Machine Learning (2024) 113:6141–6163
https://doi.org/10.1007/s10994-024-06564-5

The impact of data distribution on Q-learning with function approximation

Pedro P. Santos[1,2] · **Diogo S. Carvalho**[1,2] · **Alberto Sardinha**[1,3] · **Francisco S. Melo**[1,2]

Received: 29 November 2022 / Revised: 14 March 2024 / Accepted: 26 April 2024 /
Published online: 7 June 2024
© The Author(s) 2024

Abstract
We study the interplay between the data distribution and Q-learning-based algorithms with function approximation. We provide a unified theoretical and empirical analysis as to how different properties of the data distribution influence the performance of Q-learning-based algorithms. We connect different lines of research, as well as validate and extend previous results, being primarily focused on offline settings. First, we analyze the impact of the data distribution by using optimization as a tool to better understand which data distributions yield low concentrability coefficients. We motivate high-entropy distributions from a game-theoretical point of view and propose an algorithm to find the optimal data distribution from the point of view of concentrability. Second, from an empirical perspective, we introduce a novel four-state MDP specifically tailored to highlight the impact of the data distribution in the performance of Q-learning-based algorithms with function approximation. Finally, we experimentally assess the impact of the data distribution properties on the performance of two offline Q-learning-based algorithms under different environments. Our results attest to the importance of different properties of the data distribution such as entropy, coverage, and data quality (closeness to optimal policy).

Keywords Machine learning · Reinforcement learning · Offline reinforcement learning · Off-policy learning

Editors: Fabio Vitale, Tania Cerquitelli, Marcello Restelli, Charalampos Tsourakakis.

✉ Pedro P. Santos
pedro.pinto.santos@tecnico.ulisboa.pt

Diogo S. Carvalho
diogo.s.carvalho@tecnico.ulisboa.pt

Alberto Sardinha
sardinha@inf.puc-rio.br

Francisco S. Melo
fmelo@inesc-id.pt

[1] INESC-ID, Lisbon, Portugal

[2] Instituto Superior Técnico, Lisbon, Portugal

[3] Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil

1 Introduction

Recent years witnessed significant progress in solving challenging problems across various domains using reinforcement learning (RL) (Lillicrap et al., 2016; Mnih et al., 2015; Silver et al., 2017). Q-learning algorithms with function approximation are among the most used methods (Arulkumaran et al., 2017). However, the combination of Q-learning with function approximation is non-trivial, especially for the case of large capacity approximators such as neural networks. Several works analyze the unstable behavior of such algorithms both experimentally (Fu et al., 2019; van Hasselt et al., 2018) and theoretically (Carvalho et al., 2020; Zhang et al., 2021).

The interplay between the data distribution and the outcome of the learning process is one potential source of instability of Q-learning-based algorithms (Kumar et al., 2020; Sutton & Barto, 2018). Different lines of research shed some light on how the data distribution impacts algorithmic stability. For example, some works provide examples that induce unstable behavior in off-policy learning (Baird, 1995; Kolter, 2011); some theoretical works derive error bounds on the performance of Q-learning-related algorithms (Chen & Jiang, 2019; Munos, 2005; Munos & Szepesvári, 2008); yet other studies investigate the stability of RL methods with function approximation (Fu et al., 2019; Kumar et al., 2020) or study unsupervised reward-free exploration for offline RL (Lambert et al., 2022; Yarats et al., 2022).

We center our study around the following research question: *which data distributions lead to improved algorithmic stability and performance?* In the context of this work, we refer to the data distribution as the distribution used to sample experience or the distribution induced by a dataset of transitions. We investigate how different data distribution properties influence performance in the context of Q-learning-based algorithms with function approximation. We add to previous works by providing a systematic and comprehensive study that connects different lines of research, as well as validating and extending previous results. We primarily focus on offline RL settings with discrete state and action spaces (Levine et al., 2020), in which an RL agent aims to learn reward-maximizing behavior using previously collected data without additional interaction with the environment. Nevertheless, our conclusions are also relevant in online RL settings, particularly for algorithms that rely on large-scale replay buffers. Our conclusions contribute to a deeper understanding of the influence of the data distribution properties in the performance of approximate value iteration (AVI) methods.

We start by presenting some background and the notation used throughout the paper in Sect. 2, as well as reviewing bounds on the performance of AVI-related methods in Sect. 2.1. In particular, we review the different concentrability coefficients proposed to quantify the suitability of the data distribution under offline settings, as well as their impact on the tightness of the bounds. Then, we investigate how the data distribution impacts the performance of AVI-related methods, being primarily focused on offline settings: in Sect. 3, we address this question from the point of view of concentrability and, in Sect. 4, we address it from an empirical point of view. In Sect. 3.1, we motivate high entropy distributions from a game-theoretical point of view. In Sect. 3.2, we use optimization as a tool to better understand which data distributions yield low concentrability coefficients by proposing a gradient descent-based algorithm to estimate the optimal data distribution. We expect our algorithm to open new directions for the development of theoretically grounded pre-processing schemes for offline RL or exploration methods for online RL. From an empirical point of view, we propose, in Sect. 4.1, a novel four-state MDP specifically tailored to highlight how the data distribution impacts algorithmic

performance, both online and offline. Then, in Sect. 4.2, we empirically assess the impact of the data distribution on the performance of offline Q-learning-based algorithms with function approximation under different environments, connecting the obtained results with the discussion from the previous sections. According to our results: (i) high entropy data distributions are well-suited for offline learning; and (ii) a certain degree of data diversity (data coverage) and data quality (closeness to optimal policy) are jointly desirable for offline learning. In Sect. 5, we connect our work with previous research. Finally, in Sect. 6, we present our conclusions and explore how our findings and contributions can extend to the online setting.

2 Background

We model the agent-environment interaction as a Markov decision process (MDP) (Puterman, 2014), formally defined as a tuple $(\mathcal{S}, \mathcal{A}, p, p_0, r, \gamma)$, where $\mathcal{S}$ denotes the discrete state space, $\mathcal{A}$ denotes the discrete action space, $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the state transition probability function with $\Delta(\mathcal{S})$ being the set of distributions on $\mathcal{S}$, $p_0 \in \Delta(\mathcal{S})$ is the initial state distribution, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is a discount factor. At each step t, the agent observes the state of the environment $s_t \in \mathcal{S}$ and chooses an action $a_t \in \mathcal{A}$. Depending on the chosen action, the environment evolves to state $s_{t+1} \in \mathcal{S}$ with probability $p(s_{t+1}|s_t, a_t)$, and the agent receives a reward r_t with expectation given by $r(s_t, a_t)$. A policy $\pi \in \Delta(\mathcal{A})^{|\mathcal{S}|}$ is a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. We denote by P^π the $|\mathcal{S}| \times |\mathcal{S}|$ matrix with elements $P^\pi(s, s') = \mathbb{E}_{a\sim\pi(a|s)}\big[p(s'|s, a)\big]$. A trajectory, $\tau = (s_0, a_0, \ldots, s_\infty, a_\infty)$, comprises a sequence of states and actions. The probability of a trajectory τ under a given policy π is given by $\varrho_\pi(\tau) = p_0(s_0) \prod_{t=0}^{\infty} \pi(a_t|s_t)p(s_{t+1}|s_t, a_t)$. The discounted reward objective can be written as

$$J(\pi) = \mathbb{E}_{\tau\sim\varrho_\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right].$$

The objective of the agent is to find an optimal policy π^* that maximizes the objective function above such that $J(\pi^*) \geq J(\pi), \forall\pi$. The optimal value function, $V^* \in \mathbb{R}^{|\mathcal{S}|}$, as well as the optimal action-value function, $Q^* \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, both satisfy the Bellman optimality equation:

$$Q^*(s, a) = r(s, a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\max_{a'\in\mathcal{A}} Q^*(s', a')\right] \tag{1}$$

$$V^*(s) = \max_{a\in\mathcal{A}} \left\{r(s, a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\big[V(s')\big]\right\}, \tag{2}$$

and $V^*(s) = \max_{a\in\mathcal{A}} Q^*(s, a)$. The optimal policy can be recovered directly from Q^*, but also from V^* if one has access to the transition probability function and the reward function. Planning algorithms, such as value-iteration (VI) (Puterman, 2014), can compute V^* or Q^* by taking advantage of the fact that such functions are the unique fixed-point of the Bellman optimality operator. For action-value functions, the Bellman optimality operator $\mathcal{T} : \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, is defined, for arbitrary $Q \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, as

$$(\mathcal{T}Q)(s, a) = r(s, a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\max_{a'\in\mathcal{A}} Q(s', a')\right],$$

and a similar operator exists for value-functions.

