Provably (More) Sample-Efficient Offline RL with Options

Xiaoyan Hu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong SAR, China
xyhu21@cse.cuhk.edu.hk

Ho-fung Leung
Independent Researcher
Hong Kong SAR, China
ho-fung.leung@outlook.com

Abstract

The options framework yields empirical success in long-horizon planning problems of reinforcement learning (RL). Recent works show that options improve the sample efficiency in online RL where the learner can actively explore the environment. However, these results are no longer applicable to scenarios where exploring the environment online is risky, e.g., automated driving and healthcare. In this paper, we provide the first analysis of the sample complexity for offline RL with options, where the agent learns from a dataset without further interaction with the environment. We propose the PEssimistic Value Iteration for Learning with Options (PEVIO) algorithm and establish near-optimal suboptimality bounds (with respect to the novel information-theoretic lower bound for offline RL with options) for two popular data-collection procedures, where the first one collects state-option transitions and the second one collects state-action transitions. We show that compared to offline RL with actions, using options not only enjoys a faster finite-time convergence rate (to the optimal value) but also attains a better performance (when either the options are carefully designed or the offline data is limited). Based on these results, we analyze the pros and cons of the data-collection procedures, which may facilitate the selection in practice.

1 Introduction

Planning in long-horizon tasks is challenging in reinforcement learning (RL) (Co-Reyes et al., 2018; Eysenbach et al., 2019; Hoang et al., 2021). A line of study proposes to accelerate learning in these tasks using temporally-extended actions (Fikes et al., 1972; Sacerdoti, 1973; Drescher, 1991; Jiang et al., 2019; Nachum et al., 2019; Machado et al., 2021; Erraqabi et al., 2022). One powerful approach is the options framework introduced by Sutton et al. (1999), where the agent interacts with the environment with closed-loop policies called options. Empirical success (Tessler et al., 2017; Vezhnevets et al., 2017) shows that options help achieve sample-efficient performance in long-horizon planning problems.

To provide a theoretical guarantee to the options framework, recent works have focused on the sample complexity of RL with options in the online setting, where the agent continuously explores the environment and learns a hierarchical policy to select options. Brunskill and Li (2014) establish a PAC-like sample complexity of RL with options in the semi-Markov decision processes (SMDPs), where temporally-extended actions are treated as indivisible and unknown units. Later, Fruit and Lazaric (2017) provide the first regret analysis of RL with options under the Markov decision processes (MDPs) framework. While their proposed algorithm attains a sublinear regret, it requires prior knowledge of the environment, which is not usually available in practice. To address this problem, Fruit et al. (2017) propose an algorithm that does not require prior knowledge, yet achieves
a near-optimal regret bound. However, these results are inapplicable to many real-world scenarios where online exploration is not allowed. For example, it has been argued that in healthcare (Gottesman et al., 2016) and automated driving (Shalev-Shwartz et al., 2016), learning in an online manner is risky and costly. In these scenarios, offline learning, where the agent learns a policy from a dataset, is preferred. We note that there is a line of studies on the sample complexity of offline RL with primitive actions only (i.e., without the use of options) (Levine et al., 2020; Fu et al., 2020; Rashidinejad et al., 2021). Unfortunately, to the best of our knowledge, there have been no results reported on the offline RL with options.

In this paper, we make the following contributions. First, we derive a novel information-theoretic lower bound, which generalizes the one for offline learning with actions. Second, we propose the PEssimistic Value Iteration for Learning with Options (PEVIO) algorithm and derive near-optimal suboptimality bounds for two popular data-collection procedures, where the first one collects state-option transitions and the second one collects state-action transitions. More importantly, we show that options facilitate more sample-efficient learning in both the finite-time convergence rate and actual performance. To shed light on offline RL with options in practice, we discuss the pros and cons of both data-collection procedures based on our analysis.

