Risk-Sensitive Reinforcement Learning with Function Approximation: A Debiasing Approach

Yingjie Fei [1] **Zhuoran Yang** [2] **Zhaoran Wang** [1]

Abstract

We study function approximation for episodic reinforcement learning with entropic risk measure. We first propose an algorithm with linear function approximation. Compared to existing algorithms, which suffer from improper regularization and regression biases, this algorithm features debiasing transformations in backward induction and regression procedures. We further propose an algorithm with general function approximation, which is shown to perform implicit debiasing transformations. We prove that both algorithms achieve a sublinear regret and demonstrate a tradeoff between generality and efficiency. Our analysis provides a unified framework for function approximation in risk-sensitive reinforcement learning, which leads to the first sub-linear regret bounds in the setting.

1. Introduction

In this paper, we consider the problem of risk-sensitive reinforcement learning (RL) with the entropic risk measure, a classical framework pioneered by the seminal work of Howard & Matheson (1972). Informally, for a given risk parameter $\beta \neq 0$, our goal is to learn a policy that maximizes the following objective of a total reward R,

$$V_\beta = \frac{1}{\beta} \log\{\mathbb{E}e^{\beta R}\}. \tag{1}$$

Here, the expectation is taken over the policy, transitions and possible randomness of the reward R; a formal definition is given in (2) below. The objective (1) admits the Taylor expansion $V_\beta = \mathbb{E}[R] + \frac{\beta}{2}\mathrm{Var}(R) + O(\beta^2)$. Comparing it with the risk-neutral objective $V = \mathbb{E}[R]$ studied in the standard RL setting, we see that $\beta > 0$ induces a risk-seeking objective and $\beta < 0$ induces a risk-averse one. It can also be seen that V_β tends to the risk-neutral V as $\beta \to 0$. Risk-sensitive RL has been widely applied in behavioral modeling in neuroscience and psychology (Braun et al., 2011; Nagengast et al., 2010; Niv et al., 2012; Shen et al., 2014). Therefore, better designs and analyses of risk-sensitive RL algorithms may contribute to a better understanding of human behaviors, which further helps improve human-oriented systems such as recommendation systems, online marketplaces, human-computer interfaces and etc.

Unfortunately, existing risk-sensitive RL algorithms suffer serious drawbacks: they either are designed for the tabular setting only, which do not scale to the large state-action space of the real world, or lack finite-sample guarantees, which makes it difficult to understand and interpret the behaviors of the algorithms in a principled way. We remedy this unsatisfactory situation by studying algorithm design with function approximation and regret guarantees. Function approximation enables algorithms to work efficiently in large or even infinite state spaces, and regret quantifies the performance of algorithms in terms of key model parameters. This is a challenging task, however, for reasons we outline in the following.

Challenges. Most existing works on function approximation for RL with regret guarantees focus on the risk-neutral setting and follow the paradigm of value-targeted regression. A key step of such approach is to estimate models by least-squares regression. However, this line of works heavily exploits the linearity of risk-neutral objective $V = \mathbb{E}[R]$ in both transition dynamics (implicitly captured by the expectation) and the reward R, which is not available in the risk-sensitive objective (1). It is therefore unclear whether this approach might apply to risk-sensitive RL given its nonlinear objective. Specifically, for linear function approximation, it is unclear what features, targets and regularization should be used in the regression procedure, as well as how they would lead to provable regret guarantees. Given these complications in the linear setting, a principled algorithm design for general function approximation appears even more elusive.

Our contributions. In this work, we address the above

[1]Northwestern University, Evanston, Illinois, USA [2]Princeton University, Princeton, New Jersey, USA. Correspondence to: Yingjie Fei <yf275@cornell.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

challenges and make the following contributions.

1. For linear function approximation, we introduce a novel algorithm named RSVI-L. While existing algorithms suffer from improper regularization and biases in regression procedures, RSVI-L resolves these issues with three of its mechanisms: 1) it regularizes the regression procedure in a way that adapts to the full range of risk sensitivity; 2) it applies simple yet crucial debiasing transformations to both regression features and targets; 3) it applies another debiasing transformation to backward induction in response to the debiasing of 2). Under episodic Markov decision processes (MDPs), we show that RSVI-L achieves sub-linear regret with respect to the number of episodes. We provide insights on how the regularization and debiasing together lead to the regret guarantee, which demonstrates a synergistic relation between the two.

2. We also consider the setting of general function approximation and provide a novel algorithm named RSVI-G, which is substantially different from RSVI-L. Perhaps interestingly, we show that RSVI-G performs implicit debiasing transformations to regression features and targets. In addition, we prove that it also admits sub-linear regret in terms of the number of episodes. A comparison of RSVI-L and RSVI-G shows that RSVI-L is computationally more efficient in the linear setting, while RSVI-G applies to more general settings of function approximation.

3. Our proof establishes a unified framework for analyzing both linear and general function approximation. The framework is flexible enough to incorporate other types of function approximation in future studies and could be of independent interest.

To the best of our knowledge, this is the first work that provides risk-sensitive RL algorithms with function approximation that attain sub-linear regret.

Related work. Initiated by the seminal work of Howard & Matheson (1972), risk-sensitive control/RL with the entropic risk measure has been studied in a vast body of literature (Bäuerle & Rieder, 2014; Borkar, 2001; 2002; 2010; Borkar & Meyn, 2002; Cavazos-Cadena & Hernández-Hernández, 2011; Coraluppi & Marcus, 1999; Di Masi & Stettner, 1999; 2000; 2007; Fleming & McEneaney, 1995; Hernández-Hernández & Marcus, 1996; Jaśkiewicz, 2007; Marcus et al., 1997; Mihatsch & Neuneier, 2002; Osogami, 2012; Patek, 2001; Shen et al., 2013; 2014; Whittle, 1990). Yet, this line of works either assumes known transition kernels or focuses on asymptotic behaviors of the problem/algorithms.

The most relevant work to ours is perhaps Fei et al. (2020), who consider the same problem as ours under the tabular setting. They propose two algorithms based on value iteration and Q-learning. They prove regret bounds for their algorithms, which are then certified to be nearly optimal by a lower bound. However, their algorithms and analysis are restricted to the tabular setting. Compared to Fei et al. (2020), our paper proposes novel algorithms with linear and general function approximation, which scale to large or even infinite state spaces. It is worth noting that both of the function approximation settings subsume the tabular setting.

