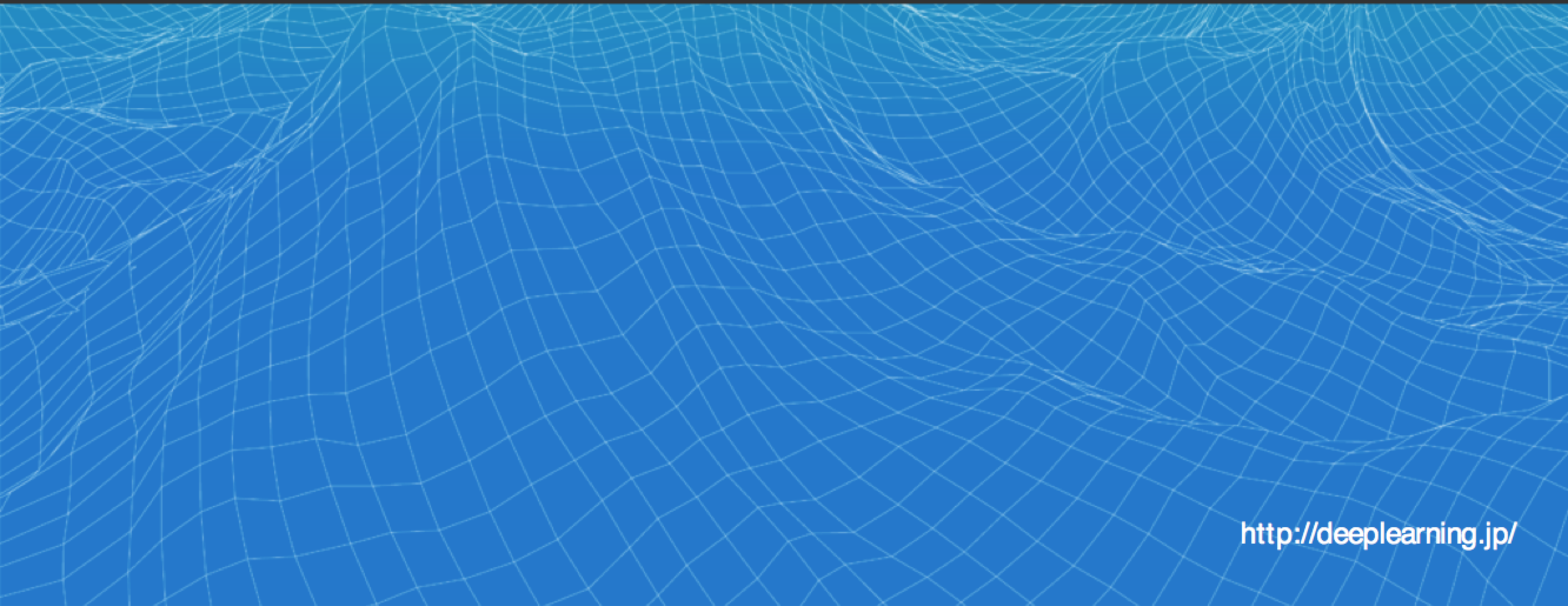
DEEP LEARNING JP
[DL Papers]

近年のオフライン強化学習のまとめ

—Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems—

2020.06.26　Presenter: Tatsuya Matsushima @__tmats__ , Matsuo Lab



本発表について

オフライン強化学習（RL）問題

- なんらかの方法によって集められたデータセットから，追加的な環境との相互作用をせずに方策を学習

モチベーション

- 応用先によっては実環境での探索が現実的ではない問題に利用したい
  - 例）ヘルスケア，対話エージェント，実ロボット…

よく使われるアプローチ

- 問題設定的にはoff-policyのRLの手法なら基本的になんでも使える

  but 「分布シフト（distributional shift）」（後述）の問題によりうまくいかない
  - なので，それに対処する手法が最近たくさん出てきている

書誌情報

Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

- Sergey Levine, Aviral Kumar, George Tucker, Justin Fu
  - UCバークレー・Google Brain勢（両者とも最近めっちゃ研究してる）
  - https://arxiv.org/abs/2005.01643 (Submitted on 4 May 2020)
- Sergey Levine先生のサーベイ論文シリーズ続編
  - 前回 Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review
    - https://arxiv.org/abs/1805.00909
- **近年のオフラインRLの研究の動向をまとめたサーベイ論文**
  - このサーベイ論文以降に公開されカバーされてない論文も多い（特にモデルベース）
- この論文関連の研究を（うすく）カバーした発表をします
  - RLArchでもうすこし詳しく発表するかも？（相談中）

論文の構成

1.イントロ

2.オフラインRLの問題設定と概観

3.オフライン方策評価と重点サンプリングに基づくRL

- （松尾研的に）あんまりやってないがめっちゃ流行ってる

4.動的計画法（DP）に基づくオフラインRL

5.オフラインモデルベースRL

- （松尾研的に）結構やってるし取り掛かりやすい

6.応用と評価

7.議論と展望

※本サーベイ論文で登場しない論文は*をつけて表記

表記とRLの前提知識

表記

MDP $\mathscr{M} = \left(\mathcal{S}, \mathcal{A}, T, d_0, r, \gamma\right)$

- **状態** $\mathbf{s} \in \mathcal{S}$
- **行動** $\mathbf{a} \in \mathcal{A}$
- **状態遷移確率** $T(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$,
- **初期状態分布** $d_0(\mathbf{s}_0)$
- **報酬** $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- **割引率** $\gamma \in (0,1]$
  - POMDPはとりあえず考えない
    （拡張はできる）

方策 $\pi(\mathbf{a}_t \mid \mathbf{s}_t)$

軌道 $\tau = \left(\mathbf{s}_0, \mathbf{a}_0, \cdots, \mathbf{s}_H, \mathbf{a}_H\right)$

- Horizon H （無限大にもなり得る）

軌道分布

$$p_\pi(\tau) = d_0(s_0) \prod_{t=0}^{H} \pi(\mathbf{a}_t \mid \mathbf{s}_t) T(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$$

目的関数 $J(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{H} \gamma^t r\left(\mathbf{s}_t, \mathbf{a}_t\right) \right]$

時刻 t **での状態** $\mathbf{s}$ **の状態分布** $d_t(\mathbf{s}_t)$

- 軌道分布を周辺化すれば計算できる

状態 $\mathbf{s}$ **の状態分布** $d(\mathbf{s})$

- $d_t(\mathbf{s}_t)$を時刻に関して平均して計算

モンテカルロ法（REINFORCE）

パラメータ化された方策 $\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$ の学習

- 方策勾配を計算し方策を改善

$$\nabla_\theta J\left(\pi_\theta\right) = \mathbb{E}_{\tau \sim p_{\pi_\theta}(\tau)}\left[\sum_{t=0}^{H} \gamma^t \nabla_\theta \log \pi_\theta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)\left(\sum_{t'=t}^{H} \gamma^{t'-t} r\left(\mathbf{s}_{t'}, \mathbf{a}_{t'}\right) - b\left(\mathbf{s}_t\right)\right)\right]$$

ベースライン

アドバンテージの推定値 $\hat{A}(\mathbf{s}_t, \mathbf{a}_t)$

$$= \sum_{t=0}^{H} \mathbb{E}_{\mathbf{s}_t \sim d_t^\pi(\mathbf{s}_t), \mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)}\left[\gamma^t \nabla_\theta \log \pi_\theta\left(\mathbf{a}_t \mid \mathbf{s}_t\right) \hat{A}\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$$