2 Related Work

Learning with Options Building upon the theory of semi-Markov decision processes (SMDPs) (Bradtke and Duff, 1994; Mahadevan et al., 1997; Sutton et al., 1999) propose to learn with options. Following their seminal work, learning with options has been widely studied in the function approximation setting (Sorg and Singh, 2010) and hierarchical RL (Igl et al., 2020; Klissarov and Precup, 2021; Wulfmeier et al., 2021). Discovering useful options has also been the subject of extensive research (Stolle and Precup, 2002; Riemer et al., 2018; Mankowitz et al., 2018; Harb et al., 2018; Hirooka et al., 2019; Bagaria et al., 2021). Despite its empirical success, there have been fairly limited studies on the sample efficiency of learning with options. Brunskill and Li (2014) analyze the sample complexity bound for an RMAX-like algorithm for SMDPs. Fruit and Lazaric (2017) derive the first regret analysis of learning with options. They propose an algorithm that attains sublinear regret in the infinite-horizon average-reward MDP while requiring prior knowledge of the environment. Later, Fruit et al. (2017) remove this requirement.

Offline RL In the offline setting, a dataset that is collected by executing a behavior policy in the environment is provided, and the agent is asked to learn a near-optimal policy using only this dataset. A key challenge in offline RL is the insufficient coverage of the dataset (Wang et al., 2021), which is also known as distributional shift (Chen and Jiang, 2019; Levine et al., 2020). To address this problem, the previous study on sample-efficient learning assumes uniform coverage of the dataset (Liu et al., 2018; Chen and Jiang, 2019; Jiang and Huang, 2020; Yang et al., 2020; Xie and Jiang, 2020; Uehara et al., 2020; Qu and Wierman, 2020; Yin et al., 2021). This assumption is relaxed in recent works by pessimism principle (Xie et al., 2021; Rashidinejad et al., 2021; Jin et al., 2021).

3 Preliminaries

3.1 Episodic MDP with Options

Let $\Delta(A)$ denote the probability simplex on space $A$ and $[N] := \{1, \ldots, N\}$ for any positive integer $N$. An episodic MDP with options is a sextuple $\mathcal{M} = (S, A, O, H, P, r)$, where $S$ is the state space, $A$ the (primitive) action set, $O$ the finite set of options, $H$ the length of each episode, $P = \{P_h : S \times A \to \Delta(S)\}_{h \in [H]}$ the transition kernel, $r = \{r_h : S \times A \to [0, 1]\}_{h \in [H]}$ the deterministic reward function. We define $S := |S|$, $A := |A|$, and $O := |O|$. A (Markov) option $o \in O$ is a pair $(\pi^o, \beta^o)$ where $\pi^o = \{\pi_h^o : S \to \Delta(A)\}_{h \in [H]}$ is the option’s policy and $\beta^o = \{\beta^o_h : S \to [0, 1]\}_{h \in [H]}$ is the probability of the option’s termination. For convenience, we define $\beta^o_{H+1}(s) = 1$ for all $(s, o) \in S \times O$, i.e., any option is terminated after the end of an

---

1 Our results can be directly generalized to stochastic rewards.
episode. We assume that the initial state $s_1$ is fixed\(^3\). Upon arriving at state $s_h$ at any timestep $h \in [H]$, if $h = 1$ (at the beginning of an episode), the agent selects option $o_1 \sim \mu_1(s_1)$, where $\mu = \{\mu_h : S \mapsto \Delta(O)\}_{h \in [H]}$ is a hierarchical policy to select an option at each state. Otherwise ($h \geq 2$), the agent first terminates option $o_{h-1}$ with probability $\beta_h^{o_{h-1}}(s_h)$. If option $o_{h-1}$ is terminated, she then selects a new option $o_h \sim \mu_h(\cdot|s_h)$ according to the hierarchical policy $\mu$. If option $o_{h-1}$ is not terminated, the agent continues to use option $o_{h-1}$ at timestep $h$, i.e., $o_h = o_{h-1}$. After that, the agent takes action $a_h \sim \pi_h^{o_h}(\cdot|s_h)$, receives a reward $r_h := r_h(s_h, a_h)$, and transits to the next state $s_{h+1} = P_h(\cdot|s_h, a_h)$. An episode terminates at timestep $H + 1$. A special case is that an action $a$ is an option $o$, such that $\pi_h^o(a|s) = 1$ and $\beta_h^o(s) = 1$ for any $(h, s) \in [H] \times S$. For convenience, we use the notation $O = \mathcal{A}$ to represent that each option corresponds to an action, which is the case in RL with primitive actions.