We also briefly discuss existing works on function approximation with regret analysis, which so far have focused on the risk-neutral setting. The works of Cai et al. (2019); Jin et al. (2019); Wang et al. (2019); Yang & Wang (2019); Zhou et al. (2020) study linear function approximation, while Ayoub et al. (2020); Wang et al. (2020) investigate general function approximation. In addition, they provide sub-linear regret bounds for their algorithms. In contrast with the risk-neutral RL problem, the nonlinear objective (1) makes algorithm design and regret analysis for function approximation much more challenging in risk-sensitive settings.

Notations. For a positive integer n, we let $[n] := \{1, 2, \ldots, n\}$. For a number $u \neq 0$, we define $\text{sign}(u) = 1$ if $u > 0$ and -1 if $u < 0$. For two non-negative sequences $\{a_i\}$ and $\{b_i\}$, we write $a_i \lesssim b_i$ if there exists a universal constant $C > 0$ such that $a_i \leq C b_i$ for all i, and write $a_i \asymp b_i$ if $a_i \lesssim b_i$ and $b_i \lesssim a_i$. We use $\tilde{O}(\cdot)$ to denote $O(\cdot)$ while hiding logarithmic factors. For any $\varepsilon > 0$ and set $\mathcal{X}$, we let $\mathcal{N}_\varepsilon(\mathcal{X}, \|\cdot\|)$ be the ε-net of the set $\mathcal{X}$ with respect to the norm $\|\cdot\|$. We let $\Delta(\mathcal{X})$ be the set of probability distributions supported on $\mathcal{X}$. For any vector $u \in \mathbb{R}^n$ and symmetric and positive definite matrix $\Gamma \in \mathbb{R}^{n \times n}$, we let $\|u\|_\Gamma := \sqrt{u^\top \Gamma u}$. We denote by I_n the $n \times n$ identity matrix.

2. Problem formulation

2.1. Episodic MDP

An episodic MDP is parameterized by a tuple $(K, H, \mathcal{S}, \mathcal{A}, \{P_h\}_{h\in[H]}, \{r_h\}_{h\in[H]})$, where K is the number of episodes, H is the number of steps in each episode, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel at step h, and $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function at step h. We assume that the transitions $\{P_h\}$ are unknown. For simplicity we assume that the reward functions $\{r_h\}$ are known and deterministic, as is common in existing works (Ayoub et al., 2020; Yang & Wang, 2019; Zhou et al., 2020).

We interact with the episodic MDP as follows. In the beginning of each episode $k \in [K]$, the environment chooses an arbitrary initial state $s_1^k \in \mathcal{S}$. Then in each step $h \in [H]$, we take an action $a_h^k \in \mathcal{A}$, receive a reward $r_h(s_h^k, a_h^k)$ and transition to the next state $s_{h+1}^k \in \mathcal{S}$ sampled from

$P_h(\cdot \mid s_h^k, a_h^k)$. Once we reach s_{H+1}^k, the current episode terminates and we advance to the next episode unless $k = K$.

2.2. Value functions, Bellman equations and regret

We assume that β is fixed prior to the learning process, and for notational simplicity we omit it from quantities to be introduced subsequently. In risk-sensitive RL with the entropic risk measure, we aim to find a policy $\pi = \{\pi_h : \mathcal{S} \to \mathcal{A}\}$ so as to maximize the *value function* given by

$$V_h^\pi(s) := \frac{1}{\beta} \log\left\{\mathbb{E}\left[e^{\beta \sum_{h'=h}^H r_{h'}(s_{h'}, \pi_{h'}(s_{h'}))} \,\middle|\, s_h = s\right]\right\}, \tag{2}$$

for all $(h, s) \in [H] \times \mathcal{S}$. In the above the expectation is taken over π_h and P_h. Under some mild regularity conditions, there exists a greedy policy $\pi^* = \{\pi_h^*\}$ which gives the optimal value $V_h^{\pi^*}(s) = \sup_\pi V_h^\pi(s)$ for all $(h, s) \in [H] \times \mathcal{S}$ (Bäuerle & Rieder, 2014). In addition to the value function, another key notion is the *action-value function* defined as

$$Q_h^\pi(s, a) := \frac{1}{\beta} \log\left\{\mathbb{E}\left[e^{\beta \sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'})} \,\middle|\, \begin{smallmatrix} s_h = s \\ a_h = a \end{smallmatrix}\right]\right\}, \tag{3}$$

for all $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$.

The action-value function Q_h^π is associated with the value function V_h^π via the so-called *Bellman equation*:

$$\begin{aligned} Q_h^\pi(s, a) &= r_h(s, a) + \frac{1}{\beta} \log\{\mathbb{E}_{s'}[e^{\beta \cdot V_{h+1}^\pi(s')}]\}, \qquad (4) \\ V_h^\pi(s) &= Q_h^\pi(s, \pi_h(s)), \qquad V_{H+1}^\pi(s) = 0, \end{aligned}$$

which holds for all $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$. Here, the expectation $\mathbb{E}_{s'}$ is taken over $P_h(\cdot|s, a)$. Similarly, the *Bellman optimality equation* is given by

$$\begin{aligned} Q_h^*(s, a) &= r_h(s, a) + \frac{1}{\beta} \log\{\mathbb{E}_{s'}[e^{\beta \cdot V_{h+1}^*(s')}]\}, \qquad (5) \\ V_h^*(s) &= \max_{a \in \mathcal{A}} Q_h^*(s, a), \qquad V_{H+1}^*(s) = 0, \end{aligned}$$

again for all $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$. In the above, we use the shorthand $Q_h^* := Q_h^{\pi^*}$ for all $h \in [H]$ and V_h^* is similarly defined. The identity $V_h^*(\cdot) = \max_{a \in \mathcal{A}} Q_h^*(\cdot, a)$ implies that the optimal π^* is the greedy policy with respect to the optimal action-value function $\{Q_h^*\}_{h \in [H]}$.

During the learning process, the policy π^k in each episode k may be different from the optimal π^*. We quantify this difference over all K episodes through the notion of *regret,* defined as

$$\text{Regret}(K) := \sum_{k \in [K]} \left[V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k)\right]. \tag{6}$$

Since $V_1^*(s) \geq V_1^\pi(s)$ for any π and $s \in \mathcal{S}$, the regret characterizes the sub-optimality of $\{\pi^k\}$ relative to the optimal π^*.

2.3. Function approximation

In this paper, we focus on linear and general function approximation. We consider the following form of linear function approximation, where each transition kernel admits a linear form.