For problems featuring large state spaces, it is usually prohibitive to learn exact solutions as those yielded by VI-related algorithms. Instead, we need to resort to approximate solutions. Approximate value iteration algorithms aim to approximate V^* or Q^* by selecting a function $f \in \mathcal{F}$, where $\mathcal{F}$ is the class of functions encoding the representable value or action-value functions. In practice, $\mathcal{F}$ can correspond to that of linear approximators or more complex mappings such as neural networks. AVI-related algorithms start from an initial value-function, f_0, and iteratively apply an approximation of the operator $\mathcal{T}$,

$$f_{n+1} = \text{Proj}(\mathcal{T} f_n), \quad \text{Proj}(g) = \arg\min_{f \in \mathcal{F}} \|g - f\|_{p,\mu},$$

where μ denotes the data distribution and $\|\cdot\|_{p,\mu}$ is the L_p-norm weighted by distribution μ, defined as $\|x\|_{p,\mu} = \left(\sum_{s \in \mathcal{S}} \mu(s)|x(s)|^p\right)^{1/p}$ in the case of value functions. We have that $\mu \in \Delta(\mathcal{S})$ when learning value-functions and $\mu \in \Delta(\mathcal{S} \times \mathcal{A})$ in the case of action-value functions. In general, $f_{n+1} \neq \mathcal{T} f_n$ because the function space $\mathcal{F}$ is not representative enough. Furthermore, as is the case in RL, we have no access to the transition probability function and the reward function; thus, we do not have access to operator $\mathcal{T}$ but only some samples from it. Putting all together, AVI-related algorithms iteratively update f_n as described in Algorithm 1, where N denotes the number of iterations of the algorithm, M the number of samples, and l is a loss function (e.g., L_2 loss). A similar algorithm exists to learn V-functions instead of Q-functions.

Algorithm 1 Approximate Q-Iteration.

$f_0 \in \mathcal{F}$
for $n \in \{1, \ldots, N\}$ **do**
  $\{(s_m, a_m) \sim \mu\}_{m \in \{1,\ldots,M\}}$ ▷ Sample M state-action pairs from μ.
  $y_m = (\mathcal{T} f_{n-1})(s_m, a_m), \quad \forall m \in \{1, \ldots, M\}$
  $f_{n+1} = \arg\min_{f \in \mathcal{F}} \frac{1}{M} \sum_{m \in \{1,\ldots,M\}} l(f(s_m, a_m) - y_m)$
end for
Return f_N

Fitted Q-iteration (FQI) (Riedmiller, 2005), as well as the deep Q-network (DQN) (Mnih et al., 2015) algorithm, can be seen as particular instances of the algorithm above. The setting in which μ is fixed and arbitrary is generally known as offline (or batch) RL (Levine et al., 2020) because the agent is unable to control the data generation process. This is in contrast to the online RL setting where the agent can, up to some extent, control the data generation.

The fundamental problem of offline RL is that of distributional shift: out-of-distribution samples lead to algorithmic instabilities and performance loss, both at training and deployment time. The conservative Q-learning (CQL) (Kumar et al., 2020) algorithm is an offline RL algorithm that aims to estimate the optimal Q-function while mitigating the impact of distributional shift. The algorithm avoids the overestimation of out-of-distribution actions by considering an additional penalty loss term, with

$$f_{n+1} = \arg\min_{f \in \mathcal{F}} \frac{1}{M} \sum_m \left(l(f(s_m, a_m) - y_m) + k \sum_{a \in \mathcal{A}} \nu(a|s_m) f(s_m, a) \right), \tag{3}$$

where $k \in \mathbb{R}_0^+$ and distribution ν adversarially selects overestimated Q-values with high probability, e.g., by maximizing the second term in (3).

In the next section, we review works that aim to quantify the performance of AVI-related methods under offline settings, being particularly interested in understanding how proposed error bounds depend on the data distribution μ.

2.1 Concentrability coefficients

Different works analyze error propagation in AVI-related methods (Chen & Jiang, 2019; Munos, 2003, 2005; Munos & Szepesvári, 2008; Yang et al., 2019). Specifically, the aforementioned works provide upper bounds of the type $\|V^* - V^{\pi_k}\|_{p,\rho} \leq C \cdot \mathcal{F} + \mathcal{E}$ or $\|Q^* - Q^{\pi_k}\|_{p,\rho} \leq C \cdot \mathcal{F} + \mathcal{E}$. Distribution ρ reflects the importance of various regions of the state/state-action space and is selected by the practitioner. Intuitively, the bounds correspond to ρ-weighted L_p-norms between V^*/Q^* and the value/action-value function induced by the greedy policy π_k with respect to the estimated value/action-value function at the k-th timestep.[1] Such bounds comprise, generally, three key components:

(1) A concentrability coefficient, C, that quantifies the suitability of the sampling distribution $\mu \in \Delta(\mathcal{S})$ or $\mu \in \Delta(\mathcal{S}, \mathcal{A})$.
(2) A measure of the approximation power of the function space, $\mathcal{F}$, which reflects how well the function space is aligned with the dynamics and reward of the MDP.
(3) A coefficient $\mathcal{E}$ that captures the sampling error of the algorithm, i.e., the error that accumulates due to limited sampling and iterations.

From the three components above, we focus our attention on the study of the concentrability coefficient as it captures the impact of the data distribution in the tightness of the upper bound.

Munos (2003) introduces the first version of this data-dependent concentrability coefficient, which is related to the density of the transition probability function. Specifically, the author defines the coefficient $C_1 \in \mathbb{R}^+ \cup \{+\infty\}$ as

$$C_1 = \max_{s,s' \in \mathcal{S}, a \in \mathcal{A}} \frac{p(s'|s,a)}{\mu(s')}, \tag{4}$$

with $\mu \in \Delta(\mathcal{S})$ and the convention that $0/0 = 0$, and $C_1 = \infty$ if $\mu(s') = 0$ and $p(s'|s,a) > 0$. We use this convention for all upcoming coefficients. Intuitively, the noisier the dynamics of the MDP, the smaller the coefficient C_1 and the tighter the bound. Munos (2005) introduces a different concentrability coefficient related to the discounted average concentrability of future states on the MDP. Specifically, coefficient $C_2 \in \mathbb{R}^+ \cup \{+\infty\}$ is defined as

$$C_2 = (1-\gamma)^2 \sum_{m=1}^{\infty} m\gamma^{m-1} c(m), \tag{5}$$

$$c(m) = \max_{\pi_1,\ldots,\pi_m \in \Delta(\mathcal{A})^{|\mathcal{S}|}, s \in \mathcal{S}} \frac{(\rho^\top P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m})(s)}{\mu(s)}, \tag{6}$$

[1] We use the bounds of Munos and Szepesvári (2008) as reference.

with $\mu, \rho \in \Delta(\mathcal{S})$. Intuitively, coefficient C_2 expresses some smoothness property of the future state distribution with respect to μ for an initial distribution ρ. Munos (2005) and Munos and Szepesvári (2008) note that assumption $C_1 < \infty$ is stronger than assumption $C_2 < \infty$.

Farahmand et al. (2010) and Yang et al. (2019) replace coefficient (6) with

$$c(m) = \max_{\pi_1,\ldots,\pi_m \in \Delta(\mathcal{A})^{|\mathcal{S}|}} \left(\mathbb{E}_{(s,a)\sim\mu} \left[\left| \frac{(\rho^\top P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m})(s,a)}{\mu(s,a)} \right|^2 \right] \right)^{1/2}, \tag{7}$$

for $\mu, \rho \in \Delta(\mathcal{S} \times \mathcal{A})$. Let $C_3 \in \mathbb{R}^+ \cup \{+\infty\}$ denote the coefficient defined by (5) and (7).

Other works (Antos et al., 2008; Chen & Jiang, 2019; Lazaric et al., 2012, 2016; Munos, 2007; Tosatto et al., 2017; Xie & Jiang, 2020) use concentrability coefficients similar to those presented to derive performance bounds for various algorithms.

More recently, Chen and Jiang (2019) revisit the assumption of a bounded concentrability coefficient and justify the necessity of mild distribution shift via an information-theoretic lower bound. The authors show that, under a bounded concentrability coefficient defined using (6), near-optimal policy learning in polynomial sample complexity is precluded if the MDP dynamics are not restricted. In subsequent work, Xie and Jiang (2020) break the hardness conjecture introduced by Chen and Jiang (2019) albeit under a more restrictive concentrability coefficient similar to that of (4). Other works (Amortila et al., 2020; Wang et al., 2020; Zanette, 2020) have also proved hardness results for offline RL, however, under an even weaker form of concentrability than that induced by Eqs. (4) and (6). For example, Wang et al. (2020) show that good coverage over the feature space is not sufficient to sample-efficiently perform offline policy evaluation with linear function approximation and that significantly stronger assumptions on distributional shift may be needed. We refer to Xie and Jiang (2020) and Uehara and Sun (2021) for a detailed discussion on the relation between the different proposed concentrability coefficients and hardness results for offline RL.