Algorithm 1 On-policy policy gradient with Monte Carlo estimator

1: initialize θ_0
2: **for** iteration $k \in [0, \ldots, K]$ **do**
3: sample trajectories $\{\tau_i\}$ by running $\pi_{\theta_k}(\mathbf{a}_t|\mathbf{s}_t)$ ▷ each τ_i consists of $\mathbf{s}_{i,0}, \mathbf{a}_{i,0}, \ldots, \mathbf{s}_{i,H}, \mathbf{a}_{i,H}$
4: compute $\mathcal{R}_{i,t} = \sum_{t'=t}^{H} \gamma^{t'-t} r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$
5: fit $b(\mathbf{s}_t)$ to $\{\mathcal{R}i, t\}$ ▷ use constant $b_t = \frac{1}{N}\sum_i \mathcal{R}i, t$, or fit $b(\mathbf{s}_t)$ to $\{\mathcal{R}i, t\}$
6: compute $\hat{A}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) = \mathcal{R}_{i,t} - b(\mathbf{s}_t)$
7: estimate $\nabla_{\theta_k} J(\pi_{\theta_k}) \approx \sum_{i,t} \nabla_{\theta_k} \log \pi_{\theta_k}(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \hat{A}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$
8: update parameters: $\theta_{k+1} \leftarrow \theta_k + \alpha \nabla_{\theta_k} J(\pi_{\theta_k})$
9: **end for**

（近似）動的計画法（DP）（Q-learning）

価値関数を用いて方策を表現

- 状態価値 V^{π}，行動価値（Q値）Q^{π}

$$V^{\pi}\left(\mathbf{s}_t\right) = \mathbb{E}_{\tau \sim p_{\pi}\left(\tau \mid \mathbf{s}_t\right)}\left[\sum_{t'=t}^{H} \gamma^{t'-t} r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$$

$$Q^{\pi}\left(\mathbf{s}_t, \mathbf{a}_t\right) = \mathbb{E}_{\tau \sim p_{\pi}\left(\tau \mid \mathbf{s}_t, \mathbf{a}_t\right)}\left[\sum_{t'=t}^{H} \gamma^{t'-t} r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$$

Q関数の再帰表現

$$Q^{\pi}\left(\mathbf{s}_t, \mathbf{a}_t\right) = r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim T\left(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t\right), \mathbf{a}_{t+1} \sim \pi\left(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}\right)}\left[Q^{\pi}\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}\right)\right]$$

- ベルマンオペレータ$\mathscr{B}^{\pi}$を用いて

$$\overrightarrow{Q}^{\pi} = \mathscr{B}^{\pi} \overrightarrow{Q}^{\pi}$$

Q-learning

$$Q^{\star}\left(\mathbf{s}_t, \mathbf{a}_t\right) = r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim T\left(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t\right)}\left[\max_{\mathbf{a}_{t+1}} Q^{\star}\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}\right)\right]$$

- ベルマン最適オペレータ$\mathscr{B}^{*}$を用いて

$$\overrightarrow{Q}^{\pi} = \mathscr{B}^{*} \overrightarrow{Q}^{\pi}$$

Algorithm 2 Generic Q-learning (includes FQI and DQN as special cases)

initialize ϕ_0
initialize $\pi_0(\mathbf{a}|\mathbf{s}) = \epsilon \mathcal{U}(\mathbf{a}) + (1-\epsilon)\delta(\mathbf{a} = \arg\max_{\mathbf{a}} Q_{\phi_0}(\mathbf{s}, \mathbf{a}))$ ▷ Use ϵ-greedy exploration
initialize replay buffer $\mathcal{D} = \emptyset$ as a ring buffer of fixed size
initialize $\mathbf{s} \sim d_0(\mathbf{s})$
for iteration $k \in [0, \ldots, K]$ **do**
  for step $s \in [0, \ldots, S-1]$ **do**
    $\mathbf{a} \sim \pi_k(\mathbf{a}|\mathbf{s})$ ▷ sample action from exploration policy
    $\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ ▷ sample next state from MDP
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}, \mathbf{a}, \mathbf{s}', r(\mathbf{s}, \mathbf{a}))\}$ ▷ append to buffer, purging old data if buffer too big
  end for
  $\phi_{k,0} \leftarrow \phi_k$
  for gradient step $g \in [0, \ldots, G-1]$ **do**
    sample batch $B \subset \mathcal{D}$ ▷ $B = \{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_t)\}$
    estimate error $\mathcal{E}(B, \phi_{k,g}) = \sum_i \left(Q_{\phi_{k,g}} - (r_i + \gamma \max_{\mathbf{a}'} Q_{\phi_k}(\mathbf{s}', \mathbf{a}'))\right)^2$
    update parameters: $\phi_{k,g+1} \leftarrow \phi_{k,g} - \alpha \nabla_{\phi_{k,g}} \mathcal{E}(B, \phi_{k,g})$
  end for
  $\phi_{k+1} \leftarrow \phi_{k,G}$ ▷ update parameters
end for

Actor-Critic

方策と価値関数の両方をパラメータ化

- 現在の方策 $\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$ のもとで

  Q関数の学習

$$Q^\pi\left(\mathbf{s}_t, \mathbf{a}_t\right) = r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim T\left(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t\right), \mathbf{a}_{t+1} \sim \pi_\theta\left(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}\right)}\left[Q^\pi\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}\right)\right]$$

- 方策評価と方策改善の2段階からなる
  - 方策評価：方策を固定しQ値を評価
  - 方策改善：Q値を最大化する行動を
    選択するように方策を更新

Algorithm 3 Generic off-policy actor-critic

initialize ϕ_0
initialize θ_0
initialize replay buffer $\mathcal{D} = \emptyset$ as a ring buffer of fixed size
initialize $\mathbf{s} \sim d_0(\mathbf{s})$
for iteration $k \in [0, \ldots, K]$ **do**
  for step $s \in [0, \ldots, S-1]$ **do**
    $\mathbf{a} \sim \pi_{\theta_k}(\mathbf{a}|\mathbf{s})$ ▷ sample action from current policy
    $\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ ▷ sample next state from MDP
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}, \mathbf{a}, \mathbf{s}', r(\mathbf{s}, \mathbf{a}))\}$ ▷ append to buffer, purging old data if buffer too big
  end for
  $\phi_{k,0} \leftarrow \phi_k$
  for gradient step $g \in [0, \ldots, G_Q - 1]$ **do**
    sample batch $B \subset \mathcal{D}$ ▷ $B = \{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_t)\}$
    estimate error $\mathcal{E}(B, \phi_{k,g}) = \sum_i \left(Q_{\phi_{k,g}} - (r_i + \gamma \mathbb{E}_{\mathbf{a}' \sim \pi_k(\mathbf{a}'|\mathbf{s}')} Q_{\phi_k}(\mathbf{s}', \mathbf{a}'))\right)^2$
    update parameters: $\phi_{k,g+1} \leftarrow \phi_{k,g} - \alpha_Q \nabla_{\phi_{k,g}} \mathcal{E}(B, \phi_{k,g})$
  end for
  $\phi_{k+1} \leftarrow \phi_{k,G_Q}$ ▷ update Q-function parameters
  $\theta_{k,0} \leftarrow \theta_k$
  for gradient step $g \in [0, \ldots G_\pi - 1]$ **do**
    sample batch of states $\{\mathbf{s}_i\}$ from $\mathcal{D}$
    for each $\mathbf{s}_i$, sample $\mathbf{a}_i \sim \pi_{\theta_{k,g}}(\mathbf{a}|\mathbf{s}_i)$ ▷ do not use actions in the buffer!
    for each $(\mathbf{s}_i, \mathbf{a}_i)$, compute $\hat{A}(\mathbf{s}_i, \mathbf{a}_i) = Q_{\phi_{k+1}}(\mathbf{s}_i, \mathbf{a}_i) - \mathbb{E}_{\mathbf{a} \sim \pi_{k,g}(\mathbf{a}|\mathbf{s}_i)}[Q_{\phi_{k+1}}(\mathbf{s}_i, \mathbf{a})]$
    $\nabla_{\theta_{k,g}} J(\pi_{\theta_{k,g}}) \approx \frac{1}{N} \nabla_{\theta_{k,g}} \log \pi_{\theta_{k,g}}(\mathbf{s}_i, \mathbf{a}_i) \hat{A}(\mathbf{s}_i, \mathbf{a}_i)$
    $\theta_{k,g+1} \leftarrow \theta_{k,g} + \alpha_\pi \nabla_{\theta_{k,g}} J(\pi_{\theta_{k,g}})$
  end for
  $\theta k + 1 \leftarrow \theta_{k,G_\pi}$ ▷ update policy parameters
end for