Assumption 1. *We assume that the MDP is equipped with a known feature function $\psi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ such that for any $h \in [H]$, there exists a vector $\theta_h \in \mathbb{R}^d$ with $\|\theta_h\|_2 \leq \sqrt{d}$ and the transition kernel is given by*

$$P_h(s' \mid s, a) = \psi(s, a, s')^\top \theta_h$$

for any $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. We also assume that

$$\left\| \int_{\mathcal{S}} \psi(s, a, s') f(s') \mathrm{d}s' \right\|_2 \leq \sqrt{d} \sup_{s' \in \mathcal{S}} |f(s')|,$$

for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ and function $f : \mathcal{S} \to \mathbb{R}$.

This form of linear function approximation is also studied in Ayoub et al. (2020); Cai et al. (2019); Zhou et al. (2020), whose setting is equivalent to ours when $\beta \to 0$. The setting of Assumption 1 may be reduced to the tabular setting in which $\psi(s, a, s')$ is a canonical basis vector in $\mathbb{R}^d$ with $d = |\mathcal{S}|^2 |\mathcal{A}|$, i.e., the (s, a, s')-th entry of $\psi(s, a, s')$ is equal to one and the other entries are equal to zero. It also subsumes various settings of function approximation including linear combinations of base models (Modi et al., 2020) and the matrix bandit setting (Yang & Wang, 2019); we refer readers to (Zhou et al., 2020) for more details on the generality of Assumption 1.

Sometimes, the underlying model is so rich and complicated that the assumption of linear kernels may be too restrictive. We therefore also consider general function approximation, for which we make the following general assumption.

Assumption 2. *We assume that we have access to a function set[1] $\mathcal{P}$ such that the transition kernel $P_h \in \mathcal{P}$ for all $h \in [H]$.*

This setting is also considered in Ayoub et al. (2020). It is not hard to see that Assumption 2 subsumes Assumption 1. Under Assumption 2, we may measure the complexity of function sets using the notion of the eluder dimension. To introduce the eluder dimension, we need to set forth the concept of ε-independence.

Definition 1. *For any $\varepsilon > 0$ and function set $\mathcal{G}$ whose elements are in the domain $\mathcal{X}$, we say that an $x \in \mathcal{X}$ is ε-dependent on the set of elements $\mathcal{X}_n := \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$ with respect to $\mathcal{G}$, if any pair of functions $g, g' \in \mathcal{G}$ satisfying $\sum_{i \in [n]} (g(x_i) - g'(x_i))^2 \leq \varepsilon^2$ also satisfies $g(x) - g'(x) \leq \varepsilon$. We say that x is ε-independent of $\mathcal{X}_n$*

[1] Throughout the paper, we use *function class* and *function set* interchangeably.

with respect to $\mathcal{G}$ if x is not ε-dependent on $\mathcal{X}_n$ with respect to $\mathcal{G}$.

Hence, ε-independence characterizes a notion of dissimilarity of a point x to the elements in subset $\mathcal{X}_n$ of function set $\mathcal{G}$. Now we are ready to formally define the eluder dimension, which quantifies the length of the longest possible chain of dissimilar elements in a function set.

Definition 2. *For any $\varepsilon > 0$ and function set $\mathcal{G}$ whose elements are in the domain $\mathcal{X}$, the ε-eluder dimension $\dim_E(\mathcal{G}, \varepsilon)$ is defined as the length d' of the longest sequence of elements in $\mathcal{X}$ such that, for some $\varepsilon' \geq \varepsilon$, every element is ε'-independent of its predecessors.*

The eluder dimension extends the concept of dimension in linear spaces and generalizes to non-linear function spaces. It is also related to the notions of Kolmogorov and VC dimensions. We refer readers to Russo & Van Roy (2014) for further details on the eluder dimension and its advantages compared to other complexity measures.

Although we focus on function approximation of transition kernels, a similar approach can be taken to apply function approximation to reward functions and our regret guarantees presented below remain valid, as argued in Yang & Wang (2019).

3. Algorithms

To streamline the presentation of our algorithms, we first introduce a meta algorithm in Algorithm 1, namely **Meta** **R**isk-**S**ensitive **V**alue **I**teration (MetaRSVI), which is a high-level framework including key features of value iteration algorithms (Bradtke & Barto, 1996; Jin et al., 2019). It consists of a value estimation step (Lines 3–6) and policy execution step (Lines 8–11). In the value estimation step, the algorithm estimates the optimal Q_h^* by its iterates Q_h^k based on historical data. We focus on greedy policies, and in Line 5 the estimated value function $V_h^k(\cdot)$ is taken as the maximum among $\{Q_h^k(\cdot, a')\}_{a' \in \mathcal{A}}$. The key machinery of value estimation, known as **R**isk-**S**ensitive **T**emporal **D**ifference or RSTD, is abstracted out in Line 4; we will provide concrete forms of this function for both linear and general function approximation in the sections to follow. In the policy execution step, the algorithm uses the policy learned in the current episode (represented by Q_h^k) to collect data for subsequent update procedures.

Algorithm 1 MetaRSVI

Input: risk parameter β, number of episodes K
1: **for** episode $k = 1, \ldots, K$ **do**
2: $V_{H+1}^k(\cdot) \leftarrow 0$
3: **for** step $h = H, H-1, \ldots, 1$ **do**
4: $Q_h^k(\cdot, \cdot) \leftarrow \mathrm{RSTD}(k, h, \beta, \{V_{h+1}^\tau\}_{\tau \in [k]})$
5: $V_h^k(\cdot) \leftarrow \max_{a' \in \mathcal{A}} Q_h^k(\cdot, a')$
6: **end for**
7: Receive initial state s_1^k from environment
8: **for** step $h = 1, 2, \ldots, H$ **do**
9: Take action $a_h^k \leftarrow \mathrm{argmax}_{a' \in \mathcal{A}} Q_h^k(s_h^k, a')$
10: Receive next state s_{h+1}^k
11: **end for**
12: **end for**