Unfortunately, although the concentrability coefficients above attempt to quantify distributional shift under offline settings, they have limited interpretability. Specifically, it is hard to infer from the coefficients above which exact sampling distributions should be used. For example, if we consider coefficients C_2 and C_3, even if we know which parts of the state space are relevant according to the distribution ρ, the computation of (5) still depends on the complex interactions between ρ and the dynamics of the MDP under any possible policy. What can be concluded is that the concentrability coefficient will depend on all states that can be reached by any policy when the starting state distribution is given by ρ. However, it is not obvious which exact target distribution μ we should aim, especially in the face of uncertainty regarding the underlying MDP. In the face of such uncertainty, previous works assume sufficient coverage of the state (and action) space, thus using upper bounded coefficients to analyze the performance of the algorithms (Chen & Jiang, 2019; Farahmand et al., 2010; Munos & Szepesvári, 2008; Yang et al., 2019).

Finally, it is important to note that the significance of the previous results is highly dependent on the actual tightness of the bound (Munos, 2005); rather loose bounds can trivially upper bound the error but be of little help to understanding algorithmic behavior. Thus, it is important to understand, from a practical point of view, if the properties suggested by the surveyed bounds contribute to improved performance. We address this concern in Sect. 4, by empirically investigating how the data distribution impacts performance under a four-state MDP, as well as under high dimensional environments and two

Q-learning-based algorithms. For now, we further analyze, in the next section, the coefficients herein introduced by casting the problem of finding the optimal data distribution as an optimization problem.

3 Assessing the impact of data distribution through the lens of concentrability

In this section, we analyze the concentrability coefficients previously presented, providing insights as to what may constitute a good data distribution through the lens of concentrability. We give a new motivation for the use of maximum entropy data distributions from a game-theoretical point of view in Sect. 3.1 and study the optimization of concentrability coefficients in Sect. 3.2. We focus our attention on C_3 as: (i) it is not associated with a specific type of function approximation space, as opposed to other coefficients (Wang et al., 2020; ii) it does not directly impose assumptions on the dynamics of the MDP such as C_1; and (iii) Farahmand et al. (2010) suggest it allows for tighter bounds in comparison to coefficients similar to C_2. We consider the offline setting, where data distributions can be arbitrary, i.e., $\mu \in \Delta(\mathcal{S})$[2], and elaborate on how our analysis can extend to the online setting in Sect. 6. Putting all together, we study the impact of $\mu \in \Delta(\mathcal{S})$, for arbitrary $\rho \in \Delta(\mathcal{S})$, as quantified by

$$C_3(\mu;\rho) = (1-\gamma)^2 \sum_{m=0}^{\infty} m\gamma^{m-1} c(m;\mu,\rho), \tag{8}$$

$$c(m;\mu,\rho) = \max_{\pi_1,\ldots,\pi_m \in \Delta(\mathcal{A})^{|\mathcal{S}|}} \left\| \frac{\rho^\top P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m}}{\mu} \right\|_{2,\mu}, \tag{9}$$

and the convention that $\|\beta/\mu\|_{2,\mu} = \left(\sum_{s\in\mathcal{S}} \mu(s)(\beta(s)/\mu(s))^2 \right)^{1/2}$.

3.1 Maximum entropy distributions are adversarially robust in the face of uncertainty

For each m in (8), let $\beta_m = \rho P^{\pi_1} P^{\pi_2} \ldots P^{\pi_m}$. We have that $\left\|\beta_m/\mu\right\|_{2,\mu} = \sqrt{\mathcal{D}_f(\beta_m||\mu) + 1}$, for $f(x) = x^2 - 1$, where $\mathcal{D}_f$ denotes the f-divergence (Liese & Vajda, 2006). Optimizing C_3 over the distribution μ is non-trivial due to the fact that we want to minimize an expression involving multiple coefficients $\mathcal{D}_f(\beta_m||\mu)$, each with a β_m distribution that is chosen adversarially. Furthermore, the set of possible distributions β_m depends on the, usually unknown, transition probability function. Thus, we analyze the problem of picking an optimal μ as a robust optimization problem. We formulate a minimax objective where the minimizing player chooses μ to minimize $\mathcal{D}_f(\beta||\mu)$ and the maximizing player chooses $\beta \in \Delta(\mathcal{S})$ to maximize $\mathcal{D}_f(\beta||\mu)$. Essentially, our analysis assumes that $\beta_m \in \Delta(\mathcal{S})$ for all m (an upper-bound on C_3). We prove the following result, which characterizes the solution of such minimax objective.

[2] Our analysis can be similarly extended to the case where $\mu \in \Delta(\mathcal{S} \times \mathcal{A})$.

Proposition 1 *Let* $L_\mu : \Delta(\mathcal{S}) \to \mathbb{R}_0^+$ *with* $L_\mu(\beta) = \|\beta/\mu\|_{2,\mu}$. *For any* $\mu \in \Delta(\mathcal{S})$,

$$\max_{\beta \in \Delta(\mathcal{S})} L_\mu(\beta) = \max_{s \in \mathcal{S}} \mu(s)^{-1/2}.$$

Given the above, we have that the solution μ^* *to*

$$\arg\min_{\mu \in \Delta(\mathcal{S})} \max_{\beta \in \Delta(\mathcal{S})} L_\mu(\beta) \tag{10}$$

is the maximum entropy distribution over $\mathcal{S}$, *equivalent to the uniform distribution. Proof in Appendix A.*1.

As stated in Proposition 1, the uniform distribution is the solution to the robust optimization problem. This result provides a theoretical justification for the benefits of using high entropy sampling distributions through the lens of concentrability, as suggested by previous works (Kakade & Langford, 2002; Munos, 2003): in the face of uncertainty regarding the underlying MDP, high entropy distributions ensure coverage over the state-action space, thus contributing to keeping concentrability coefficients bounded.

3.2 Optimizing concentrability coefficients

We cast the problem of finding the optimal data distribution, $\mu^* \in \Delta(\mathcal{S})$, from the point of view of concentrability, as the optimization problem

$$\mu^* = \arg\min_{\mu \in \Delta(\mathcal{S})} C_3(\mu; \rho). \tag{11}$$

To solve the optimization problem above, we assume access to the transition probability function of the MDP, or an estimation thereof. In this section, we mainly use optimization as a tool to better understand what constitutes a good data distribution from the point of view of concentrability. Nevertheless, we believe our optimization procedure can find practical applications as the optimal estimated data distribution can be used to: (i) sample data if one has access to a simulator of the environment; or (ii) apply a reweighting scheme when sampling transitions from a dataset or replay buffer. We also elaborate on how to extend our algorithm to the online setting in Sec 6.

We display, in Algorithm 2, the optimization procedure we propose to estimate μ^*, where $\{\alpha_k\}_{k \in \{1,\ldots,K\}}$ is the set of learning rates. As can be seen, our algorithm starts from an initial distribution $\mu_0 \in \Delta(\mathcal{S})$ that is iteratively updated by: (i) calculating the best response of an adversary player, Π^*, where Π^* denotes the set of policies that maximize coefficients (9) for each m, given our current iterate μ_k (line 3); (ii) calculating the sub-gradient of our objective function at $\mu = \mu_k$, given Π^* (line 4); and (iii) updating μ_k while guaranteeing that $\mu_{k+1} \in \Delta(\mathcal{S})$ using a projection operator (line 5). In Appendix A.2, we provide a complete description of our algorithm.

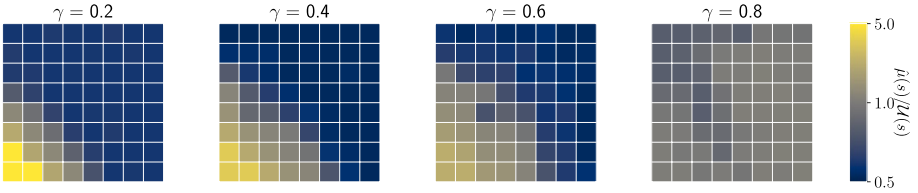


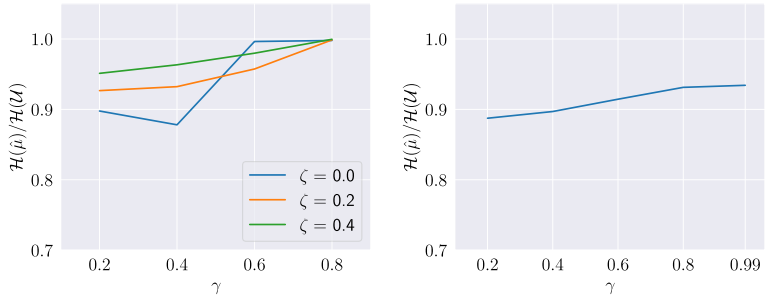
Fig. 1 Optimal data distributions $\hat{\mu}$ estimated by Algorithm 2 under the *grid* ζ environment for different γ values and $\zeta = 0.2$. The initial state distribution ρ puts all probability mass in the bottom left corner state. The colormap encodes, for each state s (grid cell), the proportion between $\hat{\mu}(s)$ and the probability of such state under the uniform distribution, $\mathcal{U}(s)$

Algorithm 2 Projected sub-gradient algorithm for optimizing $C_3(\mu;\rho)$.

