Randomized Value Functions via Posterior State-Abstraction Sampling

Dilip Arumugam*[1] and Benjamin Van Roy†[1,2]

[1]Stanford University
[2]DeepMind

Abstract

State abstraction has been an essential tool for dramatically improving the sample efficiency of reinforcement-learning algorithms. Indeed, by exposing and accentuating various types of latent structure within the environment, different classes of state abstraction have enabled improved theoretical guarantees and empirical performance. When dealing with state abstractions that capture structure in the value function, however, a standard assumption is that the true abstraction has been supplied or unrealistically computed a priori, leaving open the question of how to efficiently uncover such latent structure while jointly seeking out optimal behavior. Taking inspiration from the bandit literature, we propose that an agent seeking out latent task structure must explicitly represent and maintain its uncertainty over that structure as part of its overall uncertainty about the environment. We introduce a practical algorithm for doing this using two posterior distributions over state abstractions and abstract-state values. In empirically validating our approach, we find that substantial performance gains lie in the multi-task setting where tasks share a common, low-dimensional representation.

1 Introduction

The complexity of the state space within an environment has profound impact on a decision-making agent's capacity for sample-efficient reinforcement learning [Kearns and Singh, 2002, Kakade, 2003, Strehl et al., 2009, Auer et al., 2009]. Oftentimes, however, this complexity (as measured by the cardinality of a finite set of states or the dimensionality of real-valued observations/features) is an exaggeration of the amount of information actually needed to make optimal (or even near-optimal) decisions. And yet, many reinforcement-learning algorithms make no concerted effort to fully exploit this structure so as to accelerate learning of the optimal policy or value function [Watkins and Dayan, 1992, Williams, 1992, Sutton, 1988, Sutton et al., 2000, Mnih et al., 2015].

In contrast, there exist several works which closely examine state abstractions from a theoretical or empirical perspective, highlighting the advantages of acknowledging and utilizing latent problem structure [Bertsekas et al., 1988, Dean and Givan, 1997, Ferns et al., 2004, Jong and Stone, 2005, Li et al., 2006, Van Roy, 2006, Ferns et al., 2012, Jiang et al., 2015, Abel et al., 2016, 2019, Dong et al., 2019, Du et al., 2019, Misra et al., 2019, Russo, 2020]. While important for gaining clarity and an appreciation for what state abstraction brings to the reinforcement-learning problem [Lake et al., 2017, Konidaris, 2019], some of these works can be unrealistic in their acquisition of the very object being studied. Specifically, those approaches focusing on abstractions that capture structure within the optimal value function often assume that an ideal state abstraction has been provided by an oracle [Li et al., 2006, Abel et al., 2016, Dong et al., 2019]; while this

*dilip@cs.stanford.edu
†bvr@stanford.edu

Preprint. Work in progress.

may be a suitable provision for the purpose of theoretical analysis, this engenders a two-step procedure in practice whereby the optimal value function is computed exactly and then used to compute the corresponding state abstraction [Abel et al., 2016, 2018]. These approaches beg a natural question: *can an agent learn a value-based state abstraction while simultaneously learning how to act optimally within an environment?* Incorporating this additional abstraction criterion, beyond the standard RL objective of learning an optimal policy, amounts to inserting a form of inductive bias within the learning algorithm. Thus, a natural follow-up question arises: *when is it advantageous to learn a state abstraction alongside learning optimal behavior?*

In this work, we offer an affirmative answer to the first question and provide an empirically-supported hypothesis to the second. In addressing the former, we build upon an existing line of work that grounds (both theoretically and empirically) the efficient learning of optimal behavior through an agent's representation of epistemic uncertainty over its environment [Russo and Van Roy, 2016, Osband et al., 2016a, Russo and Van Roy, 2018, O'Donoghue et al., 2018, Osband et al., 2019]. Informally, it is the agent's resolution of this uncertainty that naturally fosters deep exploration and provably-efficient learning. Adopting this perspective, we assert that an agent must maintain an explicit belief over the state abstraction that underlies the environment. This belief can then be coupled with a conditional posterior over abstract-state values to render the agent's full uncertainty over the optimal value function. As for when it is advantageous to incorporate this factored posterior, we empirically demonstrate considerable performance gains when an agent engages with multiple tasks in the same environment, all supported by a single, underlying state abstraction.