3.1. RSVI-L

We introduce **R**isk-**S**ensitive **V**alue **I**teration with **L**inear function approximation, or RSVI-L, in Algorithm 2. This algorithm is inspired by RSVI proposed in Fei et al. (2020) under the tabular setting. In Line 6 the iterate w_h^k can be interpreted as the solution of the following least-squares problem:

$$w_h^k \leftarrow \operatorname*{argmin}_{w \in \mathbb{R}^d} \ \lambda \|w\|_2^2 + \sum_{\tau \in [k-1]} [(e^{\beta \cdot V_{h+1}^\tau(s_{h+1}^\tau)} - 1) - w^\top \phi_h^\tau(s_h^\tau, a_h^\tau)]^2, \tag{7}$$

$$= (\Lambda_h^k)^{-1} \sum_{\tau \in [k-1]} \phi_h^\tau(s_h^\tau, a_h^\tau)(e^{\beta \cdot V_{h+1}^\tau(s_{h+1}^\tau)} - 1)$$

where the regression features $\{\phi_h^\tau\}_{\tau=1}^{k-1}$ are constructed in Line 4 and $\lambda = (e^{\beta H} - 1)^2$ is a regularization parameter. The above regression procedure computes an estimate of θ_h, the parameter of the unknown transition kernel P_h. Note that the regression targets are proportional to a *shifted* exponential V-estimates $\{e^{\beta \cdot V_{h+1}^\tau} - 1\}$; similar construction is applied to the regression feature ϕ_h^k in Line 4. This is a distinctive property of Algorithm 2, which we will compare and contrast with RSVI soon. To update Q_h^k in Line 7, we perform backward induction

$$Q_h^k(\cdot, \cdot) \leftarrow r_h(\cdot, \cdot) + \frac{1}{\beta} \log(q_{h,L}^k(\cdot, \cdot)), \tag{8}$$

where

$$q_{h,L}^k(\cdot, \cdot) := \begin{cases} \min\{e^{\beta(H-h)}, \langle \phi_h^k(\cdot, \cdot), w_h^k \rangle + b_h^k(\cdot, \cdot) + 1\}, & \beta > 0, \\ \max\{e^{\beta(H-h)}, \langle \phi_h^k(\cdot, \cdot), w_h^k \rangle - b_h^k(\cdot, \cdot) + 1\}, & \beta < 0. \end{cases} \tag{9}$$

Here, $q_{h,L}^k$ can be seen as an optimistic estimate for the expected value of $e^{\beta \cdot V_{h+1}^k}$ under the transition kernel P_h. The quantity $\pm b_h^k$ takes the role of bonus to enable efficient exploration for $\beta > 0$ and $\beta < 0$, respectively. Therefore, Algorithm 2 follows the principle of Risk-Sensitive Optimism in the Face of Uncertainty (RS-OFU) postulated in Fei

et al. (2020). Finally, the thresholding step in (9) ensures the estimate Q_h^k to be on the same scale as the optimal action values Q_h^* entrywise.

Comparison with existing algorithms. We highlight three major features of Algorithm 2 that differentiate it from RSVI of (Fei et al., 2020) designed for the tabular setting as well as risk-neutral algorithms with linear function approximation proposed in (Jin et al., 2019; Yang & Wang, 2019; Cai et al., 2019).

First, Algorithm 2 applies carefully designed regularization λ in the regression procedure (7), in contrast with existing algorithms whose regularization is inappropriate for our setting. One purpose of regularization λ is to keep the covariance matrix Λ_h^k from being singular. Another important role of λ is to regulate the error of $q_{h,L}^k$ in estimating the expectation of $e^{\beta \cdot V_{h+1}^k}$ with respect to the true model (where $q_{h,L}^k$ is defined in (9)). The scale of the estimation error depends on β and we therefore require λ to adapt to the full range of risk sensitivity (both $\beta > 0$ and $\beta < 0$). As we will explain later in Section 4.1, the choice of $\lambda = (e^{\beta H} - 1)^2$ manages to serve all of these purposes at once. This is in sharp contrast with $\lambda = 0$ used in RSVI and the common choice of $\lambda = 1$ in risk-neutral algorithms: setting $\lambda = 0$ as in RSVI would cause the covariance Λ_h^k to be singular (and therefore destabilizing the algorithm), whereas the fixed regularization $\lambda = 1$ in risk-neutral algorithms fails to adapt to the estimation error of $q_{h,L}^k$ that varies in β.

Second, Algorithm 2 has a distinct design of regression features and targets, which can be seen as a result of debiasing the regression step in RSVI. In particular, the regression features of Algorithm 2 take the form of

$$\phi_h^\tau(\cdot,\cdot) = \int_{\mathcal{S}} \psi(\cdot,\cdot,s')(e^{\beta \cdot V_{h+1}^\tau(s')} - 1)\mathrm{d}s'$$

and satisfy $\|\phi_h^\tau(\cdot,\cdot)\|_2 \approx |e^{\beta H} - 1|\sqrt{d}$, whereas those of RSVI are given by

$$\tilde{\phi}_h^\tau(\cdot,\cdot) = \int_{\mathcal{S}} \psi(\cdot,\cdot,s')e^{\beta \cdot V_{h+1}^\tau(s')}\mathrm{d}s'$$

whose norm is approximately $e^{\beta H}\sqrt{d}$. To ensure stability and efficiency of Algorithm 2, we would like Λ_h^k to be well-behaved, in the sense that the spectrums of $\phi_h^\tau(\cdot,\cdot)\phi_h^\tau(\cdot,\cdot)^\top$ and λI_d are close to each other; otherwise, a dominating $\phi_h^\tau(\cdot,\cdot)\phi_h^\tau(\cdot,\cdot)^\top$ would lead to a near-singular Λ_h^k, while a dominating λ would prevent the algorithm from efficient learning. This means that, for all fixed β, we want to design regression features to minimize the quantity

$$|\mathrm{Tr}(\phi_h^\tau(\cdot,\cdot)\phi_h^\tau(\cdot,\cdot)^\top - \lambda I_d)| = |\|\phi_h^\tau(\cdot,\cdot)\|^2 - \lambda d|,$$

which can be interpreted as the bias of features $\{\phi_h^\tau\}$ with respect to λ. Given $\lambda = (e^{\beta H} - 1)^2$, the bias of features $\{\tilde{\phi}_h^\tau\}$ used in RSVI would be excessive, especially when $|\beta|$ is small ($\|\tilde{\phi}_h^\tau(\cdot,\cdot)\| \to \sqrt{d}$ compared to $\lambda \to 0$, as $|\beta| \to 0$), which may cause Λ_h^k to be near-singular. On the other hand, the new feature design ϕ_h^τ in Algorithm 2 alleviates such bias by ensuring that $\|\phi_h^\tau(\cdot,\cdot)\|^2$ and λd are compatible. Along with the choice of $e^{\beta \cdot V_{h+1}^\tau} - 1$ as the regression targets, we may think of the regression features and targets in Algorithm 2 as debiasing those in RSVI. The relationship between regression features/targets and regularization λ demonstrates a synergistic interplay between the two in Algorithm 2.