On-policy vs Off-policy

On-policy

- 学習時に制御に用いた方策と同じ方策で価値を推定
  - 例）SARSA, （素の）モンテカルロ法

Off-policy

- 学習時の制御に用いる方策（挙動方策）と,
  評価され改善される方策（推定方策・ターゲット方策）が異なる
  - 例）Q-learning, experience replayを用いる手法

2.オフラインRLの問題設定と概観

オフラインRLの問題設定

固定データセット $\mathscr{D} = \left\{(\mathbf{s}_t^i, \mathbf{a}_t^i, \mathbf{s}_{t+1}^i, r_t^i)\right\}$ から方策を学習する前提をおく

- $(\mathbf{s}, \mathbf{a}) \in \mathscr{D}$ は，$\mathbf{s} \sim d^{\pi_\beta}(\mathbf{s}), \mathbf{a} \sim \pi_\beta(\mathbf{a}|\mathbf{s})$ からサンプルされていると仮定
  - π_β を**挙動方策**という

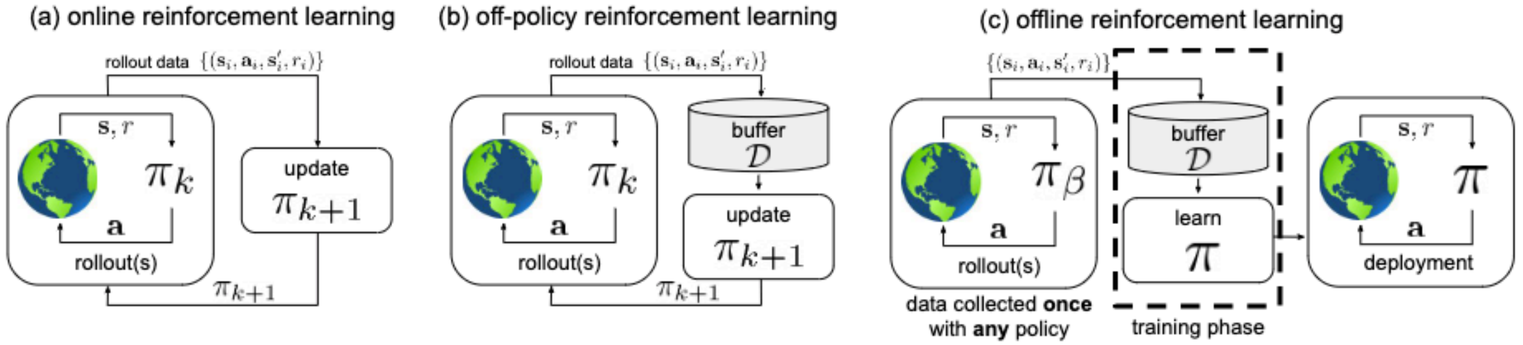


Figure 1: Pictorial illustration of classic online reinforcement learning (a), classic off-policy reinforcement learning (b), and offline reinforcement learning (c). In online reinforcement learning (a), the policy π_k is updated with streaming data collected by π_k itself. In the classic off-policy setting (b), the agent's experience is appended to a data buffer (also called a replay buffer) $\mathcal{D}$, and each new policy π_k collects additional data, such that $\mathcal{D}$ is composed of samples from $\pi_0, \pi_1, \ldots, \pi_k$, and all of this data is used to train an updated new policy π_{k+1}. In contrast, offline reinforcement learning employs a dataset $\mathcal{D}$ collected by some (potentially unknown) behavior policy π_β. The dataset is collected once, and is not altered during training, which makes it feasible to use large previous collected datasets. The training process does not interact with the MDP at all, and the policy is only deployed after being fully trained.

問題設定の呼称の混乱 [Lange+12]

論文ごとにオフラインRLの設定の呼称が違う

- 「**batch RL**」でオフラインRLを指すこともある
  - ミニバッチ学習するRL（DQNなど）との混乱
    - これらはオンラインRL
  - 混同を避けるために「**pure-batch RL**」と呼ばれることもある
- 「**fully off-policy**」と表記されることもある
  - 追加のデータ収集をしない意味で「fully」
- ややこしいので本論文・本発表では「**オフラインRL**」で統一
  - 最近（去年ぐらいから）の論文は「オフライン」の語を使っているイメージ

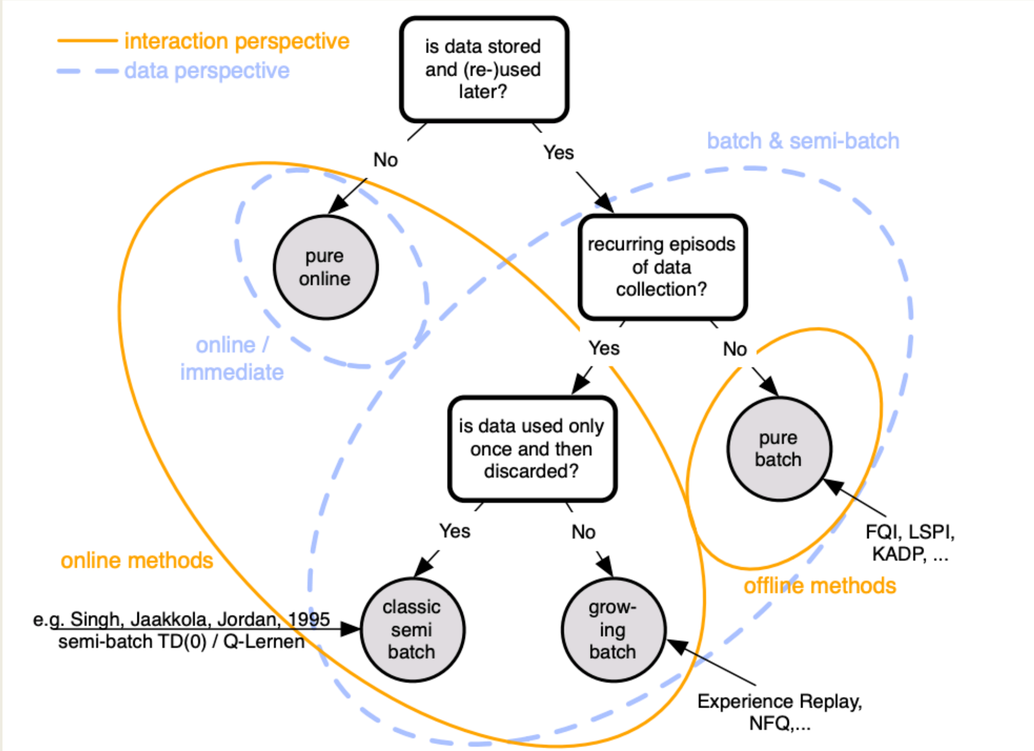


Fig. 4 Classification of batch vs. non-batch algorithms. With the interaction perspective and the data-usage perspective there are at least two different perspectives to define the category borders.