$\mu_0 \in \Delta(\mathcal{S})$
for k in $\{1, \ldots, K\}$ **do**
  $\Pi^* = \arg\max_{\Pi} C_3(\mu_k, \Pi;\ \rho)$
  $g(\mu_k) = \nabla_{\mu} C_3(\mu, \Pi^*;\ \rho)|_{\mu=\mu_k}$
  $\mu_{k+1} = \mathrm{Proj}_{\Delta(\mathcal{S})}\left(\mu_k - \alpha_k g(\mu_k)\right)$
end for

We run Algorithm 2 to estimate the optimal data distribution $\hat{\mu}$ under two environments. The *grid* ζ environment (Appendix A.2.1), consists of a standard tabular environment where the agent can move between adjacent grid cells. Parameter ζ controls the stochasticity of the environment: an action succeeds with probability $(1 - \zeta)$, and with probability ζ the agent transitions to an arbitrary grid cell. In the *multi-path* environment (Appendix B.2.1), an agent needs to select a sequence of actions to reach a goal state while avoiding falling into an absorbing non-rewarding state. We consider both environments because their respective transition functions are different: in *grid* ζ, each state is reachable from any other state; this is not possible in the *multi-path* environment.

Under the *grid* ζ environment, we display, in Fig. 1, an illustration of distribution $\hat{\mu}$ as estimated by Algorithm 2 for $\zeta = 0.2$ and different γ values. As can be seen, for higher γ values, distribution $\hat{\mu}$ converges to the uniform distribution. However, it is noticeable that, for lower γ values, $\hat{\mu}$ concentrates in the bottom left corner of the grid, assigning higher probability to states that are closer, in terms of transitions in the underlying MDP, to the initial state distribution ρ. In Fig. 2a, we display the entropy of $\hat{\mu}$, $\mathcal{H}(\hat{\mu})$, for different γ and ζ values. As can be seen, the overall trend is that $\mathcal{H}(\hat{\mu})$ increases as γ increases, irrespective of ζ. First, the fact that $\hat{\mu}$ concentrates around ρ follows from the fact that γ geometrically discounts coefficients $c(m;\mu,\rho)$ in the calculation of C_3: lower values of γ make the optimization of $\hat{\mu}$ rather short-sighted. Nevertheless, we note that coverage is always ensured ($\min_{s\in\mathcal{S}} \hat{\mu}(s) = 0.0067 > 0$, across all γ and ζ values). Second, the fact that $\hat{\mu}$ converges to the uniform distribution as γ increases follows from two facts: (i) as γ increases, the optimization of $\hat{\mu}$ becomes rather long-sighted, putting progressively more emphasis on states that are farther away from ρ; (ii) since any state of the MDP is reachable from any other state, the adversary player that aims to find Π^*, which intuitively corresponds to the set of non-stationary policies that visit with high probability states where $\hat{\mu}$ is low, is always able to visit any state of the MDP (i.e., for sufficiently high m, we have that $c(m;\mu,\rho)$ can



(a) *Grid* ζ environment. (b) *Multi-path* environment.

Fig. 2 Normalized entropy of the data distributions $\hat{\mu}$ estimated by Algorithm 2 under the *grid* ζ and *multi-path* environments for different γ values

put probability mass on any state of the MDP). Thus, $\hat{\mu}$ converges to the uniform distribution as it is a best response to the adversary's behavior.

Regarding the *multi-path* environment, as shown in Fig. 2b, we observe that, similarly to the *grid* environment, the entropy of $\hat{\mu}$ increases as γ increases. However, as opposed to the *grid* environment, $\mathcal{H}(\hat{\mu})$ never reached the entropy of the uniform distribution across all tested γ values (we note the inclusion of an experimental setting with $\gamma = 0.99$). This observation is supported by the fact that, as opposed to the *grid* environment, the underlying transition probability function for the *multi-path* environment is such that the adversary is rather restricted by the dynamics of the MDP when aiming to visit states where $\hat{\mu}$ is low. For example, as γ increases and the optimization becomes rather long-sighted, states that are reachable in a small number of steps from ρ, but that there are unreachable in future timesteps (as is the case for the *multi-path* environment), become less important for objective C_3.

In this section, we observed that the analysis of what constitutes an optimal data distribution from the point of view of concentrability becomes rather intricate when we take into account the properties of the MDP. In particular, our results show that coverage is necessary and high entropy distributions contribute to keeping concentrability coefficients low, as discussed in Sects. 2.1 and 3.1. However, we provided evidence that better distributions exist and that such distributions depend on properties of the MDP such as the transition probability function and the discount factor γ. We display our complete experimental results in Appendix A.2.2.

4 Empirically assessing the impact of data distribution on *Q*-learning with function approximation

In this section, we empirically assess the impact of different properties of the data distribution on the performance of AVI-related algorithms. We start by showing, in Sect. 4.1, that the data distribution, indeed, plays an important role in regulating the performance of *Q*-learning-based algorithms: we propose a four-state MDP designed to highlight the impact of the data distribution on the performance of AVI-related methods.

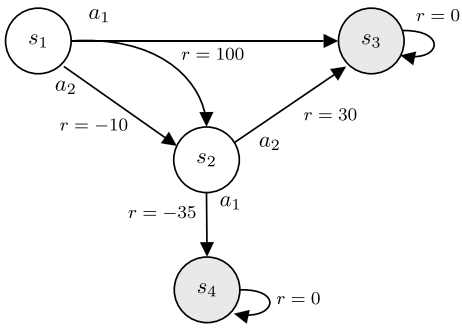


Fig. 3 Four-state MDP, with states $\{s_1, s_2, s_3, s_4\}$ and actions $\{a_1, a_2\}$. State s_1 is the initial state and states s_3 and s_4 are absorbing states. All actions are deterministic except for the state-action pair (s_1, a_1), where $p(s_3|s_1, a_1) = 0.99$ and $p(s_2|s_1, a_1) = 0.01$. The reward is $r(s_1, a_1) = 100$, $r(s_1, a_2) = -10$, $r(s_2, a_1) = -35$, and $r(s_2, a_2) = 30$

4.1 Four-state MDP

We now study how the data distribution influences the performance of a Q-learning algorithm with function approximation under the four-state MDP (Fig. 3). We show that the data distribution can significantly influence the quality of the resulting policies and affect the stability of the learning algorithm. Due to space constraints, we focus our discussion on the main conclusions and refer to Appendix B.1 for an in-depth discussion.

We focus our attention on non-terminal states s_1 and s_2 and set $\gamma = 1$. In state s_1 the optimal/correct action is a_1, whereas in state s_2 the optimal/correct action is a_2. We consider a linear function approximator $Q_w(s_t, a_t) = w^\top \phi(s_t, a_t)$, where ϕ is a feature mapping, defined as $\phi(s_1, a_1) = [1, 0, 0]^\top$, $\phi(s_1, a_2) = [0, 1, 0]^\top$, $\phi(s_2, a_1) = [\alpha, 0, 0]^\top$, and $\phi(s_2, a_2) = [0, 0, 1]^\top$, with $\alpha \in [1, 3/2)$. As can be seen, the capacity of the function approximator is limited and $Q_w(s_1, a_1)$ and $Q_w(s_2, a_1)$ are correlated.

4.1.1 Offline learning

We consider an offline RL setting and denote by μ the distribution over $\mathcal{S} \times \mathcal{A}$ induced by a static dataset of transitions. We focus our attention on probabilities $\mu(s_1, a_1)$ and $\mu(s_2, a_1)$, since these are the probabilities associated with the two partially correlated state-action pairs. Figure 4 displays the influence of the proportion between $\mu(s_1, a_1)$ and $\mu(s_2, a_1)$ on the number of correct actions yielded by the learned policy. We identify three regimes: (i) when $\mu(s_1, a_1) \approx 0.5$, we learn the optimal policy; (ii) if $\mu(s_1, a_1) < (\approx 0.48)$ or $(\approx 0.52) < \mu(s_1, a_1) < (\approx 0.65)$, the policy is only correct at one of the states; (iii) if $\mu(s_1, a_1) > (\approx 0.65)$, the policy is wrong at both states. The results show that, due to the limited approximation power and correlation between features, the data distribution impacts performance as the number of correct actions depends on the properties of μ. As our results show, due to bootstrapping, it is possible that under certain data distributions neither action is correct.