The paper proceeds as follows: in Section 2, we formulate the problem and clarify our setting where the optimal value function admits a natural state abstraction, which we formalize as exhibiting low-rank structure. We then discuss related work and background in Sections 3 and 4, respectively, before introducing our approach, Posterior State-abstraction Sampling (PS2), in Section 5. We conclude with illustrative experiments in contextual-bandit problems, exploring both the single-task and multi-task learning settings (Sections 6 and 7).

2 Problem Formulation

2.1 Reinforcement Learning

We formulate an individual task as a finite-horizon, episodic Markov Decision Process (MDP) [Bellman, 1957, Puterman, 1994] defined by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma, H \rangle$ where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the finite action set, $\mathcal{R}_h : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is a (deterministic) reward function for timestep h, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ is the transition function producing a distribution over next states given the current state-action pair, $\gamma \in [0, 1)$ is the discount factor, and $H \in \mathbb{N}$ is the horizon or episode duration. We use $|\mathcal{A}| = A$ and, when $\mathcal{S}$ is finite, $|\mathcal{S}| = S$ to denote the respective sizes of the action space and state space.

Learning proceeds in T stages or episodes where, at each timestep of the current episode $h \in [H]$, the agent observes the current state s_h and selects an action a_h according to its current policy $\pi_h : \mathcal{S} \mapsto \Delta(\mathcal{A})$. We assume that $\mathcal{M}$ has a fixed initial state distribution $\rho \in \Delta(\mathcal{S})$ such that $s_1 \sim \rho(\cdot)$ is observed at the start of each episode. The objective for the agent is to synthesize a policy so as to maximize expected return $\mathbb{E}[\sum\limits_{h=1}^{H} \gamma^{h-1} \mathcal{R}_h(s_h, a_h)]$. The value function of a policy π denotes the expected future discounted return by following the policy from a given state s:

$$V_h^\pi(s_h) = \mathbb{E}[\sum_{h'=0}^{H-h} \gamma^{h'} \mathcal{R}_h(s_{h'}, a_{h'}) | s_{h'} = s_h]$$

where the expectation is taken with respect to the stochasticity in the environment transition dynamics and policy. Similarly, we use the action-value function $Q^\pi(s, a)$ to define the expected future discounted return from being in a state s, taking action a, and following policy π thereafter:

$$Q_h^\pi(s, a) = \mathcal{R}_h(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)}[V_{h+1}^\pi(s')]$$

Denoting the policy class containing all stationary, stochastic policies as $\Pi = \{\pi | \pi : \mathcal{S} \mapsto \Delta(\mathcal{A})\}$, we may define the optimal policy $\pi^\star = \sup\limits_{\pi \in \Pi^H} V_1^\pi(s)$, where Π^H denotes the class of non-stationary policies, whose value functions are given by the Bellman optimality equations:

$$V_h^\star(s) = \max_{a \in \mathcal{A}} Q_h^\star(s,a)$$
$$Q_h^\star(s,a) = \mathcal{R}_h(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)}[\max_{a' \in \mathcal{A}} Q_{h+1}^\star(s',a')]$$

where $V_{H+1}^\star(s) = 0$. Concretely, our goal is to leverage experience sampled in each episode to learn $Q^\star$, yielding the optimal policy $\pi_h^\star(s) = \arg\max\limits_{a \in \mathcal{A}} Q_h^\star(s,a)$ [Sutton and Barto, 1998, Watkins and Dayan, 1992].

2.2 State Abstractions

A well-studied tool for accelerating RL algorithms is the use of state aggregation or state abstraction to reduce the size of the MDP state space [Bertsekas et al., 1988, Li et al., 2006, Van Roy, 2006]. Indeed, given the dependence on $|\mathcal{S}|$ that appears in numerous sample complexity results for RL [Kakade, 2003, Strehl et al., 2009], it seems only natural that a reduction in the overall number of states under consideration can dramatically alleviate the burdens of learning an optimal policy.

As outlined in Li et al. [2006], several classes of state abstractions exist, each elucidating structure contained in different components of the MDP. A choice of state abstraction from one of these classes characterizes a particular function $\phi : \mathcal{S} \mapsto \mathcal{S}_\phi$ mapping original or ground states of the MDP into an aggregate or abstract state space $\mathcal{S}_\phi$. Naturally, the abstract state space is taken to be smaller, in some sense, than the original (for instance, $|\mathcal{S}_\phi| < |\mathcal{S}|$) such that ϕ defines a (lossy) compression of the original state space [Abel et al., 2019]. While state abstraction constitutes a general mechanism for specifying latent task structure, several works often make generous assumptions concerning the provision or acquisition of ϕ in order to highlight the benefits of leveraging such structure in RL. In contrast, this work is concerned with weakening those assumptions specifically by learning ϕ concurrently with $Q^\star$.