オフラインRLの想定される利用例

ヘルスケア

- 過去の治療や投薬の記録は得られるが，実際の患者でexplorationはできない

対話エージェント

- 例）eコマースのサイトで対話botを通じて顧客のコンバージョンを得たい
  - 対話エージェントを使って実際の環境でexplorationは難しいが，
    生身の人間の会話のデータは得られる

ロボットのマニピュレーション

- 幅広い環境で実行できる多様なスキルを学習しようとした場合，
  それぞれのスキルに対して実環境での相互作用による学習では不十分な可能性
  - 例）QT-Opt[Kalashnikov+18]（オフラインデータと実環境の相互作用の両方を用いる）

オフライン強化学習の難しさ

本質的には**探索ができない**ことが問題

- もしデータセット $\mathcal{D}$ に高い報酬の領域が全く含まれていなければ,
  それらを見つけるような方策を学習できない
  - なので，学習ができるぐらい適切に高い報酬の領域が含まれていることが前提
    - but どのぐらい高い報酬の領域が含まれているべきかはopen problem
- **counterfactualな問題**を解く必要がある
  - データセット $\mathcal{D}$ の挙動とは異なる方策を学習する問題
  - 「**分布シフト**」（distributional shift）として説明される
    - 訓練時の（データセットの）分布と，学習した方策のもとで
      テスト時に実際に現れるデータの分布が異なる

3.オフライン方策評価と重点サンプリングに基づくRL

off-policy方策評価（off-policy policy evaluation, OPE）

重点サンプリングを利用して方策 π_θ を評価

- $J(\pi_\theta)$ を評価できれば，最も性能の良い方策を選べるという発想 [Thomas+15]

$$J\left(\pi_\theta\right) = \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}\sum_{t=0}^{H}\gamma^t r(\mathbf{s},\mathbf{a})\right]$$

- $$= \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\left(\prod_{t=0}^{H}\frac{\pi_\theta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)}{\pi_\beta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)}\right)\sum_{t=0}^{H}\gamma^t r(\mathbf{s},\mathbf{a})\right] \approx \sum_{i=1}^{n} w_H^i \sum_{t=0}^{H}\gamma^t r_t^i$$

  - ただし，$w_t^i = \prod_{t=0}^{H}\frac{\pi_\theta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)}{\pi_\beta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)}$，$\left\{(\mathbf{s}_t^i, \mathbf{a}_t^i, r_t^i, \mathbf{s}_{t+1}^i, \cdots)\right\}_{i=1}^{n}$ は挙動方策 π_β による n 個の軌道のサンプル
  - 重点サンプル率の積のせいでunbiasedだが分散が非常に大きい
    - 重点サンプル率をself-normalize（$\sum_{i=1}^{n} w_H^i$ で割る）するとbiasedだが分散は小さくなる

off-policy方策評価（off-policy policy evaluation, OPE）

- ステップごとの重点サンプリングによる推定
  - $J\left(\pi_\theta\right) = \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\sum_{t=0}^{H}\left(\prod_{t'=0}^{t}\frac{\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_\beta(\mathbf{a}_t \mid \mathbf{s}_t)}\right)\gamma^t r(\mathbf{s},\mathbf{a})\right] \approx \frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{H} w_t^i \gamma^t r_t^i$
    - r_t が将来（$t' > t$）の $\mathbf{s}_t, \mathbf{a}_t$に依存しないことを利用
    - これも同様に高分散なことが多い（self-normalizationも使える）
- Doubly robust estimator
  - $J\left(\pi_\theta\right) \approx \sum_{i=1}^{n}\sum_{t=0}^{H}\gamma^t\left(w_t^i\left(r_t^i - \hat{Q}^{\pi_\theta}\left(\mathbf{s}_t,\mathbf{a}_t\right)\right) - w_{t-1}^i \mathbb{E}_{\mathbf{a}\sim\pi_\theta(\mathbf{a}\mid\mathbf{s}_t)}\left[\hat{Q}^{\pi_\theta}\left(\mathbf{s}_t,\mathbf{a}\right)\right]\right)$
    - $(\mathbf{s}_t, \mathbf{a}_t)$ に対するQ値のモデル $\hat{Q}(\mathbf{s}_t, \mathbf{a}_t)$ を作成し利用
    - 挙動方策 π_β が既知 or モデルが正確な場合にunbiased

off-policy方策勾配（off-policy policy gradient）

重点サンプリングを利用して方策 π_θ の方策勾配 $\nabla_\theta J(\pi_\theta)$ を評価

$$\nabla_\theta J\left(\pi_\theta\right) = \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}\sum_{t=0}^{H}\gamma^t\nabla_\theta\log\pi_\theta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)\hat{A}\left(\mathbf{s}_t,\mathbf{a}_t\right)\right]$$

$$= \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\left(\prod_{t=0}^{H}\frac{\pi_\theta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)}{\beta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)}\right)\sum_{t=0}^{H}\gamma^t\nabla_\theta\log\pi_\theta\left(\mathbf{a}_t \mid \mathbf{s}_t\right)\hat{A}\left(\mathbf{s}_t,\mathbf{a}_t\right)\right]$$

- $$\approx \sum_{i=1}^{n} w_H^i \sum_{t=0}^{H}\gamma^t\nabla_\theta\log\pi_\theta\left(\mathbf{a}_t^i \mid \mathbf{s}_t^i\right)\hat{A}\left(\mathbf{s}_t^i,\mathbf{a}_t^i\right)$$

- $\left\{(\mathbf{s}_t^i,\mathbf{a}_t^i,r_t^i,\mathbf{s}_{t+1}^i,\cdots)\right\}_{i=1}^{n}$ は挙動方策 π_β による n 個の軌道のサンプル
- 一般に高分散，self-normalizationもできる

off-policy方策勾配（off-policy policy gradient）

- 他によく用いられる推定法
  - ステップごとの重点サンプリングによる方策勾配の推定
    - $\nabla_\theta J\left(\pi_\theta\right) \approx \sum_{i=1}^{n} \sum_{t=0}^{H} w_t^i \gamma^t \left(\sum_{t'=t}^{H} \gamma^{t'-t} \frac{w_{t'}^i}{w_t^i} r_{t'} - b\left(\mathbf{s}_t^i\right) \right) \nabla_\theta \log \pi_\theta\left(\mathbf{a}_t^i \mid \mathbf{s}_t^i\right)$
  - Doubly robust estimator
    - off-policy方策評価のときと同様
  - Softmax正則化つきの方策勾配の推定
    - $\nabla_\theta \bar{J}\left(\pi_\theta\right) \approx \left(\sum_{i=1}^{n} w_H^i \sum_{t=0}^{H} \gamma^t \nabla_\theta \log \pi_\theta\left(\mathbf{a}_t^i \mid \mathbf{s}_t^i\right) \hat{A}\left(\mathbf{s}_t^i, \mathbf{a}_t^i\right) \right) + \lambda \log \left(\sum_{i=1}^{n} w_H^i \right)$
- 高分散の問題はある程度低減されるがやはり残る

off-policy方策勾配の近似

現在の方策の状態分布 $d^{\pi}(\mathbf{s})$ の代わりに**挙動方策の状態分布 $d^{\pi_\beta}(\mathbf{s})$ を用いる**