To define the $Q$-function and the value function, we introduce some useful notations\(^4\): Let $T = \{T_h : S \times O \mapsto \Delta(S \times [H + 1])\}_{h \in [H]}$ and $U = \{U_h : S \times O \mapsto [0, H]\}_{h \in [H]}$ denote the option transition function and the option utility function, respectively. Particularly, for any $(h, s, o) \in [H] \times S \times O$, the option transition function $T_h(s'|s, o, \tau)$ is the probability that the agent uses option $o$ at state $s$ at timestep $h$, reaches state $s'$ at timestep $h + \tau$ without terminating option $o$ in these $\tau$ timesteps, and finally terminates option $o$ at state $s'$ at timestep $h + \tau$. The option utility function $U_h(s, o)$ is the expected cumulative reward within timesteps that the option is used without being terminated. Given any arbitrary series of functions $\{y_h : S \mapsto \mathbb{R}\}_{h \in [H]}$, define the operator $[T_h y_{h+\tau}](s, o) := \sum_{\tau'} T_h(s'|s, o, \tau') y_{h+\tau'}(s')$ for any $(s, o, \tau) \in S \times O \times [H - h + 1]$. In the following, we derive the $Q$-function and the value function for learning with options. (The detailed proof can be found in Appendix B.)

**Theorem 1 (Q-function and value function).** For any hierarchical policy $\mu$ and $(h, s, o) \in [H] \times S \times O$, the $Q$-function is given by

$$
Q^\mu_h(s, o) := \mathbb{E}_\mu \left[ \sum_{h' = h}^H r_{h'}(s_{h'}, o_{h'}) \bigg| s_h = s, o_h = o \right] = U_h(s, o) + \sum_{\tau \in [H - h + 1]} [T_h V^\mu_{h+\tau}](s, o)
$$

and the value function is given by

$$
V^\mu_h(s) := \mathbb{E}_\mu \left[ \sum_{h' = h}^H r_{h'}(s_{h'}, o_{h'}) \bigg| s_h = s, o_h \sim \mu_h(\cdot|s_h) \right] = \sum_{o \in O} \mu_h(o|s) Q^\mu_h(s, o)
$$

where $V^\mu_{H+1}(s) = Q^\mu_{H+1}(s, o) = 0$ for any $(s, o) \in S \times O$.

Intuitively, the first term $U_h(s, o)$ of the $Q$-function is the expected reward within timesteps that option $o$ is used without being terminated, and the second term $\sum_{\tau \in [H - h + 1]} [T_h V^\mu_{h+\tau}](s, o)$ corresponds to the expected reward within timesteps after option $o$ is terminated and a new option is selected according to $\mu$. It can be shown that there exists an optimal (and deterministic) hierarchical policy $\mu^* = \{\mu_h^* : S \mapsto \mathcal{O}\}_{h \in [H]}$ that attains the optimal value function, i.e., $V^*_h(s) = \sup_\mu V^\mu_h(s)$ for all $(h, s) \in [H] \times S$ (Sutton et al., 1999).

### 3.2 Offline RL with Options

We consider learning with options in the offline setting. That is, given a dataset $\mathcal{D}$ that is collected by an experimenter through interacting with the environment, the algorithm outputs a hierarchical policy $\tilde{\mu}$. The sample complexity is measured by the suboptimality, i.e., the shortfall in the value function of the hierarchical policy $\tilde{\mu}$ compared to that of the optimal hierarchical policy $\mu^*$, which is given by

$$
\text{SubOpt}_\mathcal{D}(\tilde{\mu}, s_1) := V^*_1(s_1) - V^*_1(\tilde{\mu}(s_1))
$$

To derive a novel information-theoretic lower bound of $\text{SubOpt}_\mathcal{D}$, we first define some useful notations. For any hierarchical policy $\mu$, we denote by $\theta^\mu = \{\theta^\mu_h : S \times O \mapsto [0, 1]\}_{h \in [H]}$ its state-option

\(^3\)Note that any $H$-length episodic MDP with a stochastic initial state is equivalent to an $(H + 1)$-length MDP with a fixed initial state $s_0$.

