) + \lambda \log \pi_\beta(\mathbf{a}|\mathbf{s})] + \lambda\mathcal{H}(\pi(\mathbf{a}|\mathbf{s}))\right]$$

Lagrange multiplier $\lambda \geq 0$

- Since for the constraints we have

$$D_{\mathrm{KL}}(\pi\|\pi_\beta) = E_\pi[\log\pi(\mathbf{a}|\mathbf{s}) - \log\pi_\beta(\mathbf{a}|\mathbf{s})] = -E_\pi[\log\pi_\beta(\mathbf{a}|\mathbf{s})] - \mathcal{H}(\pi)$$

Modified reward function:

$$\bar{r}(s,a) = r(s,a) - D_{KL}(\pi,\pi_\beta)$$

Penalize action distribution divergence

Policy constraint

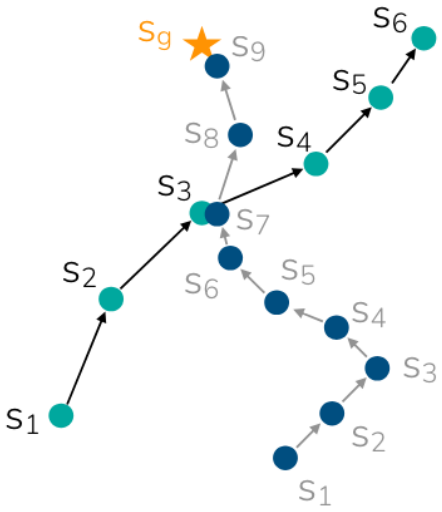
- **Like imitation learning? No!**

behavioral cloning

- Imitation methods can't outperform the expert
- Offline data may not be optimal!
- Offline RL can leverage reward information to outperform behavior policy.
- Good offline RL methods can *stitch* together good behaviors.

- Imitation learning vs. offline reinforcement learning

Good offline RL methods can ***stitch*** together good behaviors.



s_1 -> s_3 is good behavior

s_7 -> s_9 is good behavior

Offline RL methods can learn a policy that goes from s_1 to s_9!

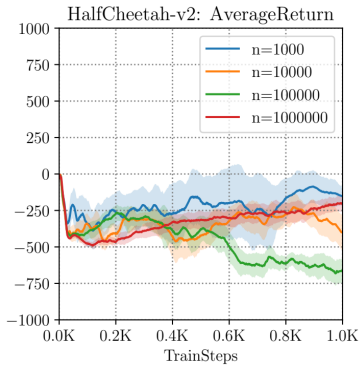
Contents at a glance

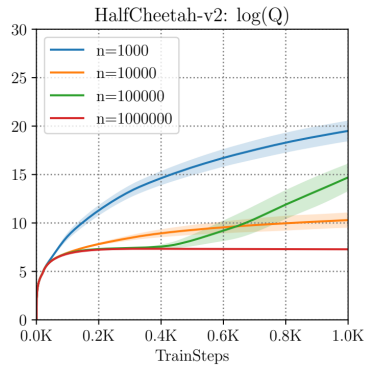
1. Preliminary
2. Challenges of Offline RL
3. Solutions
   a) Policy constraint methods
   b) Value-regularized method
   c) Implicit constraints/regularization
   d) Model-based method
   e) Conditional imitation method

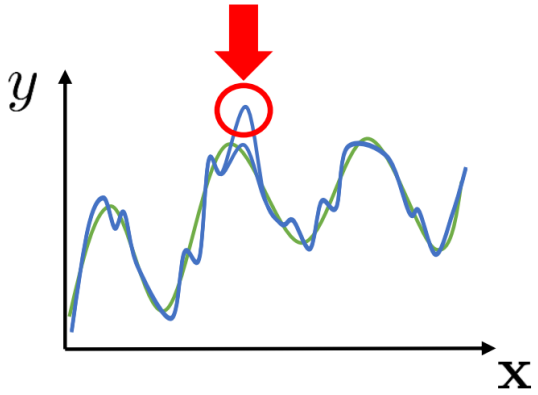
Conservative Q-learning (CQL)

- **Regularizing the Q-function directly**

  Avoid overestimation for out-of-distribution actions







General objective for conservative Q-function

$$\mathcal{E}(\mathcal{D}, Q_\phi) = \mathcal{E}(\mathcal{D}, Q_\phi) + \alpha \mathcal{C}(\mathcal{D}, Q_\phi)$$

Regular objective: Bellman error term
Computed from dataset $\mathcal{D}$

Conservative penalty term

Conservative Q-learning (CQL)

- **Regularizing the Q-function directly**

$$\hat{Q}^\pi = \arg\min_Q \max_\mu E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_\pi[Q(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right] + \alpha E_{\mathbf{s}\sim D,\mathbf{a}\sim\mu(\cdot|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]$$

standard critic update — push down on large Q-values

- Push down large Q-values
- Can show that $Q^\pi_\phi < Q^\pi$ (true Q-function) for large enough α

Kumar, Aviral, et al. "Conservative q-learning for offline reinforcement learning." *Advances in Neural Information Processing Systems* 33 (2020): 1179-1191.