Various prior works have focused on the capacity of state abstraction to enable provably-efficient RL algorithms that scale to tasks with high-dimensional observations [Du et al., 2019, Misra et al., 2019, Agarwal et al., 2020]. These approaches build state abstractions based on latent structure in the transition dynamics; by operating in the reward-free setting [Hazan et al., 2019, Jin et al., 2020], which lacks a specific reward function, these algorithms employ various mechanisms to facilitate thorough exploration of the environment, yielding a strong approximation of the transition model. This style of approach seems like a natural "path of least resistance" in so far as each step within the environment yields a ground-truth signal that can be aimed at further distilling the true, underlying abstraction. In contrast, a state abstraction based on the similarity of $Q^\star$-values [Li et al., 2006, Abel et al., 2016, 2019, Dong et al., 2019] inherently draws upon knowledge of $Q^\star$ which, if readily accessible to the agent, would imply knowledge of the optimal policy itself. Nevertheless, in this paper, we explicitly direct our focus to these $Q^\star$-based state abstractions and avert the apparent "chicken-or-egg" problem through the agent's own epistemic uncertainty about the environment.

2.3 Low-Rank Value Functions

We formalize the type of $Q^\star$-based state abstraction studied in this work under the following definition of a low-rank value function:

Definition 1 (Low-Rank $Q^\star$-function): *The optimal action-value function $Q^\star$ of an MDP $\mathcal{M}$ is characterized as low-rank if there exists two functions, $\phi_h^\star : \mathcal{S} \mapsto \mathbb{R}^M$ and $\psi_h^\star : \mathcal{A} \mapsto \mathbb{R}^M$, such that $\forall s \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$:*

$$Q_h^\star(s,a) = \langle \phi_h(s), \psi_h(a) \rangle$$

where $\langle u, v\rangle$ denotes the inner product between vectors u, v. We interpret the latent dimension $M \in \mathbb{N}$ as the number of abstract states.

Assumption 1. *Throughout this work, we will assume that $\mathcal{M}$ admits a low-rank $Q^\star$-function with rank M.*

To clarify this structural assumption, consider the case where $\mathcal{M}$ has a finite state-action space. Dropping the timestep subscript for clarity, note that $Q^\star$ may then be compactly represented as a matrix $\mathbf{Q}^\star \in \mathbb{R}^{S\times A}$ where $\mathbf{Q}^\star_{ij} = Q^\star(s_i, a_j)$, for some arbitrary indexing of states and actions. Defining the rank of a matrix as

Definition 2 (Matrix Rank): *For an arbitrary matrix $Y \in \mathbb{R}^{n\times m}$, the rank of Y is defined as*

$$\textit{rank}(Y) = \min\{k | k \in \mathbb{N}_+, \exists U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{m\times k} \textit{ such that } Y = UV^T\}$$

Assumption 1 then follows as $\text{rank}(\mathbf{Q}^\star) = M$. In words, we assume that the matrix of optimal action-values admits a low-rank factorization $\mathbf{Q}^\star = \mathbf{\Phi}\mathbf{\Psi}^T$, with $\mathbf{\Phi} \in \mathbb{R}^{S\times M}$, $\mathbf{\Psi} \in \mathbb{R}^{A\times M}$. Under this view, we can interpret learning a low-rank decomposition of $\mathbf{Q}^\star$ as constructing predictive features in a M-dimensional subspace that enable accurate estimation of $Q^\star$ for each state-action pair [Tsitsiklis and Van Roy, 1996].

We conclude this section by briefly making explicit some connections between low-rank value functions and the $Q^\star$-similarity state abstractions employed in prior work. Specifically, notice that the exact $Q^\star$-irrelevance abstraction of Li et al. [2006] corresponds to an abstraction function $\phi^\star : \mathcal{S} \mapsto \{0, 1\}^M$ where $\forall s \in \mathcal{S}, \sum\limits_{i=1}^{M} \phi(s)_i = 1$; in words, $\phi^\star$ maps each state to a one-hot vector thereby guaranteeing that $\phi(s_1) = \phi(s_2) \iff Q^\star(s_1, a) = Q^\star(s_2, a), \forall s_1, s_2 \in \mathcal{S}, a \in \mathcal{A}$. An approximate version of this abstraction, as studied in [Abel et al., 2016, Dong et al., 2019], can be achieved by maintaining the same conditions on $\phi^\star$ and weakening the definition of a low-rank value function to $|Q^\star(s, a) - \langle\phi^\star(s)\psi^\star(a)\rangle| \leq \frac{\epsilon}{2}$, for some constant ϵ. Finally, the soft state aggregations of Singh et al. [1995] impose the alternative condition $\phi^\star : \mathcal{S} \mapsto \Delta^{M-1}$, mapping individual states to the $(M - 1)$-dimensional probability simplex.