4.1.2 Online learning with unlimited replay

Instead of considering a fixed μ distribution, we now consider a setting where μ is dynamically induced by a replay buffer obtained using ϵ-greedy exploration. Figure 5 shows the results when $\alpha = 1.2$, under: (i) an ϵ-greedy policy with $\epsilon = 1.0$; and (ii) an ϵ-greedy policy with $\epsilon = 0.05$. We consider a replay buffer with unlimited capacity. We use a uniform data

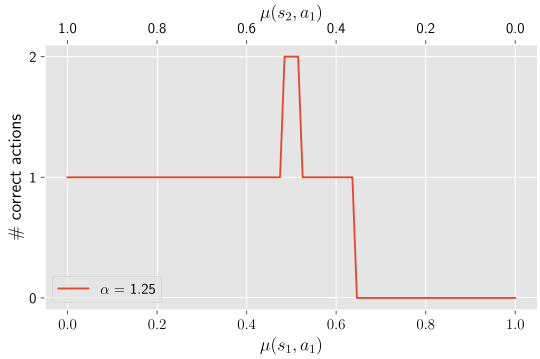
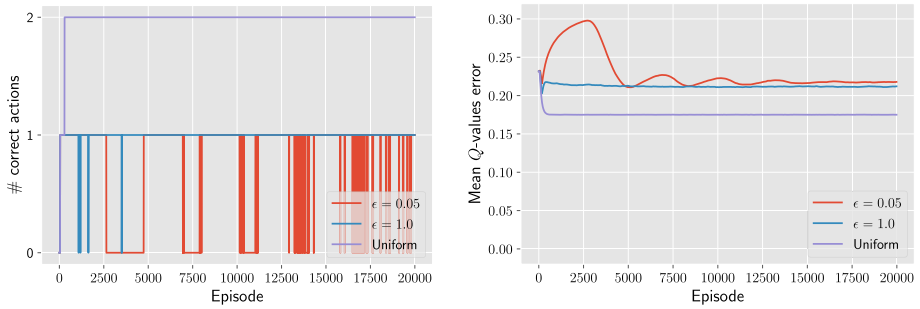


Fig. 4 The number of correct actions at states s_1 and s_2 for different data distributions ($\alpha = 1.25$)



(a) Number of correct actions. (b) Q-values mean error.

Fig. 5 Experiments for different exploratory policies (∞-sized replay buffer)

distribution as baseline.[3] As seen in Fig. 5, the baseline outperforms all other data distributions, as expected given our discussion in the previous section. Regarding the ϵ-greedy policy with $\epsilon = 1.0$, the agent is only able to pick the correct action at state s_1, featuring a higher average Q-value error in comparison to the baseline. This is due to the fact that the data distribution induced by the fully exploratory policy is too far from the uniform distribution to retrieve the optimal policy. Finally, for the ϵ-greedy policy with $\epsilon = 0.05$, the performance of the agent further deteriorates. Such policy induces oscillations in the Q-values (Fig. 5b), which eventually damp out as learning progresses. The oscillations are due to an undesirable interplay between the features and the data distribution: exploitation causes abrupt changes in the data distribution, hindering learning.

4.1.3 Discussion

We presented a set of experiments using a four-state MDP that shows how the data distribution can influence the performance of the resulting policies and the stability of the learning algorithm. First, we showed that, under an offline RL setting, the number of optimal actions identified is directly dependent on the properties of the data distribution due

[3] We note that the uniform distribution over the state-action space may be outside the space of possible distributions that can be generated by running policies on the MDP.

to an undesirable correlation between features. Second, not only the quality of the computed policies depends on the data collection mechanism, but also an undesirable interplay between the data distribution and the function approximator can arise: exploitation can lead to abrupt changes in the data distribution and hinder learning.

4.2 The impact of data distribution in offline RL

In this section, we experimentally assess the impact of different data distribution properties on the performance of offline DQN (Mnih et al., 2015) and CQL (Kumar et al., 2020). We evaluate the performance of the algorithms under six different environments: the *grid 1* and *grid 2* environments consist of standard tabular environments with highly uncorrelated state features, the *multi-path* environment is a hard exploration environment, and the *pendulum*, *mountaincar* and *cartpole* environments are benchmarking environments featuring a continuous state-space domain. All reported values are calculated by aggregating the results of different training runs. The description of the experimental environments and the experimental methodology, as well as the complete results, can be found in Appendix B.2. The developed software can be found at https://github.com/PPSantos/rl-data-distribution-public. We also provide an interactive dashboard with all our experimental results at https://rldatadistribution.pythonanywhere.com/.

In this section, we denote by μ the data distribution over state-action pairs induced by a static dataset of transitions. We consider two types of offline datasets: (i) ϵ-*greedy* datasets, generated by running an ϵ-optimal policy on the MDP, i.e., a policy that is ϵ-greedy with respect to the optimal Q-values, with $\epsilon \in [0, 1]$; and (ii) *Boltzmann*(T) datasets, generated by running a Boltzmann policy with respect to the optimal Q-values with temperature coefficient $T \in [-10, 10]$. Additionally, we artificially enforce that some of the generated datasets have full coverage over the $\mathcal{S} \times \mathcal{A}$ space. We do this by running an additional procedure that ensures that each state-action pair appears at least once in the dataset. We chose not to use publicly available datasets for offline RL (Fu et al., 2020; Gülçehre et al., 2020; Qin et al., 2021) in order to have complete control over the dataset generation procedure, which allows us to rigorously control different datasets' metrics and systematically compare our experimental results. Nevertheless, our results are representative of a diverse set of discrete action-space control tasks.

Two aspects are worth highlighting. First, in all environments, the sampling error is low due to the highly deterministic nature of the underlying MDPs. Thus, a single next-state sample is sufficient to correctly evaluate the Bellman optimality operator (Eq. (1)). Second, the function approximator has enough capacity to correctly represent the optimal Q-function, a property known as realizability (Chen & Jiang, 2019).

4.2.1 High entropy is beneficial

We start our analysis by studying the impact of the dataset distribution entropy, $\mathcal{H}(\mu)$, on the performance of the offline RL algorithms. Figure 6 displays the average normalized rollouts reward for datasets with different normalized entropies. As can be seen, under all environments and for both offline RL algorithms, high entropy distributions tend to achieve increased rewards. In other words, distributions with a large entropy appear to be well-suited to be used in offline learning settings. Such observation is inline with the discussion in Sect. 2.1 and works such as (Kakade & Langford, 2002; Munos, 2003): high

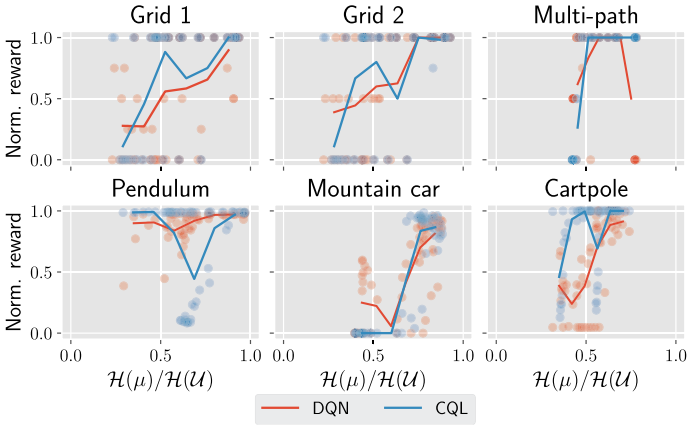


Fig. 6 Average rollouts reward for datasets with different entropies

entropy distributions contribute to increased coverage, keeping concentrability coefficients bounded and, thus, mitigating algorithmic instabilities.

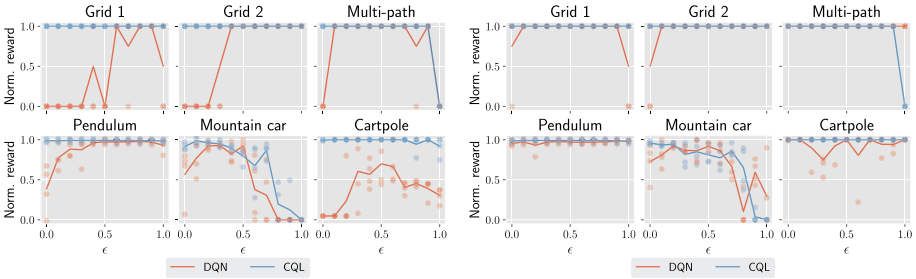
Importantly, we do not claim that high entropy distributions are the only distributions suitable to be used. As seen in Fig. 6, certain lower-entropy distributions also perform well. In the next sections, we investigate which other properties of the distribution are of benefit to offline RL.

4.2.2 Dataset coverage matters

We now study the impact of dataset coverage, i.e., the diversity of the transitions in the dataset, in the performance of the offline agents. In order to keep the discussion concise, in this section we focus our attention on ϵ-*greedy* datasets, and refer to Appendix B.2 for the complete results.

We start by focusing our attention on the offline DQN algorithm. Figure 7a displays the average normalized rollouts reward under ϵ-greedy datasets with dataset coverage not enforced. As can be seen, DQN struggles to achieve optimal rewards for low values of ϵ, i.e., even though the algorithm is provided with optimal or near-optimal trajectories, it is unable to steadily learn under such setting. However, as ϵ increases, the performance of the algorithm increases, eventually decaying again for high ϵ values. Such results suggest that a certain degree of data coverage is required by DQN to robustly learn in an offline manner, despite being provided with data rich in rewards. On the other hand, the decay in performance for highly exploratory policies under some environments can be explained by the fact that such policies induce trajectories that are poor in reward (this is further explored in the next section). Figure 7b displays the obtained experimental results under the exact same datasets, except that we enforce coverage over $\mathcal{S} \times \mathcal{A}$. We note an improvement in the performance of DQN across all environments, supporting our hypothesis that data coverage is important to regulating the stability of offline RL algorithms.