- $J_\beta(\pi_\theta) = \mathbb{E}_{\mathbf{s}\sim d^{\pi_\beta}(s)}\left[V^{\pi}(\mathbf{s})\right]$
  - $J(\pi_\theta)$ のbiasedな推定であり，sub-optimalな解になることもある[Imani+18]が，
    許容できる程度であることが実験的に分かっている[Fu+19]
  - オフラインの場合 $d^{\pi_\beta}(\mathbf{s})$ に関する期待値は $\mathcal{D}$ からサンプルすれば計算可能
    重点サンプリングが必要ない
- これを微分して近似することてoff-policy方策勾配も求まる

$$\nabla_\theta J_\beta\left(\pi_\theta\right) = \mathbb{E}_{\mathbf{s}\sim d^{\beta}(\mathbf{s}),\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}\left[Q^{\pi_\theta}(\mathbf{s},\mathbf{a})\,\nabla_\theta \log \pi_\theta(\mathbf{a}\mid\mathbf{s}) + \nabla_\theta Q^{\pi_\theta}(\mathbf{s},\mathbf{a})\right]$$

- $$\approx \mathbb{E}_{\mathbf{s}\sim d^{\beta}(\mathbf{s}),\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}\left[Q^{\pi_\theta}(\mathbf{s},\mathbf{a})\,\nabla_\theta \log \pi_\theta(\mathbf{a}\mid\mathbf{s})\right]$$
  - off-policyの軌道から追加的に $Q^{\pi_\theta}(\mathbf{s},\mathbf{a})$ を近似する必要がある
  - 様々なDRLアルゴリズムがこの形式　例）DPG [Silver+14]，IPG[Gu+17b]

Marginalized Importance Sampling

重点サンプル率 $\rho^{\pi_\theta}(\mathbf{s}) = \dfrac{d^{\pi_\theta}(\mathbf{s})}{d^{\beta}(\mathbf{s})}$ を推定する

- 状態分布の違いによるbiasや重点サンプル率の積による高分散を避ける
- **Forward Bellman Equation**に基づく手法
  - 重点サンプル率 $\rho^{\pi_\theta}(\mathbf{s})$ を直接推定
- **Backward Bellman Equation**に基づく手法
  - 凸双対性を利用して求める
  - 例）DualDICE[Nachum+19a]，AlgaeDICE[Nachum+19b]，[Nachum+20]
    - 輪読としてはあまりにもしんどすぎるので省略（もしかしたらRLArchで話すかも）

Marginalized Importance Sampling

- **Forward Bellman Equation**に基づく手法
  - 重点サンプル率 $\rho^{\pi_\theta}(\mathbf{s})$ を直接推定
  - $$\forall \mathbf{s}', \underbrace{d^{\pi_\beta}(\mathbf{s}')\rho^{\pi}(\mathbf{s}')}_{\left(d^{\pi_\beta}\circ\rho^{\pi}\right)(s,a)} = \underbrace{(1-\gamma)d_0(\mathbf{s}') + \gamma\sum_{\mathbf{s},\mathbf{a}} d^{\pi_\beta}(\mathbf{s})\rho^{\pi}(\mathbf{s})\pi(\mathbf{a}\mid\mathbf{s})T\left(\mathbf{s}'\mid\mathbf{s},\mathbf{a}\right)}_{\left(\overline{\mathcal{B}}^{\pi}\circ\rho^{\pi}\right)(\mathbf{s},\mathbf{a})}$$

    の関係を利用
  - 例）TD更新に基づいてオンラインに更新[Gelada+19]
  - $$\hat{\rho}^{\pi}(\mathbf{s}') \leftarrow \hat{\rho}^{\pi}(\mathbf{s}') + \alpha\left[(1-\gamma) + \gamma\frac{\pi(\mathbf{a}\mid\mathbf{s})}{\pi_\beta(\mathbf{a}\mid\mathbf{s})}\hat{\rho}^{\pi}(\mathbf{s}) - \hat{\rho}^{\pi}(\mathbf{s}')\right]$$
    - ただし，$\mathbf{s}\sim d^{\pi_\beta}$，$\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})$，$\mathbf{s}'\sim T(\mathbf{s}'|\mathbf{s},\mathbf{a})$

Marginalized Importance Sampling

- **Forward Bellman Equation**に基づく手法
  - 例）$\min_{\rho} \max_{f} L(\rho, f)^2$ として敵対的学習を用いて推定[Liu+18]
    - $$L(\rho, f) = \gamma \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}' \sim \mathscr{D}} \left[\left(\rho(\mathbf{s}) \frac{\pi(\mathbf{a} \mid \mathbf{s})}{\pi_\beta(\mathbf{a} \mid \mathbf{s})} - \rho(\mathbf{s}') \right) f(\mathbf{s}') \right] + (1-\gamma) \mathbb{E}_{\mathbf{s}_0 \sim d_0} [(1 - \rho(\mathbf{s})) f(\mathbf{s})]$$
      - π_β を利用しないような改良もある[Mousavi+20, Kallus+19, Tang+19, Uehara+19]
  - 例）GenDICE[Zhang+20]
    - データセット $\mathscr{D}$ 全体の重点サンプル率の期待値を1に制約した上で,
      ベルマン方程式の両辺のf-divergenceの最小化

      $$\min_{\rho^\pi} D_f \left(\left(\overline{\mathscr{B}^\pi} \circ \rho^\pi \right)(\mathbf{s}, \mathbf{a}), \left(d^{\pi_\beta} \circ \rho^\pi \right)(\mathbf{s}, \mathbf{a}) \right) \quad \text{s.t.} \quad E_{\mathbf{s},\mathbf{a},\mathbf{s}' \sim \mathscr{D}} \left[\rho^\pi(\mathbf{s}, \mathbf{a}) \right] = 1$$

残る課題

重点サンプリングに基づく手法は基本的にオンラインRLのoff-policyでの想定

- つまり挙動方策 π_β が逐次的に更新され続ける前提

オフラインRLの場合 π_β が固定なので，ターゲット方策 π_θ が離れすぎると

重点サンプル率が非常に小さくなり，収益や方策勾配の分散が大きくなる

- 高次元の状態・行動空間や長いエピソードの問題で顕著
- そのため，重点サンプリングに基づく手法は，ターゲット方策 π_θ が挙動方策 π_β からあまり変動しない場合に適する

残る課題

結局，ほとんどのoff-policy方策勾配の手法は

DPを使って価値関数やstate-marginalの密度比の推定が必要

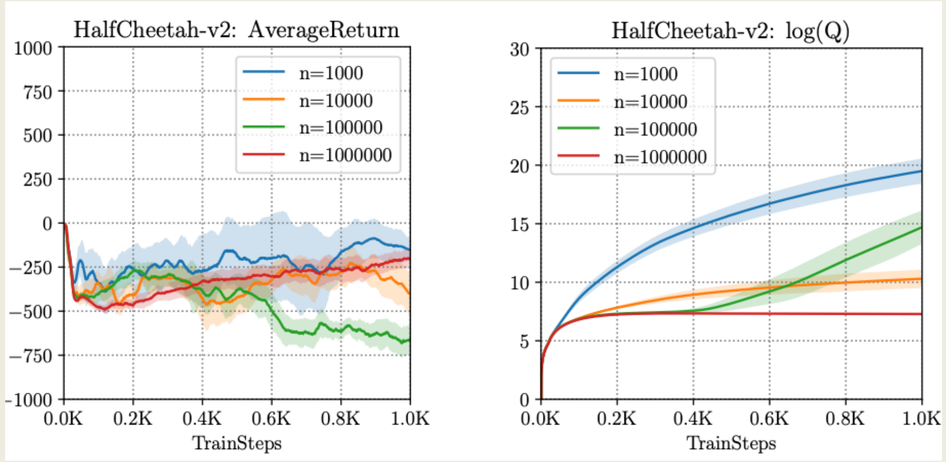
- オフラインRLの場合，テスト時のOoDの分布への対応が必要になる
  - DPを用いた場合の課題と同じ（後述）
  - 追加的な環境との相互作用による修正が不可能