3 Related Work

As previously discussed, this paper falls in with a long, rich line of work on state abstraction in reinforcement learning [Whitt, 1978, Bertsekas et al., 1988, Dean and Givan, 1997, Ferns et al., 2004, Jong and Stone, 2005, Li et al., 2006, Van Roy, 2006, Ferns et al., 2012, Jiang et al., 2015, Abel et al., 2016, 2019, Dong et al., 2019, Du et al., 2019, Misra et al., 2019]. Notably, this work is concerned with how an agent may incrementally learn a state abstraction that capitalizes on latent structure in the optimal value function [Bertsekas et al., 1988, Van Roy, 2006, Li et al., 2006, Abel et al., 2016, 2018, 2019, Dong et al., 2019]. Bertsekas et al. [1988] focus on adaptively synthesizing state aggregations based on Bellman-error residuals. Van Roy [2006] examines and provides performance guarantees on approximate value iteration under the provision of a particular state aggregation, leaving open the question of how to dynamically abstract states and maintain performance guarantees. Li et al. [2006] offer a unified perspective on a broad array of state abstraction types, showcasing how state abstraction based on $Q^\star$-similarity preserves the optimal policy. Later, Abel et al. [2016] generalize this to the case of approximate state abstraction, highlighting the approximation parameter as a knob for weighing state-space compression against value loss. Abel et al. [2018] examine the lifelong learning setting where $Q^\star$ is computed exactly for some number of MDPs before a $Q^\star$-similarity state abstraction is then applied for the remainder of the task distribution. Abel et al. [2019] formalize this intuition using tools from rate-distortion theory [Shannon, 1959], but restrict focus to the apprenticeship learning setting. Common to all of these works is the lack of a practical, scalable algorithm for jointly learning the state abstraction and corresponding abstract-state values, without knowledge of $Q^\star$; our work rectifies this and offers one such approach.

Our work is also intimately related to the problem of low-rank matrix completion or factorization. While the general problem of low-rank matrix completion is underspecified and known to be NP-hard [Chistov and Grigor'Ev, 1984, Hardt et al., 2014], a large body of prior work identifies sufficient conditions for designing provably-efficient factorization algorithms [Candès and Recht, 2009, Candès and Tao, 2010, Candès and Plan, 2010, Keshavan et al., 2010a,b, Recht, 2011]. The algorithm presented in this work aligns with gradient-descent based approaches for iteratively optimizing the latent factors $\mathbf{\Phi}, \mathbf{\Psi}$ which, despite their weaker sample complexity guarantees [Jain et al., 2013, Gunasekar et al., 2013, Hardt, 2014, Chen and Wainwright, 2015], are simple, scalable, and widely deployed in practice [Recht and Ré, 2013, Koren et al., 2009]. Unlike the standard formulation of the low-rank matrix completion problem, our focus on the sequential decision-making setting more closely aligns with adaptive-sampling approaches to matrix completion [Krishnamurthy and Singh, 2013, 2014] which are known to enjoy better sample complexity guarantees.

Several papers adopt a matrix factorization perspective for state abstraction in reinforcement learning [Barreto et al., 2016, Behzadian and Petrik, 2018, Duan et al., 2019, Zhang and Wang, 2019, Yang and Wang, 2019b,a, Agarwal et al., 2020]; crucially, however, these works opt for computing a factorization of the transition function, rather than the value function. Again, we suspect that this preference stems from the immediate inaccessibility of $Q^\star$, a fact that we show need not be an obstacle when adopting a Bayesian view of efficient exploration. Moreover, leveraging such a Bayesian approach to exploration potentially avoids the known pitfalls [Osband and Van Roy, 2017] of alternative methods that employ optimism in the face of uncertainty [Yang and Wang, 2019b]. Most related to this work is the approach of Shah et al. [2020] who do in fact aim to learn the singular value decomposition of $Q^\star$ and take advantage of low-rank structure; important differences from this work include a focus on MDPs with continuous state-action spaces along with assumptions on the Lipschitz continuity of $Q^\star$, access to a generative model for sampling transitions, and access to anchor states (states that are representative of each latent abstract state [Donoho and Stodden, 2004]) for heuristically guiding exploration/data collection. In contrast, this work is concerned with discrete-action MDPs (though extensions to the continuous-control setting are a natural future direction) and makes no assumptions on $Q^\star$ beyond being low rank (Assumption 1).