Algorithm 2 RSVI-L

Input: risk parameter β, number of episodes K, regularization λ, bonus multiplier γ_L

1: Run Algorithm 1 with RSTD therein overloaded by the following subroutine:
2: **procedure** RSTD(k, h, β, $\{V_{h+1}^\tau\}_{\tau\in[k]}$, γ_L, λ)
3: $\quad\Lambda_h^k \leftarrow \sum_{\tau\in[k-1]} \phi_h^\tau(s_h^\tau, a_h^\tau)\phi_h^\tau(s_h^\tau, a_h^\tau)^\top + \lambda I_d$
4: $\quad\phi_h^k(\cdot,\cdot) \leftarrow \int_{\mathcal{S}} \psi(\cdot,\cdot,s')(e^{\beta \cdot V_{h+1}^k(s')} - 1)\mathrm{d}s'$
5: $\quad b_h^k(\cdot,\cdot) \leftarrow \gamma_L[\phi_h^k(\cdot,\cdot)^\top(\Lambda_h^k)^{-1}\phi_h^k(\cdot,\cdot)]^{1/2}$
6: $\quad$Compute w_h^k as in (7)
7: $\quad$**return** $Q_h^k(\cdot,\cdot)$ as computed in (8)
8: **end procedure**

Third, Algorithm 2 applies debiasing to backward induction, whereas RSVI does not. Specifically, Algorithm 2 uses $\langle \phi_h^k(\cdot,\cdot), w_h^k\rangle + 1$ in its backward induction step (8), in contrast with $\langle \phi_h^k(\cdot,\cdot), w_h^k\rangle$ used in RSVI. The new backward induction step in Algorithm 2 can be considered as performing a debiasing transformation on that of RSVI. This is because, by the definition of ϕ_h^k, the quantity $\langle \phi_h^k(\cdot,\cdot), w_h^k\rangle$ estimates $\mathbb{E}_{s'}[e^{\beta \cdot V_{h+1}^k(s')}] - 1$, rather than $\mathbb{E}_{s'}[e^{\beta \cdot V_{h+1}^k(s')}]$ as suggested by the Bellman equation (4). Therefore, the debiasing transformation, i.e., adding 1 to $\langle \phi_h^k(\cdot,\cdot), w_h^k\rangle$, helps correct the bias of -1 and aligns the backward induction step with the Bellman equation. We remark that the debiasing of backward induction is unique to Algorithm 2, due to its novel design of regression features.

3.2. RSVI-G

Oftentimes, the linear transitions postulated in Assumption 1 may not be sufficient for modeling complicated dynamics in the real world. We therefore need to consider more general settings such as that of Assumption 2. To that end, we present **R**isk-**S**ensitive **V**alue **I**teration with **G**eneral function approximation (RSVI-G) in Algorithm 3. A key quantity for Algorithm 3 is the squared error

$$\Gamma_h^\tau(P, P') \coloneqq \left[\int_{\mathcal{S}} P(s' \mid s_h^\tau, a_h^\tau)e^{\beta \cdot V_{h+1}^\tau(s')}\mathrm{d}s'\right.$$

$$- \int_{\mathcal{S}} P'(s' \mid s_h^\tau, a_h^\tau) e^{\beta \cdot V_{h+1}^\tau(s')} \mathrm{d}s' \Bigg]^2, \tag{10}$$

which can be thought of as the difference in expected $e^{\beta \cdot V_{h+1}^\tau}$ under two models P, P'. In Algorithm 3, Line 3 computes an estimate P_h^k of the true model P_h by solving a least-squares problem over the class of transition kernels $\mathcal{P}$. Specifically, we define $\widehat{P}_h^\tau(\cdot \mid s, a)$ to be the delta function centered at s_{h+1}^τ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. and we compute

$$P_h^k \leftarrow \operatorname*{argmin}_{P \in \mathcal{P}} \sum_{\tau=1}^{k-1} \Gamma_h^\tau(P, \widehat{P}_h^\tau). \tag{11}$$

Given the quantity γ_G to be determined later, Line 4 then constructs a confidence ball $\mathcal{P}_h^k$ of radius γ_G around the estimate P_h^k by

$$\mathcal{P}_h^k \leftarrow \left\{ P \in \mathcal{P} : \sum_{\tau=1}^{k-1} \Gamma_h^\tau(P, P_h^k) \le \gamma_G^2 \right\}. \tag{12}$$

We then update Q_h^k by

$$Q_h^k(\cdot, \cdot) \leftarrow r_h(\cdot, \cdot) + \frac{1}{\beta} \log(q_{h,G}^k(\cdot, \cdot)), \tag{13}$$

where

$$q_{h,G}^k(\cdot, \cdot) \coloneqq \begin{cases} \max_{P \in \mathcal{P}_h^k} \int_{\mathcal{S}} P(s' \mid \cdot, \cdot) e^{\beta \cdot V_{h+1}^k(s')} \mathrm{d}s', & \text{if } \beta > 0, \\ \min_{P \in \mathcal{P}_h^k} \int_{\mathcal{S}} P(s' \mid \cdot, \cdot) e^{\beta \cdot V_{h+1}^k(s')} \mathrm{d}s', & \text{if } \beta < 0. \end{cases} \tag{14}$$

The maximization and minimization in (14) serve to maintain optimism for $\beta > 0$ and $\beta < 0$ respectively, following the principle of RS-OFU. It is worth noting that both Lines 3 and 4 implicitly operate with the shifted exponential V-functions: indeed, replacing $e^{\beta \cdot V_{h+1}^\tau}$ therein by $e^{\beta \cdot V_{h+1}^\tau} - 1$ would not make any difference for the algorithm. Therefore, this can be seen as *implicitly* debiasing the regression features and targets in the tabular algorithm RSVI, in contrast with the explicit debiasing transformations of Algorithm 2.

Comparing Algorithms 2 and 3. We remark that Algorithms 2 and 3 are very different in nature. Algorithm 2 enforces optimism by adding bonus that comes with closed-form expression, while Algorithm 3 constructs confidence sets by solving quadratic programs and maintains optimism by solving linear programs. It can be seen that Algorithm 2 has polynomial time and space complexities in d, K and H. For Algorithm 3, it is unclear how the complexities scale under Assumption 2, where the structure of $\mathcal{P}$ is unknown. Nevertheless, under Assumption 1 in which the transition kernels admit a linear form, Algorithm 3 also attains polynomial time and space complexities. Although Algorithm 2 requires explicit debiasing, it enjoys faster runtime speed and less memory consumption than Algorithm 3, since it does not construct confidence sets by solving optimization problems that may be computationally expensive, as done in Algorithm 3. On the other hand, Algorithm 3 does not require transition dynamics to be linear and therefore applies to more general settings. This represents a tradeoff in efficiency and generality between Algorithms 2 and 3.