Conservative Q-learning (CQL)

- **Regularizing the Q-function directly**

standard critic update

push down on large Q-values

$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\mu} E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_{\pi}[Q(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right] + \alpha E_{\mathbf{s}\sim D,\mathbf{a}\sim\mu(\cdot|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})] - \alpha E_{(\mathbf{s},\mathbf{a})\sim D}[Q(\mathbf{s},\mathbf{a})]$$

push up on Q-values for $(\mathbf{s},\mathbf{a})$ in the data

- No longer guaranteed that $Q_{\phi}^{\pi} < Q^{\pi}$ for all (s,a)
- Can show $E_{a\sim\pi}[Q_{\phi}^{\pi}(s,a)] < E_{a\sim\pi}[Q^{\pi}(s,a)]$ for all $s \in \mathcal{D}$

Kumar, Aviral, et al. "Conservative q-learning for offline reinforcement learning." *Advances in Neural Information Processing Systems* 33 (2020): 1179-1191.

- **Full algorithm of conservative Q-learning**

1. Update $\hat{Q}^{\pi}$ using L_{CQL} using D
2. Update policy π

If actions are discrete: $\pi(\mathbf{a}|\mathbf{s}) = \begin{cases} 1 \text{ if } \mathbf{a} = \arg\max_{\bar{\mathbf{a}}} \hat{Q}(\mathbf{s}, \bar{\mathbf{a}}) \\ 0 \text{ otherwise} \end{cases}$

If actions are continuous: $\theta \leftarrow \theta + \eta \nabla_{\theta} E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi_{\theta}(\cdot|\mathbf{s})} \left[\hat{Q}(\mathbf{s}, \mathbf{a}) \right]$

Contents at a glance

1. Preliminary
2. Challenges of Offline RL
3. Solutions
    a) Policy constraint methods
    b) Value-regularized method
    c) Implicit constraints/regularization
    d) Model-based method
    e) Conditional imitation method

Policy Constraint

- **Problems in explicit policy constraints**

Behavior policy is unknown

Recall: learn a proxy for π_β through imitation

Estimation error for approximating behavior policy

Not to explicitly model behavior policy

Implicit Policy Constraint

- **Expected improvement in terms of advantage func.**

$$\eta(\pi) = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[A^\mu(\mathbf{s},\mathbf{a})] = \mathbb{E}_{\mathbf{s}\sim d_\pi(\mathbf{s})}\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\mathcal{R}^\mu_{\mathbf{s},\mathbf{a}} - V^\mu(\mathbf{s})\right]$$

$$V^\mu(\mathbf{s}) = \int_a \pi_\beta(\mathbf{a} \mid \mathbf{s})\mathcal{R}^{\mathbf{a}}_{\mathbf{s}} d\mathbf{a}$$

Behavior policy

- **Objective**

$$\arg\max_\pi \quad \int_{\mathbf{s}} d_\mu(\mathbf{s}) \int_{\mathbf{a}} \pi(\mathbf{a} \mid \mathbf{s})[A^\mu] d\mathbf{a} d\mathbf{s}$$

$$\text{s.t.} \quad \int_{\mathbf{s}} d_\mu(\mathbf{s}) \mathrm{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}) \parallel \pi_\beta(\cdot \mid \mathbf{s})) d\mathbf{s} \leq \epsilon.$$

Peng, Xue Bin, et al. "Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning." (2020).

Implicit Policy Constraint

- **Lagrange function**

$$\mathcal{L}(\pi, \beta) = \int_{\mathbf{s}} d_\mu(\mathbf{s}) \int_{\mathbf{a}} (\mathbf{a} \mid \mathbf{s}) A^\mu d\mathbf{a} d\mathbf{s}\, \pi + \beta(\epsilon - \int_{\mathbf{s}} d_\mu(\mathbf{s}) \mathrm{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}) \parallel \pi_\beta(\cdot \mid \mathbf{s})) d\mathbf{s})$$

- **Optimal value**

$$\pi^\star(\mathbf{a} \mid \mathbf{s}) = \frac{1}{Z(\mathbf{s})} \pi_\beta(\mathbf{a} \mid \mathbf{s}) \exp(\frac{1}{\lambda} A^\mu(\mathbf{s}, \mathbf{a}))$$

hyperparameter

Peng, Xue Bin, et al. "Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning." (2020).

Implicit Policy Constraint

- **Learning a policy close to optimal**

$$\pi^{\star}(\mathbf{a} \mid \mathbf{s}) = \frac{1}{Z(\mathbf{s})}\pi_{\beta}(\mathbf{a} \mid \mathbf{s})\exp(\frac{1}{\lambda}A^{\mu}(\mathbf{s},\mathbf{a}))$$

$$\arg\min_{\pi} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\mathrm{D}_{\mathrm{KL}}\left(\pi^{\star}(\cdot \mid \mathbf{s}) \parallel \pi(\cdot \mid \mathbf{s})\right)\right]$$

$$\hat{\pi} \leftarrow \arg\max_{\pi} E_{\mathbf{s},\mathbf{a}\sim D}[\log \pi(\mathbf{a} \mid \mathbf{s})\exp(\frac{1}{\lambda}A^{\mu})]$$

hyperparameter

$$D_{\mathrm{KL}}(\pi\|\pi_{\beta}) = E_{\pi}[\log \pi(\mathbf{a}|\mathbf{s}) - \log \pi_{\beta}(\mathbf{a}|\mathbf{s})] = -E_{\pi}[\log \pi_{\beta}(\mathbf{a}|\mathbf{s})] - \mathcal{H}(\pi)$$

Peng, Xue Bin, et al. "Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning." (2020).