The CQL algorithm appears to perform more robustly than DQN. As seen in Fig. 7a, CQL is able to robustly learn with low ϵ values, i.e., using optimal or near-optimal trajectories with low coverage. Additionally, CQL does not benefit from coverage enforcement as DQN does, as seen in Fig. 7b.



(a) Dataset coverage not enforced. (b) Dataset coverage enforced.

Fig. 7 Average rollouts reward under ϵ-greedy datasets

The finding that data coverage appears to play an important role in regulating the performance of DQN, even when considering near-optimal trajectories, is in line with the discussion from Sect. 2.1. Even if we are only interested in correctly estimating the Q-values along an optimal trajectory, due to the bootstrapped nature of the updates, error in the estimation of the Q-values for adjacent states can erroneously affect the estimation of the Q-values along the optimal trajectory. This argument is suggested by concentrability coefficients: if ρ from (6) or (7) is the uniform distribution over the states of the optimal trajectory and zero otherwise, the concentrability coefficient still depends on other states than those of the optimal trajectory. Therefore, in order to keep the concentrability coefficient low, it is important that such states are present in the dataset. On the other hand, CQL is still able to robustly learn using high-quality trajectories under low coverage because of its pessimistic nature. Since CQL penalizes the Q-values for underrepresented actions in the dataset, the error for adjacent states is not propagated in the execution of the algorithm.

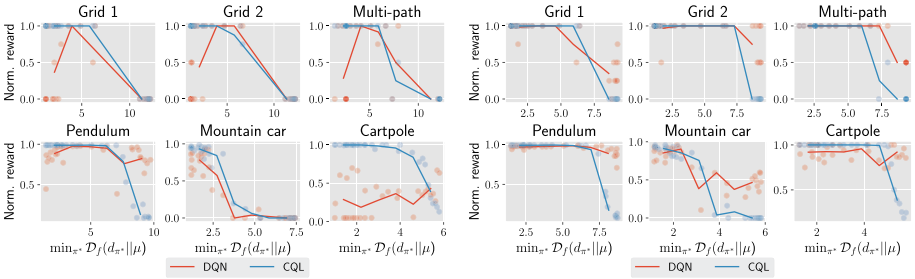
In this section, we considered datasets that are, in general, close to those induced by optimal policies. What if the data is collected by arbitrary policies? We investigate the impact of the trajectory quality in the next section.

4.2.3 Closeness to optimal policy matters

We now investigate how offline agents are affected by the quality of the trajectories contained in the dataset. More precisely, we study how the statistical distance between distribution μ and the closest distribution induced by one of the optimal policies of the MDP, d_{π^*}, affects offline learning.

The obtained experimental results are portrayed in Fig. 8, which shows the average normalized rollouts reward for different distances between μ and the closest d_{π^*}. We consider a wide spectrum of behavior policies, from optimal to anti-optimal policies (i.e., Boltzmann policies with low T values), as well as from fully exploitatory to fully exploratory policies. As can be seen, as the statistical distance between μ and the closest d_{π^*} increases, the lower the rewards obtained, irrespectively of the algorithm. We also observe an increase in rewards when dataset coverage is enforced (Fig. 8b) in comparison to when dataset coverage is not enforced (Fig. 8a), for both algorithms.

At first sight, our results appear intuitive if we focus on Fig. 8a, where dataset coverage is not enforced: if the policy used to collect the dataset is not good enough, it will fail to



(a) Dataset coverage not enforced. (b) Dataset coverage enforced.

Fig. 8 Average rollouts reward. The x-axis encodes the statistical distance between μ and the closest distribution d_{π^*} of one of the optimal policies

collect trajectories rich in rewards, key to learn reward-maximizing behavior. As an example, if the policy used to collect the data is highly exploratory, the agent will likely not reach high rewarding states and the learning signal may be too weak to learn an optimal policy.

However, the results displayed in Fig. 8b, in which dataset coverage is enforced, reveal a rather less intuitive finding: despite the fact that all datasets feature full coverage over $\mathcal{S} \times \mathcal{A}$, if the statistical distance between the two distributions is high, we observe a deterioration in algorithmic performance. In other words, despite the fact that the datasets contain all the information that can be retrieved from the environment (including transitions rich in reward), offline learning can still struggle if the behavior policy is too distant from the optimal policy. Such observation can be explained by the fact that distributions far from the optimal policy prevent the propagation of information, namely Q-values, during the execution of the offline RL algorithm.

Given the experimental results presented in this section, it is important for the data distribution to be aligned with that of optimal policies, not only to ensure that trajectories are rich in reward, but also to mitigate algorithmic instabilities. Our experimental results suggest that the assumption of a bounded concentrability coefficient, as discussed in Sect. 2.1, may not be enough to robustly learn in an offline manner and that more stringent assumptions on the data distribution are required. Wang et al. (2020) reach a similar conclusion, from a theoretical perspective.

4.2.4 Discussion

This section experimentally assessed the impact of different data distribution properties in the performance of offline Q-learning algorithms with function approximation, showing that the data distribution impacts algorithmic performance. In summary, our results show that: (i) high entropy data distributions are well-suited for learning in an offline manner; (ii) a certain degree of data diversity/coverage is desirable for offline learning; and (iii) a certain degree of data quality is desirable for offline learning.

Finding (i) is aligned with the discussion in Sect. 3.1: in the absence of detailed information regarding the underlying MDP, high entropy distributions contribute to high coverage over the state-action space, thus yielding bounded concentrability coefficients (an assumption widely adopted by the works surveyed in Sect. 2.1). However, as our experiments in Sect. 4.2.3 show, full coverage (equivalent to having bounded

Table 1 Performance metrics for the *grid 1* and *mountain car* environments under the DQN algorithm (coverage not enforced)

Environment	Dataset type	Avg. dataset coverage	Norm. avg. reward	Avg. Q-values error
Grid 1	*Uniform*	1.0	1.0	0.05
	Boltzmann(4.0)	0.92	1.0	0.04
	($\epsilon = 0.6$)-*greedy*	0.87	1.0	0.14
Mountain car	*Uniform*	1.0	0.94	1.98
	($\epsilon = 0.3$)-*greedy*	0.68	0.97	2.45
	Boltzmann(2.0)	0.61	0.91	3.49

For reference, the maximum average Q-values error recorded across all tested dataset types under the *grid 1* and *mountain car* environments are, respectively, 2.45×10^5 and 50.95

concentrability coefficients) is not enough to learn optimal policies. Thus, we hypothesize that the advantages of using high entropy distributions not only come from the fact that they yield high coverage over the state-action space, but also because they induce smooth distributions that mitigate information bottlenecks during algorithm execution, allowing Q-values to easily propagate according to the MDP dynamics. This hypothesis is supported by Proposition 1, which shows that maximum entropy distributions minimize the statistical distance to all other possible distributions.

Regarding finding (ii), a certain degree of coverage is necessary, even when the data is collected by an optimal policy, due to bootstrapping, as discussed in Sect. 4.2.2. However, according to our results (Table 1), it is not necessary to have full coverage over the state-action space to learn optimal behavior. This finding is supported by the discussion in Sect. 2.1: as suggested by (5) and (6), given ρ (e.g., uniform over the state-action pairs of an optimal trajectory), the importance of the different states depends not only on the dynamics of the MDP but also on their distance, in terms of the number of transitions, to the states in ρ. This contrasts with the works surveyed in Sect. 2.1 that, in the absence of knowledge regarding the MDP's dynamics and the quality of the trajectories provided, assume bounded concentrability coefficients, i.e., full coverage over the $\mathcal{S} \times \mathcal{A}$ space.

Finally, according to finding (iii), it appears to also be important that the distribution induced by the dataset is not very far from the distribution induced by one of the optimal policies of the MDP, even if all state-action pairs are present in the dataset. Again, we hypothesize that this finding can be explained by the fact that certain distributions prevent the propagation of Q-values throughout the iterations of the algorithm. Further investigations should be carried out in order to understand if this problem can be circumvented by using more sophisticated sampling techniques such as prioritized replaying. We leave such research direction for future work.

We refer to Appendix B.2.5 for an additional discussion on the impact of the sampling error, approximation capacity, and generalization hardness in our experiments. We also elaborate on how our experiments illustrate the importance of trading off algorithmic optimism and pessimism in offline RL.

5 Related work

We now review different lines of research that are related to the problem of studying the impact of data distribution on Q-learning-related methods.