\(^4\)The formal definitions can be found in Appendix A.
occupancy measure. That is, \( \theta_h^\mu(s, o) \) is the probability that the agent selects a particular option \( o \) at state \( s \) at timestep \( h \) (either when \( h = 1 \) or when the option \( o_{h-1} \) used at the timestep \( h - 1 \) is terminated) when following the hierarchical policy \( \mu \). With a slight abuse of the notation, we denote by \( \theta_h^\mu(s) := \sum_{o \in O} \theta_h^\mu(s, o) \) the state occupancy measure for any \( (h, s) \in [H] \times \mathcal{S} \). Further, we define

\[
Z_O^\mu := \sum_{h,s} \theta_h^\mu(s), \quad Z_O^{\mu'} := \sum_{h,s,o} \mathbb{I}[\theta_h^\mu(s, o) > 0]
\]

where \( \mathbb{I}[\cdot] \) is the indicator function. Intuitively, \( Z_O^\mu \) is the expected number of timesteps to alternate a new option and \( Z_O^{\mu'} \) is the maximal number of state-option pairs that can be visited, when following the hierarchical policy \( \mu \). The following proposition shows that options facilitate temporal abstraction and reduction of the state space.

**Proposition 1.** For any hierarchical policy \( \mu \), we have that \( Z_O^\mu \leq H \). If \( \mu \) is deterministic, i.e., \( \mu = \{\mu_h : \mathcal{S} \mapsto O\}_{h \in [H]} \), we further have that \( Z_O^{\mu'} \leq HS \). All the above equalities hold when \( O = \mathcal{A} \).

Next, we derive a novel information-theoretic lower bound of \( \text{SubOpt}_D \). The detailed proof can be found in Appendix C.

**Theorem 2** (Information-theoretic lower bound). Let \( \rho = \{\rho_h : \mathcal{S} \mapsto \Delta(O)\}_{h \in [H]} \) denote any hierarchical behavior policy to collect the dataset. Define the class of problem instances

\[ \mathcal{M}(C_{\text{option}}, z^*, \bar{z}^*) := \left\{ (M, \rho) : \text{Exists deterministic } \mu^* \text{ of an episodic MDP } M \right\} \]

such that \( \max_{h,s,o} \frac{\theta_h^{\mu^*}(s, o)}{\theta_h^\mu(s, o)} \leq C_{\text{option}}, Z_O^{\mu^*} \leq z^*, Z_O^{\mu^*} \leq \bar{z}^* \).

Suppose that \( C_{\text{option}} \geq 2, z^* \geq 1, \) and \( \bar{z}^* \geq \lfloor z^* \rfloor S \), where \( \lfloor x \rfloor := \max\{n \in \mathbb{N} : n \leq x\} \) is the largest integer no greater than \( x \in \mathbb{R} \). Then, there exists an absolute constant \( c_0 \) such that for any offline algorithm that outputs a hierarchical policy \( \tilde{\mu} \), if the number of episodes

\[
K \leq \frac{c_0 \cdot C_{\text{option}} H z^* \bar{z}^*}{\epsilon^2}
\]

then there exists a problem instance \( (M, \rho) \in \mathcal{M}(C_{\text{option}}, z^*, \bar{z}^*) \) on which the hierarchical policy \( \tilde{\mu} \) suffers from \( \epsilon \)-suboptimality, that is,

\[
\mathbb{E}_M[\text{SubOpt}_D(\tilde{\mu}, s)] \geq \epsilon
\]

where the expectation \( \mathbb{E}_M \) is with respect to the randomness during the execution of \( \rho \) within MDP \( M \).

Theorem 2 shows that, when dataset \( D \) sufficiently covers the trajectories induced by \( \mu^* \), i.e., \( \max_{h,s,o} \theta_h^{\mu^*}(s, o)/\theta_h^\mu(s, o) \leq C_{\text{option}} \), at least \( \Omega(C_{\text{option}} H Z_O^{\mu^*} \bar{z}^*/\epsilon^2) \) episodes are required to learn an \( \epsilon \)-optimal hierarchical policy from dataset \( D \). Note that when \( O = \mathcal{A} \), it recovers the lower bound \( \Omega(H^2SC^*/\epsilon^2) \) for offline RL with primitive actions, where \( C^* \) is the concentrability defined therein.

### 4 The PEVIO Algorithm

Inspired by the Pessimistic Value Iteration (PEVI) algorithm [Jin et al. 2021], we propose the PEssimistic Value Iteration for Learning with Options (PEVIO) in Algorithm 1. Given a dataset \( D \) and the corresponding Offline Option Evaluation (OOE) subroutine, whose details are specified in Sections 5.1 and 5.2, PEVIO outputs a hierarchical policy \( \tilde{\mu} = \{\tilde{\mu}_h : \mathcal{S} \mapsto \Delta(O)\}_{h \in [H]} \).