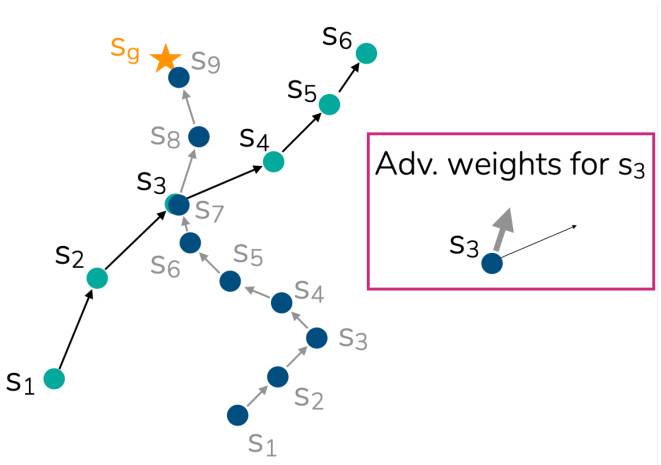
- **Advantage-weighted regression**

  Another perspective

$$\hat{\pi} \leftarrow \arg\max_{\pi} E_{\mathbf{s},\mathbf{a}\sim D}[\log \pi(\mathbf{a} \mid \mathbf{s}) \exp(\frac{1}{\lambda} A^{\mu})]$$

$$\theta \leftarrow \arg\max_{\theta} E_{\mathbf{s},\mathbf{a}\sim D}\left[\log \pi_{\theta}(\mathbf{a} \mid \mathbf{s}) \exp(A(\mathbf{s},\mathbf{a}))\right]$$

standard imitation learning with advantage weights



Peng, Xue Bin, et al. "Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning." (2020).

- **How to estimate the advantage function**

Estimate $V^{\pi_\beta}(s)$ with Monte Carlo, $\min_V E_{(\mathbf{s},\mathbf{a})\sim D}\left[\left(R_{\mathbf{s},\mathbf{a}} - V(\mathbf{s})\right)^2\right]$

Approximate $\hat{A}^{\pi_\beta}(\mathbf{s},\mathbf{a}) = R_{\mathbf{s},\mathbf{a}} - V(\mathbf{s})$

empirical return

+ Simple

+ Avoids querying or training on any OOD actions!

- Monte Carlo estimation is noisy

- $\hat{A}^{\pi_\beta}$ assumes weaker policy than $\hat{A}^{\pi_\theta}$

- **Advantage weighted actor critic**

Estimate advantage function with TD updates instead of Monte Carlo?

1. Estimate Q^π-function: $\min_Q E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r + \gamma E_{\mathbf{a}'\sim\pi(\cdot|\mathbf{s})}[Q(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right]$

2. Estimate advantage as: $\hat{A}^\pi(\mathbf{s},\mathbf{a}) = \hat{Q}^\pi(\mathbf{s},\mathbf{a}) - E_{\bar{\mathbf{a}}\sim\pi(\cdot|\mathbf{s})}[\hat{Q}^\pi(\mathbf{s},\bar{\mathbf{a}})]$

3. Update policy as before: $\hat{\pi} \leftarrow \arg\max_\pi E_{\mathbf{s},\mathbf{a}\sim D}\left[\log \pi(\mathbf{a}|\mathbf{s}) \exp\left(\frac{1}{\alpha}\hat{A}^\pi(\mathbf{s},\mathbf{a})\right)\right]$

+ Policy still only trained on actions in data.
+ Temporal difference updates instead of Monte Carlo.

What might go wrong?

- Possibly querying OOD actions!

Nair, Gupta, Dalal, Levine., AWAC: Accelerating Online Reinforcement Learning with Offline Datasets

- **Can we estimate Q without querying OOD actions?**

AWAC: Estimate Q-function: $\min_Q E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r + \gamma E_{\substack{\mathbf{a}'\sim\pi(\cdot|\mathbf{s}) \\ \mathbf{a}'\sim D}}[Q(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right]$

"SARSA algorithm"