5.1 Error propagation in AVI

On a theoretical side, there are a number of different works that analyze error propagation in AVI methods, deriving error bounds on the performance of AVI-related algorithms (Chen & Jiang, 2019; Munos, 2005; Munos & Szepesvári, 2008; Yang et al., 2019) algorithms. Common to all these works is the dependence of the derived bound on concentrability coefficients that depend on the data distribution. In this work, we review concentrability coefficients in Sect. 2.1, provide a motivation for the use of high entropy data distributions through the lens of robust optimization in Sect. 3.1, and study the optimization of concentrability coefficients in Sect. 3.2. In Sect. 4.2, we analyze our empirical results in light of the theoretical results from these previous articles.

5.2 Unstable behavior in off-policy learning

Several early studies analyze the unstable behavior of off-policy learning algorithms and the harmful learning dynamics that can lead to the divergence of the function parameters (Baird, 1995; Kolter, 2011; Tsitsiklis & van Roy, 1996; Tsitsiklis & Van Roy, 1997). For instance, Baird (1995), Tsitsiklis and van Roy (1996), Kolter (2011) provide examples that highlight the unstable behavior of AVI methods. Kolter (2011) provides an example that highlights the dependence of the off-policy distribution on the approximation error of the algorithm. In Sect. 4.1, we propose a novel four-state MDP that highlights the impact of the data distribution in the performance of AVI methods. We further explore how off-policy algorithms are affected by data distribution changes, under diverse settings. We add to previous works by considering both offline settings comprising static data distributions, and online settings in which data distributions are induced by a replay buffer.

5.3 The stability of deep and offline RL algorithms

Several works investigate the stability of deep RL methods (Fu et al., 2019; Kumar et al., 2019, 2020; Liu et al., 2018; van Hasselt et al., 2018; Wang et al., 2021; Zhang et al., 2021), as well as the development of RL methods specifically suited for offline settings (Agarwal et al., 2019; Levine et al., 2020; Mandlekar et al., 2021). For example, Kumar et al. (2020) observe that Q-learning-related methods can exhibit pathological interactions between the data distribution and the policy being learned, leading to potential instability. Fu et al. (2019) investigate how different components of DQN play a role in the emergence of the deadly triad. In particular, the authors assess the performance of DQN with different sampling distributions, finding that higher entropy distributions tend to perform better. Agarwal et al. (2019) provide a set of ablation studies that highlight the impact of the dataset size and diversity in offline learning settings. Wang et al. (2021) study the stability of offline policy evaluation, showing that even under relatively mild distribution shift, substantial error amplification can occur. In Sect. 4.2, we provide a systematic study on

how different properties of the data distribution impact the performance of deep offline RL algorithms by directly controlling the dataset generation process, allowing us to rigorously control different datasets' metrics and systematically compare our experimental results. We validate and extend previous results, as well as discuss our experimental findings in light of existing theoretical results.

Schweighofer et al. (2021) study the impact of dataset characteristics on offline RL by studying the influence of the average dataset return and state-action coverage on the performance of RL algorithms. Yarats et al. (2022) study the impact of data quality on the ability of offline RL algorithms to generalize to new tasks. The authors compare unsupervised exploration methods for data collection and then assess the quality of the collected data by learning policies for different reward functions, highlighting the importance of having diverse exploratory data. Concurrently, Lambert et al. (2022) propose a similar framework for unsupervised exploration followed by offline RL, being particularly focused on intrinsic motivation techniques for data collection. Despite some similarities with our Sect. 4.2, our work provides a more refined analysis because, as opposed to the previous works which consider datasets that are collected by running policies in the MDP, we have additional control over the data generation process, for example, by being able to enforce full dataset coverage. This allows for a more exhaustive and precise analysis of the impact of different properties of the data distribution in algorithmic performance. Additionally, the calculation of the dataset metrics and the types of environments used also differ. Finally, we note that we present a much broader picture regarding the impact of the data distribution on the stability of general off-policy RL algorithms, analyzing our empirical results in light of theoretical results.

Concurrently to our study, Al-Marjani et al. (2023) propose the notion of active coverage, where an agent aims to explore the environment such that the number of visits to state-action pairs is lower-bounded by a given target vector. The authors propose CovGame, an exploration algorithm based on the notion of active coverage. Our work shares some similarities with that of Al-Marjani et al. (2023) since CovGame can be seen as aiming to explore the state-action space such that the induced data distribution is approximately optimal from the point of view of concentrability (albeit under a slightly different notion of concentrability than those we introduce in our work). However, there are key differences between both works: (i) Al-Marjani et al. (2023) examine the episodic RL setting while we address the infinite-horizon discounted setting; (ii) while Al-Marjani et al. (2023) focus on online RL, our focus lies on the offline setting; and (iii) the algorithm proposed by Al-Marjani et al. (2023) solves a sequence of min-max games by leveraging two online learning algorithms while our algorithm, in the offline setting, starts from an initial arbitrary data distribution that is iteratively updated using a projected sub-gradient algorithm to minimize a loss function given by a concentrability coefficient.

6 Conclusion

In this work, we investigate the interplay between the data distribution and Q-learning-based algorithms with function approximation under offline settings, connecting different lines of research and validating and extending previous results. First, we study the optimization of concentrability coefficients. We show that: (i) maximum entropy distributions are adversarially robust in the face of uncertainty; and (ii) better distributions than the uniform distribution exist, which depend on properties of the MDP. Second, from an empirical

perspective, we study how different properties of the data distribution impact algorithmic performance. In particular, we: (i) provide a four-state MDP to showcase the algorithmic instabilities that may arise due to the interplay between the data distribution and the learning algorithm; and (ii) experimentally assess the impact of the data distribution on two offline Q-learning-based algorithms, attesting to the importance of different properties of the data distribution such as entropy, coverage, and data quality. We connect our empirical results with the theoretical findings of previous works.

In light of our work, future work could comprise the development of improved data processing techniques for offline RL. As suggested by our empirical results: (i) naive dataset concatenation can lead to deterioration in performance; and (ii) by simply reweighting or discarding training data, it is possible to substantially improve the performance of offline RL algorithms. Furthermore, it would be interesting to empirically test whether a reweighting scheme based on the optimal data distribution estimated by our Algorithm 2 can improve the performance of RL algorithms that rely on replay buffering. Finally, we envision that the algorithm we propose in Sect. 3.2 can also foster the development of new theoretically grounded exploration techniques for online RL. To conclude, we now further elaborate on how our findings can extend to the online setting, as well as investigate how to construct exploratory policies that induce state-action distributions with low concentrability coefficients.

Now, we focus our attention on the online RL setting where, as opposed to the offline setting, the learning agent is able to collect data by interacting with the environment. Under the offline setting, the data distribution can be arbitrary; hence, we assumed $\mu \in \Delta(\mathcal{S})$ in Sect. 3 of our work. However, under the online RL setting, the problem of finding the optimal data distribution becomes rather restricted because the set of possible state distributions induced by policies in the MDP may be only a subset of $\Delta(\mathcal{S})$.

In fact, other complexity measures than concentrability coefficients were proposed to quantify sample complexity under the online setting (Jin et al., 2021; Jiang et al., 2016). Interestingly, Xie et al. (2022) recently showed how assumptions about the data distribution in the offline setting can imply sample-efficient online learning. We take inspiration from such work and now analyze how to come up with exploratory policies that induce state distributions that yield low concentrability coefficients.

The key problem preventing us from extending our analysis in Sect. 3 to the online setting is that both the uniform distribution $\mathcal{U}$, which we have shown to be min-max optimal in the face of uncertainty, as well as the optimal data distribution $\hat{\mu}$ estimated by our Algorithm 2, may not be contained in the set of possible state distributions induced by policies in the MDP. Thus, we propose to use a state marginal matching approach (Lee et al., 2019) to find a policy π which induced state distribution d_π is as close as possible, in terms of a KL-divergence $\mathcal{D}_{\mathrm{KL}}$, to our target distribution t (either $\mathcal{U}$ or $\hat{\mu}$), i.e., we solve $\arg\min_\pi \mathcal{D}_{\mathrm{KL}}(d_\pi | t)$. Thus, we envision that by exploiting the transition probability function of the MDP, either the true or an estimated version thereof, we can construct an approximately optimal exploratory policy from the point of view of concentrability by: (i) estimating the optimal data distribution $\hat{\mu}$ using Algorithm 2; and (ii) using a state marginal matching approach to find a policy which induced distribution is close to $\hat{\mu}$.

Finally, in the face of uncertainty regarding the underlying MDP, we have shown that the uniform distribution $\mathcal{U}$ is the solution to a robust optimization problem. If we aim to find policy π such that d_π minimizes $\mathcal{D}_{\mathrm{KL}}(d_\pi | \mathcal{U})$, we have that such problem is equivalent to finding the policy which d_π has the highest entropy. Formally, we have that $\arg\min_\pi \mathcal{D}_{\mathrm{KL}}(d_\pi | \mathcal{U}) = \arg\max_\pi \mathcal{H}(d_\pi)$. This provides a clear justification for the use of

maximum entropy state exploration methods (Hazan et al., 2018) for the construction of datasets for offline RL from the point of view of concentrability.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10994-024-06564-5.