4.DPに基づくオフラインRL

分布シフトへの対応

DPにおける分布シフト

- **テスト時**
  - 状態分布 $d^{\pi_\beta}(\mathbf{s})$ と $d^{\pi}(\mathbf{s})$ の違い
- **訓練時**
  - ベルマンターゲットが方策 $\pi(\mathbf{a}\,|\,\mathbf{s})$ に依存しているので訓練時にも問題になる
    - とくにQ-learningでQ値の更新で $\arg\max Q(\mathbf{s}, \mathbf{a})$ するアルゴリズムで顕著
  - オフラインRLではQ値の過大評価を修正できず，方策を更新するたびに誤差が蓄積
- 十分に大きなデータセットならそこまで問題でないという主張もある[Agarwal+19]



分布シフトへの対応

対処法

- **方策に制約**を課す（policy constraint）手法
  - ターゲット方策 π が挙動方策 π_β に近くなるように制約
- **不確実性の推定**に基づく（uncertainty-based）手法
  - Q値のepistemic uncertaintyを推定し，分布シフトを検知

方策に制約を課す手法

価値関数のターゲットを計算する際の**方策 $\pi(\mathbf{a}'|\mathbf{s}')$ が，挙動方策 $\pi_\beta(\mathbf{a}'|\mathbf{s}')$ に近くなるように制約**し，Q関数がOoDな領域に入らないようにする

制約の掛け方

- 直接actor（方策）の更新を制約
- 報酬やQ値のターゲットにペナルティを加える

制約の種類

- explicitなf-ダイバージェンス制約
- implicitなf-ダイバージェンス制約
- integral probability metric (IPM)による制約

方策に制約を課す手法

制約の掛け方

- **直接actor（方策）の更新を制約**

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

- $$\pi_{k+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}} \left[\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})} \left[\hat{Q}^{\pi}_{k+1}(\mathbf{s},\mathbf{a}) \right] \right] \quad \text{s.t.} \quad D\left(\pi, \pi_\beta\right) \leq \epsilon$$

- **報酬やQ値のターゲットにペナルティを加える**

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

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

方策に制約を課す手法

制約の種類

- **explicitなf-ダイバージェンス制約**
  - 例）KLダイバージェンスの利用：WOP[Jaques+19]，BRAC[Wu+19]
- **implicitなf-ダイバージェンス制約**
  - 例）KLダイバージェンス制約下で最適な方策を求めて教師あり回帰：AWR[Peng+19]，ABM[Siegel+20]

$$\bar{\pi}_{k+1}(\mathbf{a} \mid \mathbf{s}) \leftarrow \frac{1}{Z}\pi_\beta(\mathbf{a} \mid \mathbf{s})\exp\left(\frac{1}{\alpha}Q_k^\pi(\mathbf{s}, \mathbf{a})\right)$$

$$\pi_{k+1} \leftarrow \arg\min_\pi D_{\mathrm{KL}}\left(\bar{\pi}_{k+1}, \pi\right)$$

  - 実装上は，挙動方策 $\pi_\beta(\mathbf{a}\,|\,\mathbf{s})$ からのサンプルを $\exp\left(\frac{1}{\alpha}Q_k^\pi(\mathbf{s}, \mathbf{a})\right)$ で重みづけた教師あり回帰

- **integral probability metric (IPM)による制約**
  - 例）MMDの利用：BEAR [Kumar+19]，Wasserstein距離の利用：BRAC[Wu+19]

方策に制約を課す手法

制約間の比較

- オフラインRLの場合，**KL(f-divergence)制約が必ずしも最適ではない**
  - 例）挙動方策がuniformにランダムな場合，理論上オフラインRLはうまくいくが，KL制約をかけるとターゲット方策が報酬が高い領域をexploitせずにランダムな方策に近づいてしまう
- 直感的には，ターゲット方策がデータの高い確率の行動の外に行くことを防ぐが，高い確率の行動に集中することは防がない制約が効果的
  - **ターゲット方策のサポートを挙動方策のサポートで制約**すれば十分なのでは説[Kumar+19]

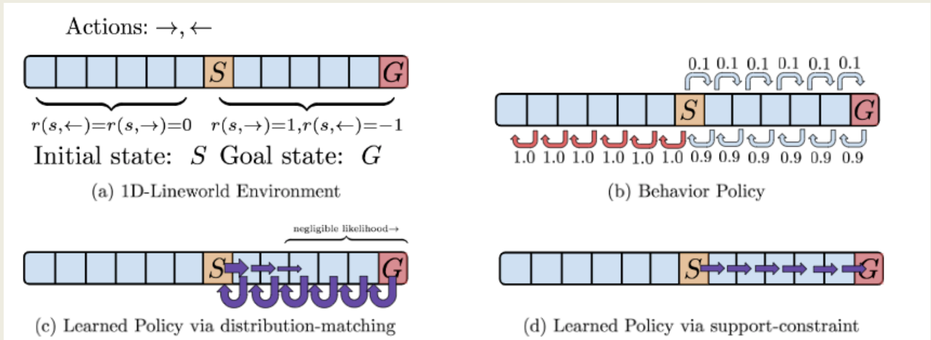


Figure 3: A comparison of support and distribution constraints on a simple 1D lineworld from Kumar (2019). The task requires moving to the goal location (marked as 'G') starting from 'S'. The behavior policy strongly prefers the left action at each state (b), such that distribution constraint is unable to find the optimal policy (c), whereas support-constraint can successfully obtain the optimal policy (d). We refer to Kumar (2019) for further discussion.

不確実性の推定に基づく手法

Q関数の不確実性（epstemic uncertainty）を推定し方策の更新に利用

- $$\pi_{k+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}} \underbrace{\left[\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})} \left[\mathbb{E}_{Q^{\pi}_{k+1}\sim\mathscr{P}_{\mathcal{D}}(Q^{\pi})} \left[Q^{\pi}_{k+1}(\mathbf{s},\mathbf{a}) \right] - \alpha\, \mathrm{Unc}\left(\mathscr{P}_{\mathcal{D}}\left(Q^{\pi}\right) \right) \right] \right]}_{\text{conservative estimate}}$$
  - $\mathscr{P}_{\mathcal{D}}(Q^{\pi})$ は $\mathcal{D}$ を基に推定したQ関数の分布，Unc は不確実性のなんらかの評価指標

Q関数の**アンサンブル**に基づくものが多い

- Unc に推定したQ値の分散を利用　例）BEAR [Kumar+19]
- Unc に推定したQ値の最も低い値を利用　例）[Agarwal+19]

残る課題

不確実性の推定に基づく手法では，**不確実性のcalibration**をどうするか？

- $\mathscr{P}_{\mathscr{D}}$ と Unc の設計が難しい
- 実際には方策を制約する手法の方がうまく行っている[Fujimoto+18a]
- Q値の過大評価を防ぐことが重要