Evaluate the Q-function only on the (s, a) in the offline dataset

The core idea of Implicit Q-Learning

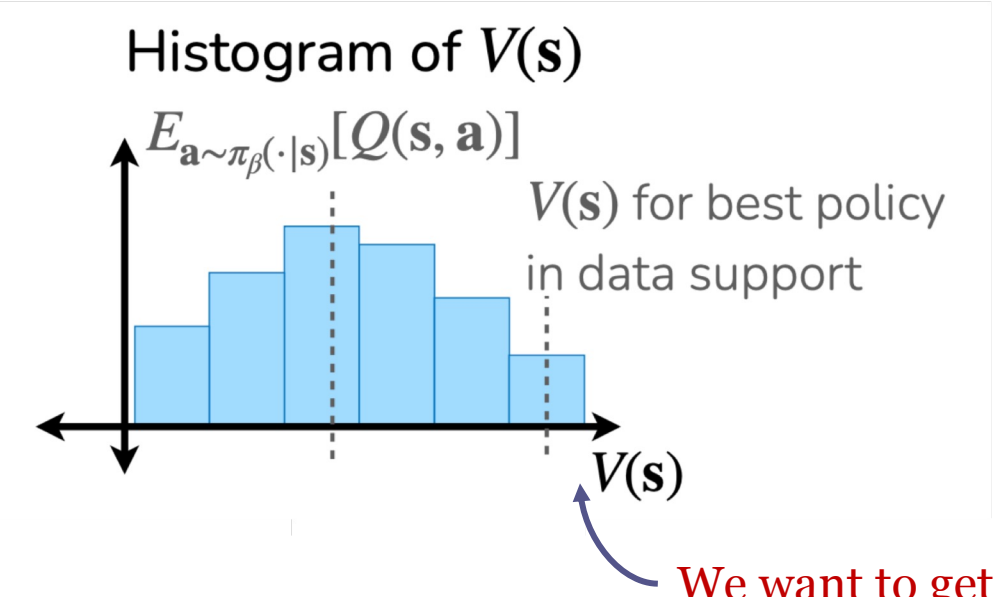
Kostrikov, Nair, Levine. Implicit Q-Learning. ICLR'22

- **Can we estimate Q for a policy $\pi'(\mathbf{a} \mid \mathbf{s})$ that is better than $\pi_\beta(\mathbf{a} \mid \mathbf{s})$?**

  We don't need to explicitly modeling $\pi'(a \mid s)$

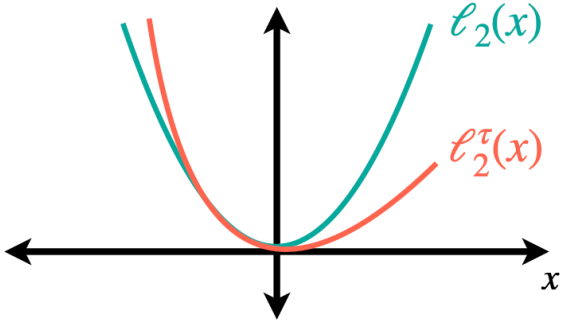
SARSA update: $\hat{Q}^{\pi_\beta} \leftarrow \arg\min_Q E_{(\mathbf{s},\mathbf{a},\mathbf{s}',\mathbf{a}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r + \gamma Q(\mathbf{s}',\mathbf{a}')\right)\right)^2\right]$

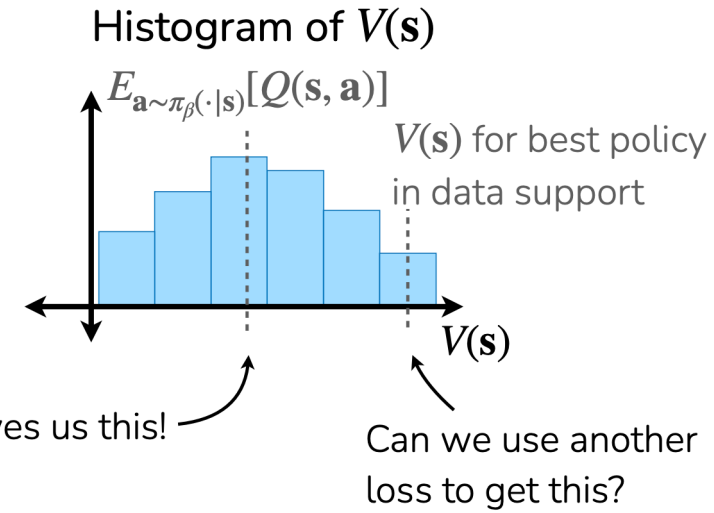
a sample of $V^{\pi_\beta}(\mathbf{s}')$



Kostrikov, Nair, Levine. Implicit Q-Learning. ICLR'22

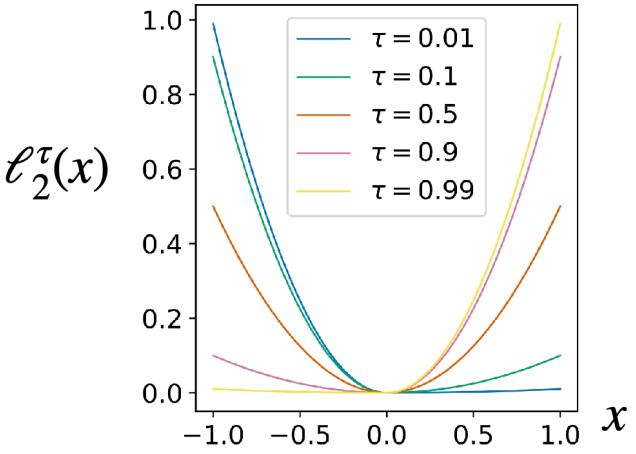
- **Key idea: Expectile regression**





Expectile regression loss:

$$\ell_2^\tau(x) = \begin{cases} (1-\tau)x^2 & \text{if } x < 0 \\ \tau x^2 & \text{otherwise} \end{cases}$$