To estimate the Q-function, given a dataset \( D \), PEVIO first constructs \( (\hat{T}_h, \hat{U}_h, \Gamma) \) by the OOE subroutine (line 3). Specifically, \( \hat{T}_h \) and \( \hat{U}_h \) are the empirical counterparts of \( T_h \) and \( U_h \) presented in the Q-function given by Equation (1), respectively. In addition, \( \Gamma \) is a penalty function computed based on dataset \( D \). We remark that the OOE subroutine varies when different data-collecting procedures
To analyze the suboptimality of the hierarchical policy with Algorithm 1 PEVI and VI-LCB, we use the following definition, which motivates the design of the penalty function Π:

Definition 1 (ξ-uncertainty quantifier for dataset D). The penalty function  \( \Gamma_h : S \times O \mapsto \mathbb{R}^+ \) for any \( h \in [H] \) satisfies that \( \mathbb{P}_D(\mathcal{E}) \geq 1 - \xi \), where \( \mathbb{P}_D \) is the joint distribution of the data collecting process.

In other words, the penalty function \( \Gamma \) is a ξ-uncertainty quantifier if it upper bounds the estimation errors in the empirical option transition function \( \hat{T} \) and the empirical option utility function \( \hat{U} \). Next, we show that the suboptimality of the \( \hat{\mu} \) output from PEVIO is upper bounded if \( \Gamma \) is a ξ-uncertainty quantifier. (The detailed proof can be found in Appendix D.)

Theorem 3 (Suboptimality of learning with options using dataset D). Let \( \hat{\mu} \) denote the hierarchical policy output by Algorithm 1. Suppose that \( \Gamma = \{\Gamma_h\}_{h \in [H]} \) from the OOE subroutine is a ξ-uncertainty quantifier. Conditioned on the successful event \( \mathcal{E} \) defined in Equation (5), which satisfies that \( \mathbb{P}_D(\mathcal{E}) \geq 1 - \xi \), we have that

\[
\text{SubOpt}_D(\hat{\mu}, s_1) \leq 2 \sum_{h=1}^{H} \mathbb{E}_{\mu^*}[\Gamma_h(s_h, o_h)|s_1]
\]

where \( \mathbb{E}_{\mu^*}[g(s_h, o_h)] = \sum_{(s, o)} \theta_h^{\mu^*}(s, o)g(s, o) \) for any \( h \in [H] \) and arbitrary function \( g : S \times O \mapsto \mathbb{R} \).

Remark 1. Since the temporal structure of learning with options is much more complex than learning with actions, PEVIO is significantly different from the algorithms proposed for offline RL with primitive actions, such as PEVI [Lim et al., 2021] or VI-LCB [Xie et al., 2021], despite sharing a similar intuition. First, in terms of the algorithm design, PEVIO and VI-LCB estimate (one-step) transition kernel and reward function to compute the Q-function of a state-action pair. However, by Equation (1), the Q-function of a state-option pair depends on multi-step transitions and rewards.
We consider dataset $D$ (Single hierarchical policy concentrability for dataset which is collected by the experimenter’s interaction with the environment for $K$ episodes). Theorem 4 (Suboptimality for dataset $D$) is finite. Under Assumption 1, with probability at least $\frac{1}{2}$, we have that

\[
\text{SubOpt}_{D_1}(\hat{\mu}, s_1) \leq \hat{O}\left(\sqrt{\frac{C_1 \Gamma H^3}{K \xi}}\right)
\]
where $Z_0^\tau := Z_0^\tau$ and $Z_0^* := Z_0^*$. Compared to the lower bound in Theorem 2, suboptimality bound (8) is near-optimal except for an extra factor of $H^\tau$. More importantly, it shows that learning with options enjoys a faster convergence rate to the optimal value than learning with primitive actions. Recall that the VI-LCB algorithm (Xie et al. 2021) that learns with primitive actions attains the suboptimality bound $\tilde{O}(\sqrt{H^\tau SC^*/K})$, where $C^*$ is the concentrability defined therein. When ignoring the concentrability parameters, the suboptimality bound (8) is smaller since $Z_0^\tau \leq H$ and $Z_0^* \leq HS$.