方策を制約する手法では，データセット $\mathscr{D}$ から挙動方策 π_β の推定が必要

- 最終的な性能が**挙動方策の推定誤差により制限される**
  - 特に真の挙動方策が多峰の場合に問題になる可能性
  - データセット $\mathscr{D}$ のサンプルから直接制約をかける手法の開発はopen problem

5.オフラインモデルベースRL

モデルベース手法

- パラメータ化された状態遷移モデル $T_\psi\left(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t\right)$ を学習し，
  - プランニングに利用　例）MPC [Tassa+12]
  - ダイナミクスモデルを通じてBPTT　例）PILCO[Deisenroth+11]
  - 方策の学習に利用　例）Dyna[Sutton+91]，PIPPS[Parmas+19]
    - ダイナミクスモデルから仮想的な状態遷移を生成し，その状態遷移から方策を学習

オフラインRLへの利用

- 安定した教師あり学習を利用できるので有望
  - ただし，distribution shift自体の解決にはならない
- しかし，オフラインモデルベースRLに関する研究は「surprisingly limited」
  - その後たくさん出ました

オフラインモデルベースRL

このサーベイ論文後にいくつか公開

（たぶん全部NeurIPS2020に投稿）

MOReL [Kidambi+20*]

- ダイナミクスモデルのアンサンブル
- Pessimistic MDP
  - 不確実性が一定以上高いとき
    その遷移を学習に用いない

MOPO [Yu+20*]

- ダイナミクスモデルのアンサンブル
- 報酬にダイナミクスモデルの不確実性
  に比例したペナルティを加える

BREMEN [Matsushima+20*]

- ダイナミクスモデルのアンサンブル
  +推定した挙動方策で初期化しTRPO
- 既存のオフラインRLの研究の5-10%の
  データセットでも方策改善を確認

1,000,000 (1M) transitions				
Method	Ant	HalfCheetah	Hopper	Walker2d
Dataset	1191	4126	1128	1376
BC	1321±141	4281±12	1341±161	1421±147
BCQ [16]	2021±31	5783±272	1130±127	2153±753
BRAC [61]	2072±285	7192±115	1422±90	2239±1124
BRAC (max Q)	2369±234	7320±91	1916±343	**2409±1210**
BREMEN (Ours)	**3328±275**	**8055±103**	**2058±852**	2346±230
ME-TRPO (offline) [31]	1258±550	1804±924	518±91	211±154
100,000 (100K) transitions				
Method	Ant	HalfCheetah	Hopper	Walker2d
Dataset	1191	4066	1128	1376
BC	1330±81	4266±21	1322±109	1426±47
BCQ	1363±199	3915±411	1129±238	**2187±196**
BRAC	-157±383	2505±2501	1310±70	2162±1109
BRAC (max Q)	-226±387	2332±2422	1422±101	2164±1114
BREMEN (Ours)	**1633±127**	**6095±370**	**2191±455**	2132±301
ME-TRPO (offline)	974±4	2±434	307±170	10±61
50,000 (50K) transitions				
Method	Ant	HalfCheetah	Hopper	Walker2d
Dataset	1191	4138	1128	1376
BC	1270±65	4230±49	1249±61	1420±194
BCQ	1329±95	1319±626	1178±235	1841±439
BRAC	-878±244	-597±73	1277±102	976±1207
BRAC (max Q)	-843±279	-590±56	1276±225	903±1137
BREMEN (Ours)	**1347±283**	**5823±146**	**1632±796**	**2280±647**
ME-TRPO (offline)	938±32	-73±95	152±13	176±343

残る課題

そもそも非常に高次元でlong-horizonなMDPのモデル化はopen problem

- 全観測の予測をしないですむRLアルゴリズムの利用が有望
  - 例）DFP[Dosovitskiy+17], VPN[Oh+17], BADGR[Kahn+20]

状態遷移のモデルの学習が有効であるかどうかの理論的な検討が必要

- DPは将来の収益，モデルベースRLは将来の状態に関する予測
  - 線形なモデルではモデルベースの更新も価値の更新（fitted value iteration）も同様の更新になることが知られている[Vanseijen+15,Parr+08]

7.議論と展望

議論と展望

オフラインRLは本質的には**counterfactualな推論**に結びついてる可能性

- ある意思決定をした元での結果の集合が与えられた場合に，それとは異なる意思決定をしたときの結果を推論する問題
  - i.i.dの枠組みから外れるので難しいとされてきた問題

汎化性能を高める・分布シフトに対処する**他の手法との組み合わせ**が有効かも

- 例）因果推論[Schölkopf19]，不確実性の推定[Gal+16]，生成モデル[Kingma+14], distributional robustness[Sinha+17]，invariance[Arjovsky+19]
- 例）モデルベース手法
  visual foresight[Finn+17,Ebert+18]，InteractionNet[Battaglia+16]

データドリブンなRLがRLの新しい時代を築く

まとめ

感想

オフラインRLのデータセット・ベンチマークが出揃ってきた

- 明らかにこれからこの分野の研究が進んでゆく
  - 例）D4RL[Kumar+20]，RL Unplugged [Gulcehre+20*]
- **今後は実機検証・マルチタスク（メタ）化**していくはず
  - もうその傾向にある　例）IRIS[Mandlekar+19*]
  - そのなかで，共通する環境のダイナミクスを学習するモデルベース手法は大事

一方で，良いフレームワーク・ライブラリがない

- 既存のoff-policy手法が互いにどのように異なり，どの要素がオフラインRLに利用できて，追加でどのような工夫をすればいいのかが明らかでない状態
  - 分布で扱うような（Control as Inference的思想の）RLフレームワークがない（ほしい）

模倣学習の手法・問題設定との融合

- 模倣学習との差分は報酬データの有無，今後相互に発展しそう
  - 最尤推定（BC）系の手法との融合　例）AWR[Peng+19]
  - 報酬関数の学習　例）GAIL[Ho+16*]
  - ランキングデータの利用　例）T-REX[Brown+19a*]，D-REX[Brown+19b*]

参考文献

[Agarwal+19] Rishabh Agarwal, Dale Schuurmans, Mohammad Norouzi. “An Optimistic Perspective on Offline Reinforcement Learning” https://arxiv.org/abs/1907.04543

[Arjovsky+19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, David Lopez-Paz. "Invariant Risk Minimization”. https://arxiv.org/abs/1907.02893

[Battaglia+16] Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, Koray Kavukcuoglu. “Interaction Networks for Learning about Objects, Relations and Physics”. https://arxiv.org/abs/1612.00222

[Brown+19a*] Daniel S. Brown, Wonjoon Goo, Prabhat Nagarajan, Scott Niekum. “Extrapolating Beyond Suboptimal Demonstrations via Inverse Reinforcement Learning from Observations”. https://arxiv.org/abs/1904.06387

[Brown+19b*] Daniel S. Brown, Wonjoon Goo, Scott Niekum. “Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations”. https://arxiv.org/abs/1907.03976

[Deisenroth+11] Marc Deisenroth, Carl E. Rasmussen. "PILCO: A model-based and data-efficient approach to policy search." *Proceedings of the 28th International Conference on machine learning*. 2011.