Kostrikov, Nair, Levine. Implicit Q-Learning. ICLR'22

- **Implicit Q-Learning**

$$L(\theta) = \mathbb{E}_{(s,a,s',a')\sim\mathcal{D}}\left[l_2^\tau\left(r(s,a) + \gamma Q_{\hat{\theta}}(s',a') - Q_\theta(s,a)\right)\right]$$

Expectile regression loss:

$$\ell_2^\tau(x) = \begin{cases} (1-\tau)x^2 & \text{if } x < 0 \\ \tau x^2 & \text{otherwise} \end{cases}$$

A large $Q_\theta(s,a)$ → a good action a

a "lucky" sample that happened to have transitioned into a good state.

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\psi(s))].$$

$$L_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a))^2].$$

• Implicit Q-Learning

Full algorithm

Fit V with expectile loss: $\hat{V}(\mathbf{s}) \leftarrow \arg\min_V E_{(\mathbf{s},\mathbf{a})\sim D}\left[\ell_2^\tau\left(V(\mathbf{s}) - \hat{Q}(\mathbf{s},\mathbf{a})\right)\right]$ using small $\tau < 0.5$

Update Q with typical MSE loss: $\hat{Q}(\mathbf{s},\mathbf{a}) \leftarrow \arg\min_Q E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r + \gamma\hat{V}(\mathbf{s}')\right)\right)^2\right]$

Extract policy with AWR: $\hat{\pi} \leftarrow \arg\max_\pi E_{\mathbf{s},\mathbf{a}\sim D}\left[\log \pi(\mathbf{a}\,|\,\mathbf{s})\exp\left(\frac{1}{\alpha}\left(\hat{Q}(\mathbf{s},\mathbf{a}) - \hat{V}(\mathbf{s})\right)\right)\right]$

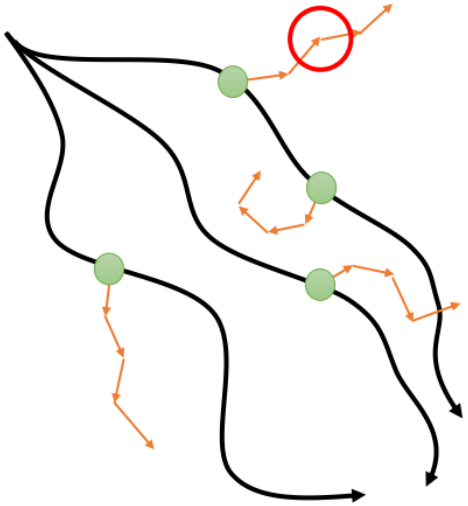
+ Never need to query OOD actions!
+ Policy (still) only trained on actions in data.
+ Decoupling actor & critic training —> computationally fast

Contents at a glance

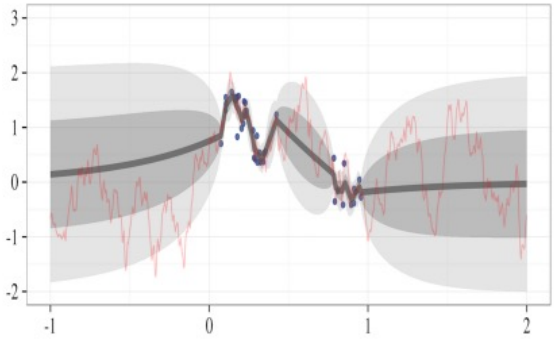
1. Preliminary
2. Challenges of Offline RL
3. Solutions
   a) Policy constraint methods
   b) Value-regularized method
   c) Implicit constraints/regularization
   d) Model-based method
   e) Conditional imitation method

Model-Based Offline RL

- **Advantage of model-based RL under offline setting**
  - Supervised learning is more stable than boostrapping RL training
  - Learned environment model can provide uncertainty estimation