Author Contributions Pedro P. Santos, Alberto Sardinha, and Francisco S. Melo, have all contributed to all parts of the research (theory, experiments, and writing). Diogo S. Carvalho contributed to the theoretical part of the article.

Funding Open access funding provided by FCT|FCCN (b-on). This work was supported by Portuguese national funds through the Portuguese Fundação para a Ciência e a Tecnologia (FCT) under projects UIDB/50021/2020 (INESC-ID multi-annual funding), PTDC/CCI-COM/5060/2021 (RELEvaNT), and PTDC/CCI-COM/7203/2020 (HOTSPOT). In addition, this research was supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No. 952215, and by the Air Force Office of Scientific Research under award number FA9550-22-1-0475. Pedro P. Santos acknowledges the FCT PhD grant 2021.04684.BD, and Diogo S. Carvalho the FCT PhD grant 2020.05360.BD.

Data Availability The experimental data is available in a csv file https://github.com/PPSantos/rl-data-distribution-public and in a dashboard https://rldatadistribution.pythonanywhere.com/.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Code availability The developed code is publicly available https://github.com/PPSantos/rl-data-distribution-public.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems.

Agarwal, R., Schuurmans, D., & Norouzi, M. (2019). An optimistic perspective on offline reinforcement learning. CoRR arxiv:1907.04543

Al-Marjani, A., Tirinzoni, A., & Kaufmann, E. (2023). Active coverage for pac reinforcement learning.

Amortila, P., Jiang, N., & Xie, T. (2020). A variant of the Wang-Foster-Kakade lower bound for the discounted setting. CoRR arxiv:2011.01075.

Antos, A., Szepesvari, C., & Munos, R. (2008). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning, 71*, 89–129.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866.

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the twelfth international conference on machine learning* (pp. 30–37). Morgan Kaufmann.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Carvalho, D., Melo, F. S., & Santos, P. (2020). A new convergent variant of q-learning with linear function approximation. *Advances in Neural Information Processing Systems, 33*, 19412–19421.

Chen, J., & Jiang, N. (2019). Information-theoretic considerations in batch reinforcement learning. CoRR arxiv:1905.00360.
Farahmand, A. M., Szepesvári, C., & Munos, R. (2010). Error propagation for approximate policy and value iteration. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems* (Vol. 23).
Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4RL: Datasets for deep data-driven reinforcement learning. CoRR arxiv:2004.07219.
Fu, J., Kumar, A., Soh, M., & Levine, S. (2019). Diagnosing bottlenecks in deep q-learning algorithms. CoRR arxiv:1902.10250.
Gülçehre, Ç., Wang, Z., Novikov, A., Paine, T. L., Colmenarejo, S. G., Zolna, K., & de Freitas, N. (2020). RL unplugged: Benchmarks for offline reinforcement learning. CoRR arxiv:2006.13888.
Hazan, E., Kakade, S. M., Singh, K., & Soest, A. V. (2018). Provably efficient maximum entropy exploration. CoRR arxiv:1812.02690.
Held, M., Wolfe, P., & Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical Programming, 6*, 62–88.
Hoffman, M., Shahriari, B., Aslanides, J., Barth-Maron, G., Behbahani, F., Norman, T., & de Freitas, N. (2020). Acme: A research framework for distributed reinforcement learning.
Horst, R., Pardalos, P., & Van Thoai, N. (2000). *Introduction to global optimization*. Springer US.
Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., & Schapire, R. E. (2016). Contextual decision processes with low bellman rank are pac-learnable. CoRR arxiv:1610.09512.
Jin, C., Liu, Q., & Miryoosefi, S. (2021). Bellman eluder dimension: New rich classes of RL problems, and sample-efficient algorithms. CoRR arxiv:2102.00815.
Kakade, S., & Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *Proceedings 19th international conference on machine learning* (pp. 267–274).
Kolter, J. (2011). The fixed points of off-policy td. In *Advances in neural information processing systems* (Vol. 24).
Kumar, A., Fu, J., Tucker, G., & Levine, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. CoRR arxiv:1906.00949.
Kumar, A., Gupta, A., & Levine, S. (2020). Discor: Corrective feedback in reinforcement learning via distribution correction. CoRR arxiv:2003.07305.
Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative q-learning for offline reinforcement learning. CoRR arxiv:2006.04779.
Lambert, N., Wulfmeier, M., Whitney, W. F., Byravan, A., Bloesch, M., Dasagi, V., & Riedmiller, M. A. (2022). The challenges of exploration for offline reinforcement learning. CoRR arxiv:2201.11861.
Lazaric, A., Ghavamzadeh, M., & Munos, R. (2012). Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research, 13*(98), 3041–3074.
Lazaric, A., Ghavamzadeh, M., & Munos, R. (2016). Analysis of classification-based policy iteration algorithms. *Journal of Machine Learning Research, 17*(19), 1–30.
Lee, L., Eysenbach, B., Parisotto, E., Xing, E. P., Levine, S., & Salakhutdinov, R. (2019). Efficient exploration via state marginal matching. CoRR arxiv:1906.05274.
Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. CoRR arxiv:2005.01643.
Liese, F., & Vajda, I. (2006). On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory, 52*(10), 4394–4412.
Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. CoRR arxiv:1509.02971.
Liu, V., Kumaraswamy, R., Le, L., & White, M. (2018). The utility of sparse representations for control in reinforcement learning. CoRR arxiv:1811.06626.
Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., & Martín-Martín, R. (2021). What matters in learning from offline human demonstrations for robot manipulation. CoRR arxiv:2108.03298.
Mangasarian, O. L., & Shiau, T. H. (1986). A variable-complexity norm maximization problem. *SIAM Journal on Algebraic Discrete Methods, 7*(3), 455–461.
Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2015). Playing atari with deep reinforcement learning. *Nature, 518*(7540), 529–533.
Munos, R. (2003). Error bounds for approximate policy iteration. In *International conference on machine learning* (Vol. 3, pp. 560–567).
Munos, R. (2005). Error bounds for approximate value iteration. In *Aaai conference on artificial intelligence* (pp. 1006–1011).

Munos, R. (2007). Performance bounds in L_p norm for approximate value iteration. *SIAM Journal on Control and Optimization, 46*(2), 541–561.
Munos, R., & Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research, 9*(27), 815–857.
Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.
Qin, R., Gao, S., Zhang, X., Xu, Z., Huang, S., Li, Z., & Yu, Y. (2021). Neorl: A near real-world benchmark for offline reinforcement learning. CoRR arxiv:2102.00714.
Riedmiller, M. (2005). Neural fitted q iteration – first experiences with a data efficient neural reinforcement learning method. In *Machine learning: Ecml 2005* (pp. 317–328).
Schweighofer, K., Hofmarcher, M., Dinu, M., Renz, P., Bitto-Nemling, A., Patil, V. P., & Hochreiter, S. (2021). Understanding the effects of dataset characteristics on offline reinforcement learning. CoRR arxiv:2111.04714.
Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature, 550*(7676), 354–359.
Sutton, R., & Barto, A. (2018). *Reinforcement learning: An introduction* (2nd ed.). The MIT Press.
Tosatto, S., Pirotta, M., D'Eramo, C., & Restelli, M. (2017). Boosted fitted q-iteration. In *Proceedings of the 34th international conference on machine learning* (pp. 3434–3443).
Tsitsiklis, J., & Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control, 42*(5), 674–690.
Tsitsiklis, J. N., & van Roy, B. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning, 22*(1), 59–94.
Uehara, M., & Sun, W. (2021). Pessimistic model-based offline RL: PAC bounds and posterior sampling under partial coverage. CoRR arxiv:2107.06226.
van Hasselt, H., Doron, Y., Strub, F., Hessel, M., Sonnerat, N., & Modayil, J. (2018). Deep reinforcement learning and the deadly triad. CoRR arxiv:1812.02648.
Wang, R., Foster, D. P., & Kakade, S. M. (2020). What are the statistical limits of offline RL with linear function approximation? CoRR arxiv:2010.11895.
Wang, R., Wu, Y., Salakhutdinov, R., & Kakade, S. M. (2021). Instabilities of offline RL with pre-trained neural representation. CoRR arxiv:2103.04947.
Xie, T., Foster, D. J., Bai, Y., Jiang, N., & Kakade, S. M. (2022). The role of coverage in online reinforcement learning.
Xie, T., & Jiang, N. (2020). Batch value-function approximation with only realizability. CoRR arxiv:2008.04990.
Yang, Z., Xie, Y., & Wang, Z. (2019). A theoretical analysis of deep q-learning. CoRR arxiv:1901.00137.
Yarats, D., Brandfonbrener, D., Liu, H., Laskin, M., Abbeel, P. , Lazaric, A., & Pinto, L. (2022). Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. CoRR arxiv:2201.13425.
Zanette, A. (2020). Exponential lower bounds for batch reinforcement learning: Batch RL can be exponentially harder than online RL. CoRR arxiv:2012.08005.
Zhang, S., Yao, H., & Whiteson, S. (2021). Breaking the deadly triad with a target network. CoRR arxiv:2101.08862.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.