[Dosovitskiy+17] Alexey Dosovitskiy, Vladlen Koltun. “Learning to Act by Predicting the Future”. https://arxiv.org/abs/1611.01779

[Ebert+18] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, Sergey Levine. “Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control”. https://arxiv.org/abs/1812.00568

[Finn+17] Chelsea Finn, Sergey Levine. “Deep Visual Foresight for Planning Robot Motion”. https://arxiv.org/abs/1610.00696

[Fu+19] Justin Fu, Aviral Kumar, Matthew Soh, Sergey Levine. “Diagnosing bottlenecks in deep Q-learning algorithms”. https://arxiv.org/abs/1902.10250

[Fujimoto+18a] Scott Fujimoto, David Meger, Doina Precup. “Off-Policy Deep Reinforcement Learning without Exploration”. https://arxiv.org/abs/1812.02900

[Gal+16] Yarin Gal, Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. https://arxiv.org/abs/1506.02142

[Gelada+19] Carles Gelada, Marc G. Bellemare. “Off-policy deep reinforcement learning by bootstrapping the covariate shift”. https://arxiv.org/abs/1901.09455

[Gu+17] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E. Turner, Bernhard Schölkopf, Sergey Levine. “Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning”. https://arxiv.org/abs/1706.00387

[Gulcehre+20*] Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Tom Le Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matt Hoffman, Ofir Nachum, George Tucker, Nicolas Heess, Nando de Freitas. “RL Unplugged: Benchmarks for Offline Reinforcement Learning”. https://arxiv.org/abs/2006.13888

[Ho+16*] Jonathan Ho, Stefano Ermon. “Generative Adversarial Imitation Learning”. https://arxiv.org/abs/1606.03476

[Imani+18] Ehsan Imani, Eric Graves, Martha White. “An Off-policy Policy Gradient Theorem Using Emphatic Weightings”. https://arxiv.org/abs/1811.09013

参考文献

[Jaques+19] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, Rosalind Picard. “Way Off-Policy Batch Deep Reinforcement Learning of Implicit Human Preferences in Dialog”. https://arxiv.org/abs/1907.00456

[Kahn+20] Gregory Kahn, Pieter Abbeel, Sergey Levine. “BADGR: An Autonomous Self-Supervised Learning-Based Navigation System”. https://arxiv.org/abs/2002.05700

[Kalashnikov+18] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, Sergey Levine. “QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation”. https://arxiv.org/abs/1806.10293

[Kallus+19] Nathan Kallus, Masatoshi Uehara. “Efficiently Breaking the Curse of Horizon in Off-Policy Evaluation with Double Reinforcement Learning”. https://arxiv.org/abs/1909.05850

[Kidambi+20*] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, Thorsten Joachims. “MOReL : Model-Based Offline Reinforcement Learning”. https://arxiv.org/abs/2005.05951

[Kingma+14] Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, Max Welling. “Semi-Supervised Learning with Deep Generative Models” https://arxiv.org/abs/1406.5298

[Kumar+19] Aviral Kumar, Justin Fu, George Tucker, Sergey Levine. “Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction”. https://arxiv.org/abs/1906.00949

[Kumar+20] Does on-policy data collection fix errors in reinforcement learning? https://bair.berkeley.edu/blog/2020/03/16/discor/.

[Lange+12] Sascha Lange, Thomas Gabel, Martin Riedmiller. “Batch reinforcement learning”. *Reinforcement learning*. Springer, 2012. 45-73.

[Liu+18] Qiang Liu, Lihong Li, Ziyang Tang, Dengyong Zhou. “Breaking the curse of horizon: Infinite-horizon off-policy estimation”. https://arxiv.org/abs/1810.12429

[Mandlekar+19*] Ajay Mandlekar, Fabio Ramos, Byron Boots, Silvio Savarese, Li Fei-Fei, Animesh Garg, Dieter Fox. “IRIS: Implicit Reinforcement without Interaction at Scale for Learning Control from Offline Robot Manipulation Data”. https://arxiv.org/abs/1911.05321

[Matsushima+20*] Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, Shixiang Gu. “Deployment-Efficient Reinforcement Learning via Model-Based Offline Optimization”. https://arxiv.org/abs/2006.03647

[Mousavi+20] Ali Mousavi, Lihong Li, Qiang Liu, Denny Zhou. “Black-box Off-policy Estimation for Infinite-Horizon Reinforcement Learning”. https://arxiv.org/abs/2003.11126

[Nachum+19a] Ofir Nachum, Yinlam Chow, Bo Dai, Lihong Li. “DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections”. https://arxiv.org/abs/1906.04733

[Nachum+19b] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, Dale Schuurmans. “AlgaeDICE: Policy Gradient from Arbitrary Experience”. https://arxiv.org/abs/1912.02074

[Nachum+20] Ofir Nachum, Bo Dai. “Reinforcement Learning via Fenchel-Rockafellar Duality”. https://arxiv.org/abs/2001.01866

参考文献

[Oh+17] Junhyuk Oh, Satinder Singh, Honglak Lee. “Value Prediction Network”. https://arxiv.org/abs/1707.03497

[Parr+08] Parr, Ronald, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, Michael L. Littman. “An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning”. *In Proceedings of the 25th international conference on Machine learning.* 2008.

[Parmas+19] Paavo Parmas, Carl Edward Rasmussen, Jan Peters, Kenji Doya. “PIPPS: Flexible Model-Based Policy Search Robust to the Curse of Chaos”. https://arxiv.org/abs/1902.01240

[Peng+19] Xue Bin Peng, Aviral Kumar, Grace Zhang, Sergey Levine. "Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning”. https://arxiv.org/abs/1910.00177

[Schölkopf19] Bernhard Schölkopf. “Causality for machine learning”. https://arxiv.org/abs/1911.10500

[Siegel+20] Noah Y. Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, Martin Riedmiller. “Keep Doing What Worked: Behavioral Modelling Priors for Offline Reinforcement Learning”. https://arxiv.org/abs/2002.08396

[Sinha+17] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, John Duchi. “Certifying Some Distributional Robustness with Principled Adversarial Training”. https://arxiv.org/abs/1710.10571

[Silver+14] David Silver. Guy Lever. Nicolas Heess. Thomas Degris. Daan Wierstra. Martin Riedmiller. ”Deterministic policy gradient algorithms”. *Proceedings of the 31st International Conference on International Conference on Machine Learning*. 2014.

[Sutton+91] Dyna, an integrated architecture for learning, planning, and reacting”. *ACM Sigart Bulletin,* 2(4), 160-163. 1991.

[Tassa+12] Yuval Tassa, Tom Erez, Emanuel Todorov. "Synthesis and stabilization of complex behaviors through online trajectory optimization." *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012.

[Tang+19] Ziyang Tang, Yihao Feng, Lihong Li, Dengyong Zhou, Qiang Liu. “Doubly Robust Bias Reduction in Infinite Horizon Off-Policy Estimation”. https://arxiv.org/abs/1910.07186

[Thomas+15] Philip S. Thomas, Georgios Theocharous, Mohammad Ghavamzadeh. "High-confidence off-policy evaluation”. *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[Uehara+19] Masatoshi Uehara, Jiawei Huang, Nan Jiang. “Minimax Weight and Q-Function Learning for Off-Policy Evaluation”, https://arxiv.org/abs/1910.12809

[Vanseijen+15] Harm Vanseijen, Rich Sutton. "A deeper look at planning as learning from replay." *In International conference on machine learning*. 2015.

[Wu+19] Yifan Wu, George Tucker, Ofir Nachum. “Behavior Regularized Offline Reinforcement Learning”. https://arxiv.org/abs/1911.11361

[Yu+20*] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, Tengyu Ma. “MOPO: Model-based Offline Policy Optimization”. https://arxiv.org/abs/2005.13239

[Zhang+20] Ruiyi Zhang, Bo Dai, Lihong Li, Dale Schuurmans. “GenDICE: Generalized Offline Estimation of Stationary Values”. https://arxiv.org/abs/2002.09072