Algorithm 3 RSVI-G

Input: risk parameter β, number of episodes K, confidence width γ_G, function set $\mathcal{P}$

1: Run Algorithm 1 with RSTD therein overloaded by the following subroutine:
2: **procedure** RSTD(k, h, β, $\{V_{h+1}^\tau\}_{\tau \in [k]}$, γ_G, $\mathcal{P}$)
3: Compute P_h^k as in (11)
4: Compute $\mathcal{P}_h^k$ as in (12)
5: **return** $Q_h^k(\cdot, \cdot)$ as computed in (13)
6: **end procedure**

4. Main results

4.1. Regret bound for Algorithm 2

In this section, we present a regret bound for Algorithm 2. Let us define the bonus multiplier in Algorithm 2 as

$$\gamma_L \coloneqq c_\gamma |e^{\beta H} - 1| \sqrt{d \log(2dKH/\delta)}, \tag{15}$$

where $c_\gamma > 0$ is an appropriate universal constant. We have the following result.

Theorem 1. *Let $\lambda = (e^{\beta H} - 1)^2$ and γ_L of (15) be input to Algorithm 2. Under Assumption 1, for any $\delta \in (0, 1]$, with probability at least $1 - \delta$, the regret of Algorithm 2 satisfies*

$$\mathit{Regret}(K) \lesssim \frac{e^{|\beta| H} - 1}{|\beta|} e^{|\beta| H^2} \sqrt{d^2 K H^2 \log^2(2dKH/\delta)}.$$

The proof is given in Appendix B. One may obtain a regret bound for the tabular setting by taking $d = |\mathcal{S}|^2 |\mathcal{A}|$ in Theorem 1, and the resulting bound matches that of Fei et al. (2020, Theorem 1) up to polynomial factors of $|\mathcal{S}|$ and $|\mathcal{A}|$.[2] The bound is also nearly optimal for small $|\beta|$ (with respect to $|\beta|$, K and H) in view of the lower bound by Fei et al. (2020, Theorem 3),

$$\mathbb{E}\left[\mathrm{Regret}(K)\right] \gtrsim \frac{e^{|\beta| H/2} - 1}{|\beta|} \sqrt{K}. \tag{16}$$

[2] By inspecting the proof of Fei et al. (2020, Theorem 1), we see that they apply the bound $(1/|\beta|)(\exp(|\beta|\, H) - 1) \exp(|\beta|\, H^2) \le (1/|\beta|)(\exp(C\, |\beta|\, H^2) - 1)$ for some universal constant $C > 0$.

In addition, as $\beta \to 0$, the setting of risk-sensitive RL tends to that of standard risk-neutral RL. We have the following corollary to Theorem 1 for that regime.

Corollary 1. *Under the setting of Theorem 1 and when $\beta \to 0$, with probability at least $1-\delta$, the regret of Algorithm 2 satisfies*

$$\textit{Regret}(K) \lesssim \sqrt{d^2 K H^4 \log^2(2dKH/\delta)}.$$

The proof is given in Appendix C. The result in Corollary 1 matches that of the risk-neutral setting, e.g. Cai et al. (2019, Theorem 3.1), up to logarithmic factors.

Roles of regularization and debiasing in analysis. Our analysis of Theorem 1 shows that the dominating factor of regret can be written in the form of $(B_1 + B_2)D$, where $B_1 \approx \sum_{k,h}(q^k_{h,L} - \mathbb{E}e^{\beta \cdot V^k_{h+1}})$ is the sum of KH random variables (given $q^k_{h,L}$ defined in (9)), B_2 is proportional to $\sqrt{\lambda}$, and D is the sum of norms of regression features. By a standard concentration inequality and the debiasing of the backward induction step in Algorithm 2, we deduce $B_1 \propto |e^{\beta H} - 1|$ and our choice of $\lambda = (e^{\beta H} - 1)^2$ puts B_2 on the same scale as B_1. This argument is made formal in Lemma 2. Moreover, we apply an elliptical potential lemma (Lemma 6) to show that $D \le \log[(\lambda + K\|\phi\|_2^2)/\lambda]$, where ϕ is a regression feature. Given λ in Theorem 1 and $\|\phi\| \propto (e^{\beta H} - 1)^2$ as a result of feature debiasing in regression (7), we conclude that $D \lesssim \log K$ (ignoring other model parameters in log). Putting together the above results yields the regret bound in Theorem 1. As a passing note, we remark that the regularization and debiasing are novel characteristics of Algorithm 2 compared to existing RL algorithms, and they play crucial roles in regret analysis.

4.2. Regret bound for Algorithm 3

To present the regret guarantee for Algorithm 3, we need to set a few additional notations. Recall the function set $\mathcal{P}$ from Assumption 2. For any $P \in \mathcal{P}$, $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $V : \mathcal{S} \to [0, H]$, we define the function set

$$\mathcal{Z} \coloneqq \{z_P : P \in \mathcal{P}\}, \tag{17}$$

where

$$z_P(s,a,V) \coloneqq \int_{\mathcal{S}} P(s' \mid s,a)|e^{\beta \cdot V(s')} - 1|\mathrm{d}s'. \tag{18}$$

For any $P, P' \in \mathcal{P}$, we define $\|P - P'\|_{\infty,1} \coloneqq \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \|P(\cdot \mid s,a) - P'(\cdot \mid s,a)\|_1$. To measure the complexity of the set $\mathcal{Z}$, we use the notion of eluder dimension (Definition 2) and we let

$$d_E \coloneqq \dim_E(\mathcal{Z}, |e^{\beta H} - 1|/K)$$

be the $(|e^{\beta H} - 1|/K)$-eluder dimension of function set $\mathcal{Z}$. In Algorithm 3, we set

$$\gamma_G \coloneqq 10|e^{\beta H} - 1|\sqrt{\zeta}, \tag{19}$$

where

$$\zeta \coloneqq \log\left(H \cdot \mathcal{N}_{1/K}(\mathcal{P}, \|\cdot\|_{\infty,1})/\delta\right) + \sqrt{\log(4K^2H/\delta)}. \tag{20}$$

We are now ready to state our result for Algorithm 3.