For understanding when it is prudent for an agent to pursue latent task structure in the form of a state abstraction, we find the multi-task setting to be a natural candidate. In the context of the low-rank matrix factorization outlined in the previous section, this amounts to asserting that the optimal value functions of all tasks share a common, latent factor $\mathbf{\Phi}$ while each individual task t also yields a specific matrix of abstract-state values, $\mathbf{\Psi}_t$. Consequently, an agent interacting to solve all tasks in parallel can greatly benefit from synthesizing shared task structure. Various prior works already assess such benefits of multiple tasks for supervised learning [Caruana, 1997, Baxter, 1997, 2000]. While work that formally develops this connection for reinforcement learning is still nascent [D'Eramo et al., 2019, Calandriello et al., 2014], empirical examples of this phenomenon are well-established [Sutton et al., 2011, Jaderberg et al., 2016]. Our work can be seen as a simple mathematical model for studying this phenomenon in the Bayesian RL setting.

4 Background

In this section, we provide background on provably-efficient approaches to addressing the exploration-exploitation trade-off in sequential decision-making problems. The mechanisms employed by these approaches for representing uncertainty will play a central role in our algorithm for learning a value-based state abstraction.

4.1 Information-Directed Sampling

A central challenge that all sequential decision-making agents must confront is that of exploration; an agent must strike a delicate balance between acquiring new knowledge in the hope of improving future performance or capitalizing on the information it has acquired thus far. Early results for provably-efficient reinforcement-learning algorithms (designated PAC-MDP [Strehl et al., 2009]) hinge on the sufficiency of an agent's exploration strategy for fully exploring the MDP [Kakade, 2003], typically based on a principle of optimism in the face of uncertainty [Kearns and Singh, 2002, Brafman and Tennenholtz, 2002, Bubeck and Cesa-Bianchi, 2012]. In recent years, exploration techniques that facilitate stronger theoretical guarantees have come about by leveraging estimates of an agent's epistemic uncertainty or uncertainty stemming from parameter estimation (rather than the aleatoric uncertainty driven by stochasticity in data) [Chapelle and Li, 2011, Russo and Van Roy, 2016, Osband et al., 2016a, Agrawal and Jia, 2017, O'Donoghue et al., 2018, Osband et al., 2019].

In the context of multi-armed bandit problems, an agent maintains uncertainty over the individual reward or payoff functions at each arm. With uncertainty in the rewards of all arms driving uncertainty over optimal actions, one choice is for the agent to employ an exploration scheme based on Thompson sampling (TS) [Thompson, 1933, Agrawal and Goyal, 2012, 2013, Russo and Van Roy, 2016] whereby an agent acts optimally with respect to a single sample drawn from its posterior beliefs at each time period. This idea naturally scales to the full reinforcement-learning scenario wherein posterior beliefs are maintained over the optimal action-value function $Q^\star$ [Osband et al., 2016a, O'Donoghue et al., 2018, Osband et al., 2019].

A significant advance on the aforementioned exploration scheme is the algorithmic design principle known as information-directed sampling (IDS) [Russo and Van Roy, 2018]. While previous approaches follow suit with Thompson sampling and act optimally according to posterior samples, IDS algorithms execute a policy at each time period that solves the following minimization problem

$$\pi_h = \min_{\pi \in \Delta(\mathcal{A})} \frac{(\mathbb{E}[\Delta_h(\pi)])^2}{\mathcal{I}(\theta; (S_h, A_h)|\mathcal{E}_{h-1})}$$