**Remark 2.** While, in the worst case, both $Z_0^\tau$ and $Z_0^*$ can scale with $H$ and $HS$, respectively, we note that in many long-horizon planning problems, they often scale with the number of sub-tasks, which are greatly smaller, especially for tasks that enables temporal abstraction and the reduction of the state space. For example, while the route-planning task of going from City A to City B by transportation takes thousands of primitive actions to finish, it can be efficiently solved by decomposing into the following sub-tasks: (1) going to the airport/train station in City A; (2) taking transportation to City B; and (3) reaching the final destination in City B, for which options are designed. In this case, both $Z_0^\tau$ and $Z_0^*/S$ may only scale as $o(H)$. In other words, options facilitate more sample-efficient learning through temporal abstraction, i.e., sticking to an option until a sub-task is finished. Another concrete example is solving a maze, where options are often designed to move agents to bottleneck states (Simšek and Barto 2004; Solway et al. 2014; Machado et al. 2017) that connect different densely connected regions of the state space, e.g., doorways. In this case, while the number of option switches may grow proportionally to $H$, i.e., $Z_0^*/H = \tilde{O}(1)$, the number of states to switch options can be greatly smaller than $S$, i.e., $Z_0^*/H = o(s)$. That is to say, options help improve the sample complexity by the reduction of the state space.

Further, we show that learning with options attains a better performance than learning with primitive actions, when either the options are carefully designed or the offline data is limited.

**Corollary 1 (Better performance).** Let $\text{TrueSubOpt}_{\mathcal{D}_1}(\tilde{m}, s_1) := V_1^{*\text{-pri}}(s_1) - V_1^{\tau}(s_1)$, where $V_1^{*\text{-pri}}$ is the optimal value function defined for the primitive actions. Ignoring the concentrability parameters, we have that $\text{TrueSubOpt}_{\mathcal{D}_1}(\tilde{m}, s_1) \leq \tilde{O}(\sqrt{H^\tau SC^*/K})$ attained by the VI-LCB algorithm (Xie et al. 2021), when either the options are carefully designed (i.e., $\Delta_O(s_1) := V_1^{*\text{-pri}}(s_1) - V_1^{*}(s_1) = 0$) or

---

---

---
the number $K$ of trajectories in the dataset is
\[ o \left( \frac{H^3}{\Delta_T} \left( \sqrt{H^2SCt} - \sqrt{C_{opt}^s\sigma^s} \right)^2 \right) \]
where $(x)_+ := \max\{0, x\}$ for any $x \in \mathbb{R}$.

Corollary 1 implies that when data is limited, e.g., in cases where the data collection is highly expensive or risky, learning with options is beneficial since the output hierarchical policy yields a higher value than learning with primitive actions.

### 5.2 Learning from State-Action Transitions

We consider dataset $D_2 := \{(s_h^k, a_h^k, r_h^k)\}_{h \in [H], k \in [K]}$ consisting of state-action-reward tuples, which is collected by an experimenter’s interaction with the environment for $K$ episodes using any arbitrary behavior policy. That is, the experimenter takes action $a_h^k$ at state $s_h^k$ at timestep $h$ of the $k$th episode, receives a reward of $r_h^k$, and transits to state $s_{h+1}^k$.

When dataset $D_2$ is provided, the OOE subroutine in Algorithm 1 is given by Subroutine 3. Note that one difficulty is that we cannot directly estimate the option transition function and the option utility function from dataset $D_2$ as it only includes the information of the primitive actions. Hence, Subroutine 3 first constructs the empirical transition kernel $\hat{P} = \{\hat{P}_h\}_{h \in [H]}$ and the empirical reward function $\hat{r} = \{\hat{r}_h\}_{h \in [H]}$ (lines 4-7), and use them to further construct $\hat{T}_h$ and $\hat{U}_h$ (lines 8-20). To