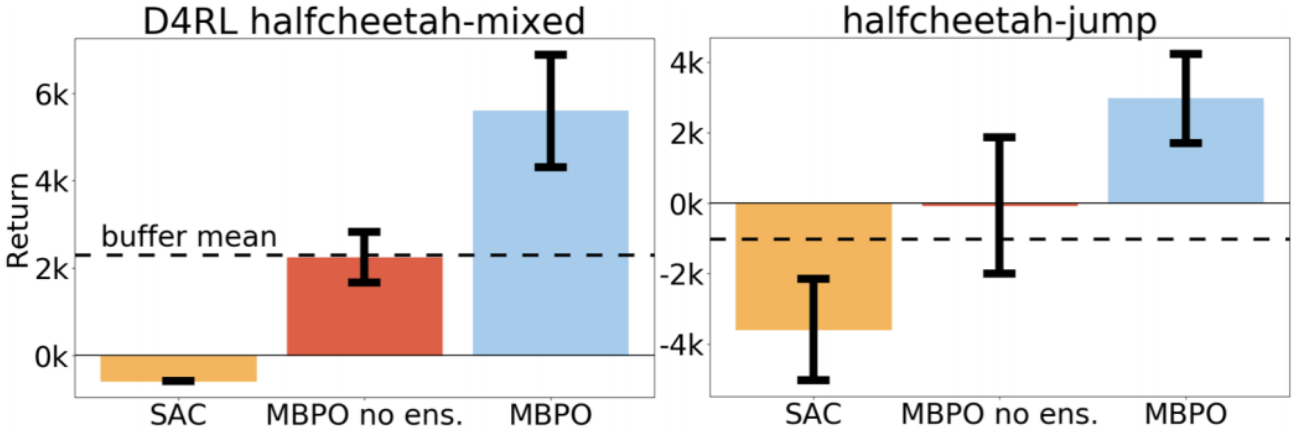
Query of OOD states can be measured by uncertainty



Gaussian process regression

Model-Based Offline RL

- **Model-based RL is well-suited for offline RL**



Algorithm 1 Framework for Model-based Offline Policy Optimization (MOPO) with Reward Penalty

Require: Dynamics model $\widehat{T}$ with admissible error estimator $u(s,a)$; constant λ.

1: Define $\tilde{r}(s,a) = r(s,a) - \lambda u(s,a)$. Let $\widetilde{M}$ be the MDP with dynamics $\widehat{T}$ and reward $\tilde{r}$.

2: Run any RL algorithm on $\widetilde{M}$ until convergence to obtain $\hat{\pi} = \text{argmax}_{\pi} \eta_{\widetilde{M}}(\pi)$

Yu, Tianhe, et al. "Mopo: Model-based offline policy optimization." *Advances in Neural Information Processing Systems* 33 (2020): 14129-14142.

MOPO: Reward penalty when model is uncertain

- Key idea: build a lower bound for the expected return of a policy π under the true dynamics and then maximize the lower bound over π

Lemma 4.1 (Telescoping lemma). *Let M and $\widehat{M}$ be two MDPs with the same reward function r, but different dynamics T and $\widehat{T}$ respectively. Let* $G^{\pi}_{\widehat{M}}(s,a) := \mathop{\mathbb{E}}_{s'\sim\widehat{T}(s,a)}[V^{\pi}_{M}(s')] - \mathop{\mathbb{E}}_{s'\sim T(s,a)}[V^{\pi}_{M}(s')]$.

Then,

$$\eta_{\widehat{M}}(\pi) - \eta_{M}(\pi) = \gamma \mathop{\bar{\mathbb{E}}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}\left[G^{\pi}_{\widehat{M}}(s,a)\right] \tag{1}$$

As an immediate corollary, we have

$$\eta_{M}(\pi) = \mathop{\bar{\mathbb{E}}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}\left[r(s,a) - \gamma G^{\pi}_{\widehat{M}}(s,a)\right] \geq \mathop{\bar{\mathbb{E}}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}\left[r(s,a) - \gamma |G^{\pi}_{\widehat{M}}(s,a)|\right] \tag{2}$$

- Measure the difference between two MDPs

$$|G^{\pi}_{\widehat{M}}(s,a)| \leq \sup_{f\in\mathcal{F}}\left|\mathop{\mathbb{E}}_{s'\sim\widehat{T}(s,a)}[f(s')] - \mathop{\mathbb{E}}_{s'\sim T(s,a)}[f(s')]\right| =: d_{\mathcal{F}}(\widehat{T}(s,a), T(s,a)),$$

$$\eta_{M}(\pi) \geq \mathop{\bar{\mathbb{E}}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}\left[r(s,a) - \gamma |G^{\pi}_{\widehat{M}}(s,a)|\right] \geq \mathop{\bar{\mathbb{E}}}_{(s,a)\sim\rho^{\pi}_{\widehat{T}}}[r(s,a) - \lambda u(s,a)]$$

where $d_{\mathcal{F}}(\hat{T}(s,a), T(s,a)) \leq u(s,a)$

- T. Yu, T. Ma, et.al. MOPO: Model-based offline policy optimization. NeurIPS, 2020.

COMBO: Conservative Q value w.r.t OOD states and actions

- Model-free conservative Q-learning
  - Policy Evaluation

$$Q^{k+1} \leftarrow \arg\min_Q \beta \left(\mathbb{E}_{\mathbf{s}\sim\mathcal{D},\mathbf{a}\sim\mu(\cdot|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[Q(\mathbf{s},\mathbf{a})\right]\right) + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \widehat{\mathcal{B}}^\pi Q^k(\mathbf{s},\mathbf{a})\right)^2\right], \quad (3)$$

  - Policy improvement

$$\pi \leftarrow \arg\max_{\pi'} \mathbb{E}_{\mathbf{s}\sim\mathcal{D},\mathbf{a}\sim\pi'(\cdot|\mathbf{s})}\left[\hat{Q}^\pi(\mathbf{s},\mathbf{a})\right]$$