Theorem 2. *Let γ_G of (19) be input to Algorithm 3. Under Assumption 2, for any $\delta \in (0,1]$ and with probability at least $1-\delta$, the regret of Algorithm 3 satisfies*

$$\textit{Regret}(K) \lesssim \frac{e^{|\beta|H} - 1}{|\beta|} e^{|\beta|H^2} \cdot \left(H \min\{d_E, K\} + \sqrt{d_E K H^2 \zeta}\right).$$

The proof is given in Appendix D. When $K \gtrsim d_E$, we have $H\min\{d_E, K\} \lesssim \sqrt{d_E K H^2 \zeta}$ and therefore Theorem 2 yields

$$\mathrm{Regret}(K) = \frac{e^{|\beta|H} - 1}{|\beta|} e^{|\beta|H^2} \tilde{O}(\sqrt{d_E K H^2}).$$

Under Assumption 1, a special case of Assumption 2 where the transition kernels in $\mathcal{P}$ take the linear form, we have $d_E \lesssim d \log K$ and $\log(\mathcal{N}_{1/K}(\mathcal{P}, \|\cdot\|_{\infty,1})) \lesssim d \log K$, implying $\zeta \lesssim d \log(KH/\delta)$. Then for sufficiently large K, the regret bound of Theorem 2 matches that of Theorem 1 up to a logarithmic factor. Furthermore, in the tabular setting, we have $d_E = \tilde{O}(d) = \tilde{O}(|\mathcal{S}|^2|\mathcal{A}|)$ and hence Theorem 2 is also nearly optimal compared to the lower bound (16).

Technical highlights. The key to the proof of Theorem 2 is to identify the amount of optimism maintained by the confidence set (12). We show that this quantity is proportional to $|e^{\beta H} - 1|$, thanks to the implicit debiasing property of Algorithm 3. Despite the apparent differences between Algorithms 2 and 3, we establish an analytic framework that unifies the proofs of Theorems 1 and 2. Specifically, we identify an optimism condition, under which any instance of Algorithm 1 achieves a regret bound on the same order of its optimism. We then show that both Algorithms 2 and 3 satisfy the optimism condition and therefore attain the claimed regret bounds. More details on this framework will be provided in Section 5 to follow. The flexibility of the framework allows incorporating other types of function approximation for future research, which could be of independent interest.

To the best of our knowledge, Theorems 1 and 2 are the first sub-linear regret guarantees for risk-sensitive RL algorithms with function approximation.

5. A unified theoretical framework

We present a unified analytic framework based on Algorithm 1, and the results can be specialized to any of its instance including Algorithm 2 and 3. We first identify a risk-sensitive optimism condition, which certifies a certain form of optimism that adapts to risk sensitivity. Under this optimism condition, we prove a regret bound for Algorithm 1. Then in Appendix we instantiate the regret bound of Algorithm 1 for Algorithms 2 and 3 under Assumptions 1 and 2, respectively, by showing that both algorithms satisfy the aforementioned optimism condition.

5.1. Optimism condition

Recall the iterates $\{Q_h^k\}$ in Algorithm 1. For each $(k,h) \in [K] \times [H]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$, we define

$$\overline{Q}_h^k(s,a) \coloneqq r_h(s,a) + \frac{1}{\beta} \log \left\{ \mathbb{E}_{s' \sim P_h(\cdot \mid s,a)} \left[e^{\beta \cdot V_{h+1}^k(s')} \right] \right\}.$$

It can be seen that $\{\overline{Q}_h^k\}$ are the ideal counterparts of $\{Q_h^k\}$ that could be constructed if the transition kernels $\{P_h\}$ were known. We set forth a risk-sensitive optimism condition, which is a central component of our unified framework.

Condition 1. *For all $(k,h,s,a) \in [K] \times [H] \times \mathcal{S} \times \mathcal{A}$, we have $Q_h^k(s,a) \in [0, H-h+1]$, and there exist some quantities $m_h^k(s,a) > 0, g \geq 1$ and universal constant $c > 0$ such that*

$$0 \leq Q_h^k(s,a) - \overline{Q}_h^k(s,a) \leq c \cdot \frac{e^{|\beta|H} - 1}{|\beta|} \cdot g \cdot m_h^k(s,a).$$

Since Q_h^k is an optimistic estimate of the ideal quantity $\overline{Q}_h^k$, the difference $Q_h^k - \overline{Q}_h^k$ in Condition 1 may be thought of as the level of optimism maintained by the algorithm for state-action pair (s,a) in step h of episode k, with its upper bound depending on risk sensitivity through the factor $\frac{e^{|\beta|H}-1}{|\beta|}$. Therefore, we say that Condition 1 is a risk-sensitive optimism condition and it serves as a quantification of the RS-OFU principle. In the upper bound of the condition, the actual values of g and m_h^k may depend on function approximation and implementation of the abstract function RSTD. An advantage of considering Condition 1 is that it helps disentangle the complications of function approximation from the regret analysis, and it is flexible to incorporate other function approximation settings beyond Assumptions 1 and 2. In Appendices B and D, we show that Algorithms 2 and 3, respectively, satisfy this condition with high probability.

5.2. Unified regret bound

Let us recall that $\{(s_h^k, a_h^k)\}$ are the state-action pairs visited by Algorithm 1. Below, we state a regret bound for Algorithm 1 that unifies Theorems 1 and 2.

Theorem 3. *Define*

$$M \coloneqq g \sum_{k \in [K]} \sum_{h \in [H]} \min\{1, m_h^k(s_h^k, a_h^k)\}$$

where g and $\{m_h^k\}$ are as given in Condition 1. On the event of Condition 1, for any $\delta \in (0,1]$, with probability at least $1-\delta$ the regret of Algorithm 1 satisfies

$$\textit{Regret}(K) \lesssim \frac{e^{|\beta|H} - 1}{|\beta|} e^{|\beta|H^2} M + e^{|\beta|H^2} \sqrt{KH^3 \log(1/\delta)}.$$

The proof is given in Appendix F. Although the actual form of M depends on specific function approximation, the derivation of Theorem 3 only requires the structure of Algorithm 1, which is agnostic of function approximation. In the above bound, the first term can be interpreted as the total optimism maintained by Algorithm 1, and is in fact a direct consequence of Condition 1. The second term is the total drift of iterates $\{V_h^k\}$ from the value functions $\{V_h^{\pi^k}\}$, which is the result of a martingale analysis. The factor $e^{|\beta|H^2}$ shared by both terms is due to a local linearization of the nonlinear objective (2) as well as a standard backward induction analysis of H-horizon MDPs. When instantiating Theorem 3 under Assumptions 1 and 2, our proof shows that $M = \tilde{O}(\sqrt{K})$ so the first term would dominate in the regret bound. Similar to M, the exponential factor $\frac{e^{|\beta|H}-1}{|\beta|}$ also comes into the bound from Condition 1. It has been shown as a distinctive feature of risk-sensitive RL algorithms that represents a tradeoff between risk sensitivity and sample complexity (Fei et al., 2020).