---

<table>
<thead>
<tr>
<th>Subroutine 3 Offline Option Evaluation (OOE) for Dataset $D_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: <strong>Input:</strong> Dataset $D_2 = {(s_h^k, a_h^k, r_h^k)}_{h \in [H], k \in [K]}$.</td>
</tr>
<tr>
<td>2: <strong>Initialize:</strong> $\hat{U}<em>{t+1}(:) \leftarrow 0$. Let $N_h(s, a) := \sum</em>{k=1}^{K} \mathbb{I}[s_h^k = s, a_h^k = a]$ denote the number of visits of state-action pair $(s, a)$ at timestep $h$ in dataset $D_2$. Function $\phi := {\phi_h : \mathcal{S} \times \mathcal{O} \mapsto \mathbb{R}}_{h \in [H]}$ is given by Equation (32) in the Appendices.</td>
</tr>
<tr>
<td>3: <strong>for</strong> $h = H, H - 1, \ldots, 1$ <strong>do</strong></td>
</tr>
<tr>
<td>4: <strong>for</strong> $(s, a) \in \mathcal{S} \times \mathcal{A}$ <strong>do</strong></td>
</tr>
<tr>
<td>5: $\hat{P}_h(s'</td>
</tr>
<tr>
<td>6: $\hat{r}_h(s, a) \leftarrow \mathbb{I}[N_h(s, a) \geq 1]r_h(s, a)$</td>
</tr>
<tr>
<td>7: <strong>end for</strong></td>
</tr>
<tr>
<td>8: <strong>for</strong> $(s, a, s') \in \mathcal{S} \times \mathcal{O} \times \mathcal{S}$ <strong>do</strong></td>
</tr>
<tr>
<td>9: $\hat{T}_h(s'</td>
</tr>
<tr>
<td>10: $\hat{U}_h(s'</td>
</tr>
<tr>
<td>11: <strong>end for</strong></td>
</tr>
<tr>
<td>12: <strong>end for</strong></td>
</tr>
<tr>
<td>13: <strong>end for</strong></td>
</tr>
<tr>
<td>14: <strong>end for</strong></td>
</tr>
<tr>
<td>15: <strong>end for</strong></td>
</tr>
<tr>
<td>16: <strong>end for</strong></td>
</tr>
<tr>
<td>17: <strong>for</strong> $(h, s, o) \in [H] \times \mathcal{S} \times \mathcal{O}$ <strong>do</strong></td>
</tr>
<tr>
<td>18: $\Gamma_h(s, o) \leftarrow \hat{O} \left( \sqrt{\sum_{m=h}^{H} \mathbb{E}<em>{(s,a) \sim \mathcal{X}</em>{h,m}^{o}} \mathcal{X}<em>{m}(s,a) Y</em>{m}(s,a)Y_{m}(s,a)_{+}} + H\phi_h(s, o) \right)$</td>
</tr>
<tr>
<td>19: $\hat{U}<em>h(s, o) \leftarrow \sum</em>{a \in \mathcal{A}} \pi_{h}^{a}(a</td>
</tr>
<tr>
<td>20: <strong>end for</strong></td>
</tr>
<tr>
<td>21: <strong>Output:</strong> $(\hat{T} = {\hat{T}<em>h}</em>{h \in [H]}, \hat{U} = {\hat{U}<em>h}</em>{h \in [H]}, \Gamma = {\Gamma_h}_{h \in [H]}).$</td>
</tr>
</tbody>
</table>

---

For convenience, we assume that $\mathcal{X}_{h, s, o}$ is known prior. When $\mathcal{X}_{h, s, o}$ is unknown, it can be replaced by a superset that does not require prior knowledge and our results directly follow (See Remark 3 in Appendix D).
We analyze the pros and cons of both data-collection procedures, which sheds light on offline RL

While, in general, suboptimality bound (10) does not compare favorably against the suboptimality

With options in practice. Compared to SubOpt_{D_2} under Assumption 2. The detailed proof can be found in Appendix F.

Theorem 5 (Suboptimality for dataset D_2). Under Assumption 2 with probability at least 1 - \xi, we have that

SubOpt_{D_2}(\hat{\mu}, s_1) \leq \hat{O} \left( \sqrt{\frac{C_2^{option} H^2 S Z^*_O \sum h \in \mathcal{H}}{K}} + \frac{H^2 S C_2^{option}}{K} \right)

which translates to

\hat{O} \left( \sqrt{\frac{C_2^{option} H^2 S Z^*_O \sum h \in \mathcal{H}}{K}} \right)

when K of dataset D_2 is sufficiently large, i.e., K \geq \hat{O}(C_2^{option} H^5 S A^2 O^2 / (Z^*_O Z^*_O)), where

Z^*_O = Z^*_O \text{ and } Z^*_O = Z^*_O \text{.}

While, in general, suboptimality bound (10) does not compare favorably against the suboptimality
\hat{O}(\sqrt{H^2 S C^*_O / K}) attained by the VI-LCB algorithm that learns with primitive actions, we argue that it can be better in long-horizon problems where the horizon H is much greater than the cardinality of the state space S.