$\mathcal{D}$ is offline dataset, μ is a sampling distribution: $\mu(a,s) = d^{\pi_\beta}(s)\mu(a|s)$

We have $\mathbb{E}_{\pi(a|s)}[\hat{Q}^\pi(s,a)] \leq V^\pi(s)$ when $\mu(a|s) = \pi(a|s)$

- Model-based
  - Policy Evaluation

$$\hat{Q}^{k+1} \leftarrow \arg\min_Q \beta \left(\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\rho(\mathbf{s},\mathbf{a})}\left[Q(\mathbf{s},\mathbf{a})\right] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[Q(\mathbf{s},\mathbf{a})\right]\right) + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim d_f}\left[\left(Q(\mathbf{s},\mathbf{a}) - \widehat{\mathcal{B}}^\pi \hat{Q}^k(\mathbf{s},\mathbf{a})\right)^2\right]. \quad (4)$$

$$\rho(\mathbf{s},\mathbf{a}) = d^\pi_{\widehat{\mathcal{M}}}(\mathbf{s})\pi(\mathbf{a}|\mathbf{s})$$

$$d^\mu_f(\mathbf{s},\mathbf{a}) := f\, d(\mathbf{s},\mathbf{a}) + (1-f)\, d^\mu_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a}),$$

ρ is a sampling distribution over learned model $\widehat{\mathcal{M}}$.

Tianhe Yu, and Chelsea Finn et al. Combo: Conservative offline model-based policy optimization. arXiv preprint arXiv:2102.08363,2021.

COMBO: Conservative Q value w.r.t OOD states and actions

- Policy Evaluation

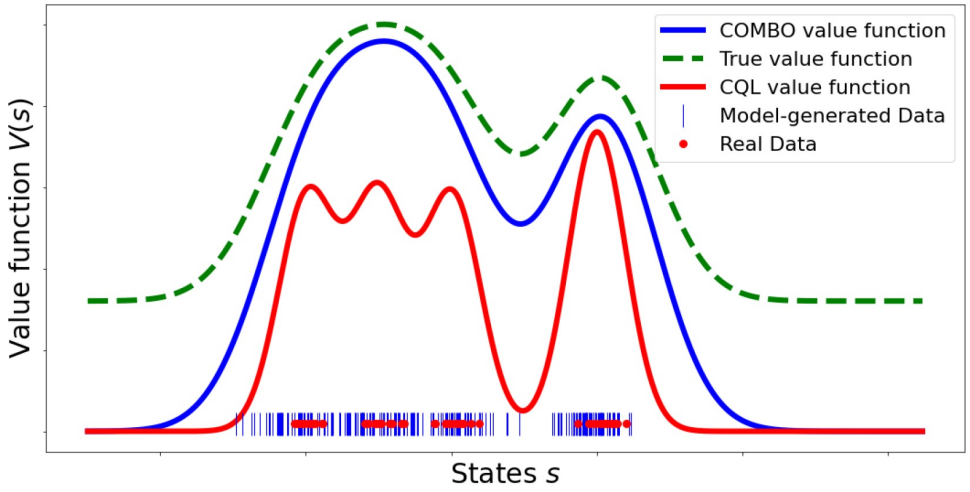
$$\rho(\mathbf{s},\mathbf{a}) = d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})\pi(\mathbf{a}|\mathbf{s})$$

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

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

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

ρ is a sampling distribution over learned model $\widehat{\mathcal{M}}$.



- Aviral Kumar, and Sergey Levine, et al.. Conservative q-learning for offline reinforcement learning. arXiv preprint arXiv:2006.04779, 2020.
- Tianhe Yu, and Chelsea Finn et al. Combo: Conservative offline model-based policy optimization. arXiv preprint arXiv:2102.08363,2021.

Contents at a glance

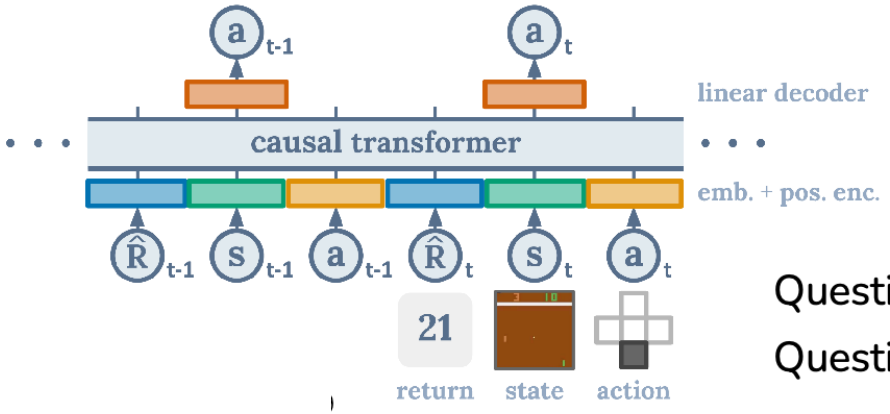
1. Preliminary
2. Challenges of Offline RL
3. Solutions
   a) Policy constraint methods
   b) Value-regularized method
   c) Implicit constraints/regularization
   d) Model-based method
   e) Conditional imitation method

Decision Transformer

1. Imitate entire dataset: $\max_{\pi} \sum_{(\mathbf{s},\mathbf{a})\in D} \log \pi(\mathbf{a} \mid \mathbf{s}, R_{\mathbf{s},\mathbf{a}})$

   Condition policy on (empirical) return to go.