6. Conclusion

This work investigates function approximation for risk-sensitive RL with the entropic risk measure. We propose two algorithms, RSVI-L and RSVI-G, under the settings of linear and general function approximation, respectively. We demonstrate that RSVI-L applies risk-adaptive regularization and debiasing transformations in its regression procedure, which correct the improper regularization and regression biases in existing algorithms. On the other hand, RSVI-G is shown to perform implicit debiasing, and applies to more general settings compared to RSVI-L. Through a unified analytic framework, we prove that both algorithms achieve sub-linear regret in terms of the number of episodes. This is the first work designing risk-sensitive RL algorithms with function approximation that achieve sub-linear regret.

Acknowledgements

We thank the reviewers for their constructive feedback. Z. Yang acknowledges Simons Institute (Theory of Reinforcement Learning). Z. Wang acknowledges National

Science Foundation (Awards 2048075, 2008827, 2015568, 1934931), Simons Institute (Theory of Reinforcement Learning), Amazon, J.P. Morgan, and Two Sigma for their supports.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24:2312–2320, 2011.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. F. Model-based reinforcement learning with value-targeted regression. *arXiv preprint arXiv:2006.01107*, 2020.

Bäuerle, N. and Rieder, U. More risk-sensitive Markov decision processes. *Mathematics of Operations Research*, 39(1):105–120, 2014.

Borkar, V. S. A sensitivity formula for risk-sensitive cost and the actor-critic algorithm. *Systems & Control Letters*, 44(5):339–346, 2001.

Borkar, V. S. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27(2):294–311, 2002.

Borkar, V. S. Learning algorithms for risk-sensitive control. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems–MTNS*, pp. 55–60, 2010.

Borkar, V. S. and Meyn, S. P. Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research*, 27(1):192–209, 2002.

Bradtke, S. J. and Barto, A. G. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996.

Braun, D. A., Nagengast, A. J., and Wolpert, D. Risk-sensitivity in sensorimotor control. *Frontiers in human neuroscience*, 5:1, 2011.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830*, 2019.

Cavazos-Cadena, R. and Hernández-Hernández, D. Discounted approximations for risk-sensitive average criteria in Markov decision chains with finite state space. *Mathematics of Operations Research*, 36(1):133–146, 2011.

Coraluppi, S. P. and Marcus, S. I. Risk-sensitive, minimax, and mixed risk-neutral/minimax control of Markov decision processes. In *Stochastic Analysis, Control, Optimization and Applications*, pp. 21–40. Springer, 1999.

Di Masi, G. B. and Stettner, L. Risk-sensitive control of discrete-time Markov processes with infinite horizon. *SIAM Journal on Control and Optimization*, 38(1):61–78, 1999.

Di Masi, G. B. and Stettner, L. Infinite horizon risk sensitive control of discrete time Markov processes with small risk. *Systems & Control Letters*, 40(1):15–20, 2000.

Di Masi, G. B. and Stettner, Ł. Infinite horizon risk sensitive control of discrete time Markov processes under minorization property. *SIAM Journal on Control and Optimization*, 46(1):231–252, 2007.

Fei, Y., Yang, Z., Chen, Y., Wang, Z., and Xie, Q. Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret. *arXiv preprint arXiv:2006.13827*, 2020.

Fleming, W. H. and McEneaney, W. M. Risk-sensitive control on an infinite time horizon. *SIAM Journal on Control and Optimization*, 33(6):1881–1915, 1995.

Hernández-Hernández, D. and Marcus, S. I. Risk sensitive control of Markov processes in countable state space. *Systems & Control Letters*, 29(3):147–155, 1996.

Howard, R. A. and Matheson, J. E. Risk-sensitive Markov decision processes. *Management Science*, 18(7):356–369, 1972.

Jaśkiewicz, A. Average optimality for risk-sensitive control with general state space. *The Annals of Applied Probability*, 17(2):654–675, 2007.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. *arXiv preprint arXiv:1907.05388*, 2019.

Marcus, S. I., Fernández-Gaucherand, E., Hernández-Hernandez, D., Coraluppi, S., and Fard, P. Risk sensitive Markov decision processes. In *Systems and Control in the Twenty-first Century*, pp. 263–279. Springer, 1997.

Mihatsch, O. and Neuneier, R. Risk-sensitive reinforcement learning. *Machine Learning*, 49(2-3):267–290, 2002.

Modi, A., Jiang, N., Tewari, A., and Singh, S. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pp. 2010–2020, 2020.

Nagengast, A. J., Braun, D. A., and Wolpert, D. M. Risk-sensitive optimal feedback control accounts for sensorimotor behavior under uncertainty. *PLoS Comput Biol*, 6 (7):e1000857, 2010.

Niv, Y., Edlund, J. A., Dayan, P., and O'Doherty, J. P. Neural prediction errors reveal a risk-sensitive reinforcement-learning process in the human brain. *Journal of Neuroscience*, 32(2):551–562, 2012.

Osogami, T. Robustness and risk-sensitivity in Markov decision processes. In *Advances in Neural Information Processing Systems*, pp. 233–241, 2012.

Patek, S. D. On terminating Markov decision processes with a risk-averse objective function. *Automatica*, 37(9): 1379–1386, 2001.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39 (4):1221–1243, 2014.

Shen, Y., Stannat, W., and Obermayer, K. Risk-sensitive Markov control processes. *SIAM Journal on Control and Optimization*, 51(5):3652–3672, 2013.

Shen, Y., Tobia, M. J., Sommer, T., and Obermayer, K. Risk-sensitive reinforcement learning. *Neural Computation*, 26(7):1298–1328, 2014.

Wang, R., Salakhutdinov, R., and Yang, L. F. Provably efficient reinforcement learning with general value function approximation. *arXiv preprint arXiv:2005.10804*, 2020.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Whittle, P. *Risk-sensitive Optimal Control*, volume 20. Wiley New York, 1990.

Yang, L. F. and Wang, M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019.

Zhou, D., He, J., and Gu, Q. Provably efficient reinforcement learning for discounted mdps with feature mapping. *arXiv preprint arXiv:2006.13165*, 2020.