where $\mathbb{E}[\Delta_h(\pi)]$ denotes the expected regret of policy π under the agent's current posterior beliefs and $\mathcal{I}(\theta; (S_h, A_h)|\mathcal{E}_{h-1})$ denotes the expected information gain between the behavior at timestep h and the environment parameters θ, conditioned on the history of episodes collected thus far, $\mathcal{E}_{h-1}$. For a multi-armed bandit problem, θ reflects the reward or payoffs at each arm whereas, for a MDP, θ captures the environment transition function and reward function. IDS embodies an intuitive principle that, rather than being biased exclusively towards the optimal action of one posterior sample, an agent should be incentivized to take one or more suboptimal actions so long as they are informative and revelatory of the underlying environment, θ. Here, the information gain term in the denominator of the information ratio above quantifies this level of informativity, weighing it against the agent's desire to minimize regret over its lifetime. Again, while simple and powerful, IDS is only a design principle to guide the development of practical, efficient algorithms. In the next section, we discuss a deliberate choice of how to represent an agent's posterior beliefs that yields a concrete instantiation of IDS for our algorithm.

4.2 Hypermodels

When faced with a need to represent epistemic uncertainty, scalability becomes an immediate challenge. For the popular choice of neural networks as expressive function approximators, the common practice has been to employ finite ensembles [Osband et al., 2016a, Lu and Van Roy, 2017] which maintain several copies of neural network weights for a single architecture. A sample from this posterior of K ensemble members is acquired by first sampling $z \sim \text{Uniform}(K)$ and then running the ensemble member indexed by z.

Recognizing the poor scaling of computationally efficiency for ensemble sampling with increasing K, Dwaracherla et al. [2020] introduce hypermodels as a scalable alternative to representing epistemic uncertainty. In short, hypermodels rely on a space of indexes $\mathcal{Z}$ as well as a corresponding reference distribution $p_z \in \Delta(\mathcal{Z})$. For a given choice of base model $f_\theta : \mathcal{X} \mapsto \mathcal{Y}$ parameterized by $\theta \in \Theta$, a hypermodel with parameters ν, $\mathfrak{H}_\nu : \mathcal{Z} \mapsto \Theta$ maps a single index to a particular instantiation of base model. Thus, by sampling an index $z \sim p_z(\cdot)$, the function $f_{\mathfrak{H}_\nu(z)}$ represents a sample from an approximate posterior. Given a dataset $\mathcal{D}$, the loss function $\mathcal{L}(\nu, \mathcal{D})$ to optimize hypermodel parameters ν will vary depending on the base model and task. We defer the definition of $\mathcal{L}(\nu, \mathcal{D})$ to the next section.

5 Approach

5.1 Approximate Posterior over State Abstractions

Our core contribution is an approach for jointly learning a state abstraction ϕ without prior knowledge of $Q^\star$. To do this, we leverage our assumption of $Q^\star$ as being low rank and we explicitly maintain two separate hypermodels $\mathfrak{H}_\nu^\phi, \mathfrak{H}_\nu^\psi$ as approximate posterior distributions over $\phi^\star$ and $\psi^\star$ respectively. It is important to note that while $\mathfrak{H}_\nu^\phi : \mathcal{Z} \mapsto \Phi$ is a standard hypermodel [Dwaracherla et al., 2020] mapping indices to instances of ϕ, $\mathfrak{H}_\nu^\psi : \mathcal{Z} \times \Phi \mapsto \Psi$ is a conditional hypermodel over possible functions ψ. The intuition here is that once an agent samples from its posterior beliefs over $\phi^\star$, the corresponding sample from its beliefs over abstract-state values must be conditioned on the particular sample $\phi \sim \mathfrak{H}_\nu^\phi(\cdot)$. More succinctly, an agent's posterior beliefs over the optimal value function is obtained by first sampling $z \sim p_z(\cdot)$ and then $\hat{Q}^\star \sim \langle \mathfrak{H}_\nu^\phi(z), \mathfrak{H}_\nu^\psi(z, \mathfrak{H}_\nu^\phi(z)) \rangle$.

Recall that we have yet to define the objective function for optimizing the hypermodels to represent an approximate posterior over $Q^\star$. Given a minibatch of past experiences $\tilde{\mathcal{D}}$, we optimize the following loss function $\mathcal{L}(\nu, \tilde{\mathcal{D}})$:

$$R(z_\phi, z_\psi, \nu) \triangleq \lambda ||\mathfrak{H}_\nu^\phi(z_\phi) - \mathfrak{H}_{\nu_0}^\phi(z_\phi)||_2^2 + \lambda ||\mathfrak{H}_\nu^\psi(z_\psi, \mathfrak{H}_\nu^\phi(z_\phi)) - \mathfrak{H}_{\nu_0}^\psi(z_\psi, \mathfrak{H}_{\nu_0}^\phi(z_\phi))||_2^2$$