- Policy will learn to mimic good and poor behaviors (and everything in between!)
- Pass in high return at test time
- Can use a sequence model:



Referred to as: upside-down RL, reward-conditioned policies, decision transformers

Question: Can this approach do data stitching?

Question: When would a sequence model be helpful?

Lili Chen and Kevin Lu, "Decision Transformer: Reinforcement Learning via Sequence Modeling," 2021

Diffusion model for Offline RL

- **Diffusion model**

  Forward process, add noise: original image → Gaussian noise

$$q(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) := \mathcal{N}(\boldsymbol{x}_{k+1}; \sqrt{\alpha_k}\boldsymbol{x}_k, (1-\alpha_k)\boldsymbol{I})$$

  Reverse process, denoise (generation): Gaussian noise→ image

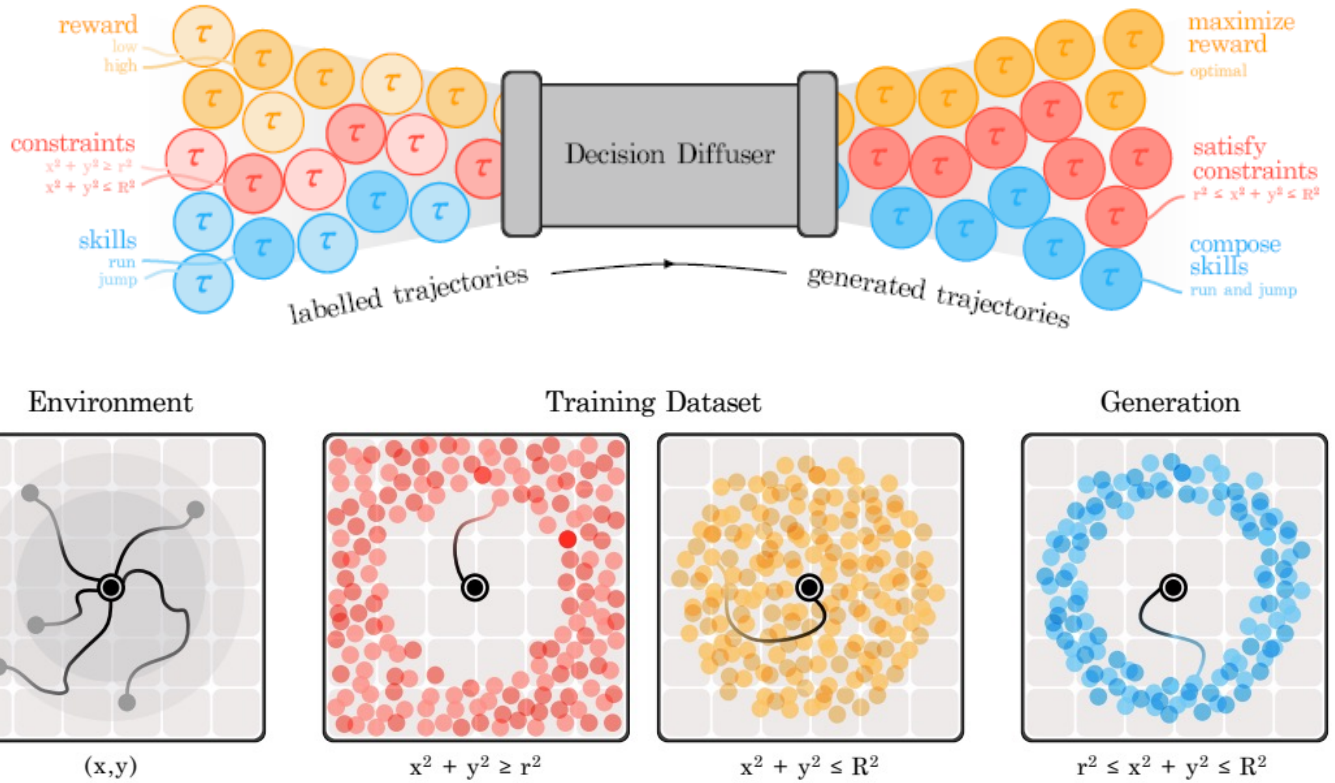
$$p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k) := \mathcal{N}(\boldsymbol{x}_{k-1}|\mu_\theta(\boldsymbol{x}_k, k), \Sigma_k)$$

  Guided diffusion: modeling distribution $q(x|y)$ for generating samples with attributes of label y

[1] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in neural information processing systems, 2020, 33: 6840-6851.

Diffusion model for Offline RL

- **Decision Making using Conditional Generative Modeling**



Ajay, Anurag, et al. "Is Conditional Generative Modeling all you need for Decision Making?." The Eleventh International Conference on Learning Representations. 2022.

Q & A

Many Thanks