5.3 Further Discussion

We analyze the pros and cons of both data-collection procedures, which sheds light on offline RL

with options in practice. Compared to D_2, dataset D_1 requires smaller storage and enjoys faster

corvergence to the optimal value, which is further illustrated as follows.

- Storage: For dataset D_2 = \{(s^k_h, a^k_h, r^k_h)\}_{h \in \mathcal{H}, k \in [K]}, the storage is simply H K. However, for dataset D_1 = \{(s^k_i, a^k_i, w^k_i)\}_{i \in [S], k \in [K]}, its expected size is K \cdot Z^*_O \leq H K, where \rho is the hierarchical behavior policy. Therefore, in the case of a small Z^*_O, dataset D_1 requires much smaller storage than D_2 (on average).

- Suboptimality: Ignoring the concentrability, the suboptimality bound (10) for dataset D_2 is worse than the suboptimality bound (8) for dataset D_1 by a factor of \sqrt{S}, which is introduced when estimating the option transition function and the option utility function using only the information of the primitive actions.

However, since dataset D_2 contains more information on the environment than D_1, it has a weaker

requirement on the behavior (hierarchical) policy and allows the evaluation of new options, which is

illustrated as follows.

- Concentrability: Recall that the suboptimality bounds for both datasets build upon the suffi-
cient coverage assumptions, i.e., Assumption 1 for D_1 and Assumption 2 for D_2. While they

are generally incomparable (as dataset D_2 can be collected by an arbitrary behavior policy), we focus on the case that both datasets are collected by the same hierarchical behavior policy \rho. Particularly, it can be shown that Assumption 2 is weaker than Assumption 1. Indeed, if \rho
covers the trajectories of state-option-utility tuples induced by $\mu^*$ (i.e., Assumption 1 holds), then it must have covered the trajectories of state-action-reward tuples induced by $\mu^*$ (i.e., Assumption 2 holds). However, the opposite does not hold in general and we provide such an example in Appendix H.

- Evaluation of New Options: In the options literature, a popular task is offline option discovery [Ajay et al., 2021; Villecroze et al., 2022], i.e., designing new and useful options from the dataset. Therefore, an important problem is whether these new options can be evaluated through the dataset. We argue that dataset $D_2$ yields greater flexibility than $D_1$ in this case. Again, we assume that both datasets are collected by the same hierarchical behavior policy $\rho$. Unfortunately, one cannot use dataset $D_1$ to evaluate any $(h, s, o)$ that is not visited by $\rho$, i.e., $\theta^\rho_h(s, o) = 0$, let alone evaluating the new options. However, this is not the case for dataset $D_2$. In fact, any $(h, s, o)$ can be evaluated if the visiting state-action pairs are also reachable by $\rho$, i.e., $\sum_{h \leq m \leq H, (s', a') \in S_{h, s, o}} 1/d^\rho_h(s', a') < \infty$. An interesting problem is how to leverage the results in this paper to facilitate offline option discovery, which we shall research in the future work.

6 Conclusions

In this paper, we provide the first analysis of the sample complexity for offline RL with options. A novel information-theoretic lower bound is established, which generalizes the one for offline RL with actions. We derive near-optimal suboptimality bounds of the PESimistic VAlue Iteration for Learning with Options (PEVIO) algorithm for two popular data-collection procedures. Our results show that options facilitate more sample-efficient learning than primitive actions in offline RL in both the finite-time convergence rate to the optimal value and the actual performance.

Acknowledgments and Disclosure of Funding

The work presented in this paper was partially supported by a research grant from the Research Grants Council, Hong Kong, China (RGC Ref. No. CUHK 14206820). Part of the research reported in this paper was done when the second author was with The Chinese University of Hong Kong. The authors are grateful to the anonymous reviewers for their helpful comments.

References