$$\hat{Q}^\star_{z_\phi, z_\psi}(s,a) \triangleq \langle \mathfrak{H}_\nu^\phi(z_\phi)(s), \mathfrak{H}_\nu^\psi(z_\psi, \mathfrak{H}_\nu^\phi(z_\phi))(a) \rangle$$

$$\mathcal{L}(\nu, \tilde{\mathcal{D}}) = \mathbb{E}_{\substack{z_\phi \sim p_z(\cdot) \\ z_\psi \sim p_z(\cdot)}} [\frac{1}{m} \sum_{(s,a,r,s' \eta_\phi, \eta_\psi) \in \tilde{\mathcal{D}}} (r + \gamma \max_{a' \in \mathcal{A}} \hat{Q}^\star_{z_\phi, z_\psi}(s', a') + \eta_\phi^T z_\phi + \eta_\psi^T z_\psi - \hat{Q}^\star_{z_\phi, z_\psi}(s,a))^2 + R(z_\phi, z_\psi, \nu)]$$

where ν_0 denotes the initial vector of hypermodel parameters, the η_ϕ, η_ψ terms denote random Gaussian perturbations of the target values, and λ is a regularization coefficient. This loss function encapsulates a randomized least-squares value iteration (RLSVI) [Osband et al., 2016b] approach to maintaining an approximate posterior distribution over $Q^\star$.

5.2 Variance-IDS

In the previous section, we specify how hypermodels can be used to maintain approximate posterior distributions over state abstraction and abstract state values. The final outstanding component that must be specified is how these approximate posterior distributions can be folded into an algorithm that instantiates IDS as the core exploration strategy. To do this, we leverage variance-IDS as introduced in Russo and Van Roy [2018] and as specified for hypermodels in Dwaracherla et al. [2020]. For clarity, we present the derivation of variance-IDS.

Recall the definition of the mutual information between two random variables X, Y:

$$\mathcal{I}(X;Y) = \mathbb{E}_X[D_{\mathrm{KL}}(p(Y|X)||p(Y))]$$

Noting the definition contains a Kullback-Leibler (KL) divergence term, we also define Pinkser's inequality [Pinsker, 1960]

$$D_{\mathrm{TV}}(p(X)||q(X)) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p(X)||q(X))}$$

where $D_{\mathrm{TV}}(p(X)||q(X))$ denotes the total variation distance between distributions p, q. Lastly, we note that the total variation distance is an integral probability metric (IPM) [Müller, 1997] which, for a random variable X with support $\mathcal{X}$, is defined as:

$$D_{\mathrm{TV}}(p(X)||q(X)) = \sup_{\substack{f:\mathcal{X} \mapsto \mathbb{R} \\ ||f||_\infty \leq 1}} \mathbb{E}_{p(X)}[f(X)] - \mathbb{E}_{q(X)}[f(X)]$$

where the supremum is taken with respect to all witness functions $f : \mathcal{X} \mapsto \mathbb{R}$ with infinity norm bounded by 1. Putting all the pieces together, we recall that IDS balances regret minimization with the selection of informative actions, where informativeness of agent behavior is measured by the (conditional) mutual information between the state-action pair observed at timestep h and the true environment parameters θ, $\mathcal{I}(\theta; S_h, A_h)$. Following Russo and Van Roy [2018], this term can be lower bounded as follows:

$$\begin{aligned}
\mathcal{I}(\theta; S_h, A_h) &= \mathbb{E}_\theta[D_{KL}(p(S_h, A_h|\theta)||p(S_h, A_h))] \\
&\geq 2\mathbb{E}_\theta[(D_{TV}(p(S_h, A_h|\theta)||p(S_h, A_h)))^2] \\
&= 2\mathbb{E}_\theta[(\sup_{\substack{f:\mathcal{S}\times\mathcal{A}\mapsto\mathbb{R} \\ ||f||_\infty \leq 1}} (\mathbb{E}_{p(S_h,A_h|\theta)}[f(S_h, A_h)] - \mathbb{E}_{p(S_h,A_h)}[f(S_h, A_h)])^2] \\
&\geq 2\mathbb{E}_\theta[(\mathbb{E}[Q^\star(S_h, A_h)|\theta] - \mathbb{E}[Q^\star_h(S_h, A_h)])^2] \\
&= 2\mathbb{V}[\mathbb{E}[Q^\star(S_h, A_h)|\theta]]
\end{aligned}$$

where the steps follow from the definition of mutual information, Pinsker's inequality, the IPM form of the total variation distance, the definition of supremum, and the definition of variance. Crucially, and just as with the original mutual information term, all of the expectations and variances above are conditioned on the past history of interactions up to this timestep. Thus, the above shows that the information gain at timestep h is lower bounded by the extent to which $Q^\star$-values produced by action a vary under the identity of the environment parameters θ. Consequently, those actions with high variance in $Q^\star$-values under the agent's current posterior beliefs are deemed to be the most informative.

5.3 Posterior State-Abstraction Sampling

We present the full algorithm for PS2 as Algorithm 1 with the explicit choice of hypermodels for representing an agent's beliefs. The algorithm proceeds according to variance-IDS by first drawing K samples from the agent's current posterior beliefs over $Q^\star$ and computing the requisite quantites for variance-IDS, namely the expected squared regret and expected variance for each action. The resulting experience collected from the environment is then accumulated via experience replay [Lin, 1992, Mnih et al., 2015] for incrementally updating the hypermodels.

Although not denoted here for clarity, this is mutual information term is conditioned on the entire history of all past interactions up to this point.

Algorithm 1 Posterior State-Abstraction Sampling with Hypermodels & Variance-IDS

```
Data: Reference distributions $p_{z_\phi}, p_{z_\psi} = \mathcal{N}(0, I)$, Minibatch size $m$, Learning rate $\alpha$, Regularization parameters $\lambda$, Noise variances $\sigma_\phi^2$, $\sigma_\psi^2$
Initialize $\mathfrak{H}_\nu^\phi, \mathfrak{H}_\nu^\psi$
$\mathcal{D} \leftarrow \emptyset$
for $t$ $1, \dots, T$ do
    for $h$ $1, \dots, H$ do
        Sample $Z = \{(z_\phi, z_\psi)_1, \dots, (z_\phi, z_\psi)_K\}, z_\phi \sim p_{z_\phi}(\cdot), z_\psi \sim p_{z_\psi}(\cdot)$
        $\hat{Q}^\star_{(z_\phi, z_\psi)}(s, a) \triangleq \langle \mathfrak{H}_\nu^\phi(z_\phi)(s), \mathfrak{H}_\nu^\psi(z_\psi, \mathfrak{H}_\nu^\phi(z_\phi))(a) \rangle$
        $\hat{\Delta}_a(s) \triangleq \frac{1}{|Z|} \sum_{i=1}^{K} \max_{a^\star \in \mathcal{A}} \hat{Q}^\star_{(z_\phi, z_\psi)_i}(s, a^\star) - \hat{Q}^\star_{(z_\phi, z_\psi)_i}(s, a)]$
        $\tilde{Z}(a) = \{(z_\phi, z_\psi) | (z_\phi, z_\psi) \in Z, a = \arg\max_{a^\star \in \mathcal{A}} \hat{Q}^\star_{(z_\phi, z_\psi)}(s, a^\star)\}$
        $\hat{v}_a(s) \triangleq \sum_{a^\star \in \mathcal{A}} \frac{|\tilde{Z}(a^\star)|}{|Z|} (\frac{1}{|\tilde{Z}(a^\star)|} \sum_{(z_\phi, z_\psi) \in \tilde{Z}(a^\star)} \hat{Q}^\star_{(z_\phi, z_\psi)}(s, a) - \frac{1}{|Z|} \sum_{(z_\phi, z_\psi) \in Z} \hat{Q}^\star_{(z_\phi, z_\psi)}(s, a))^2$
        $\pi(\cdot | s_h) = \min_{\pi \in \Delta(\mathcal{A})} \frac{\mathbb{E}_{a \sim \pi}[(\hat{\Delta}_a(s))^2]}{\mathbb{E}_{a \sim \pi}[\hat{v}_a(s)]}$
        Take action $a_h \sim \pi(\cdot | s_h)$ and observe $r_h, s_{h+1}$
        Sample random perturbations $\eta_\phi \sim \mathcal{N}(0, \sigma_\phi^2 I), \eta_\psi \sim \mathcal{N}(0, \sigma_\psi^2 I)$
        $\mathcal{D} \rightarrow \mathcal{D} \cup \{(s_h, a_h, r_h, s_{h+1}, \eta_\phi, \eta_\psi)\}$
        Sample random minibatch $\tilde{\mathcal{D}} \overset{m}{\sim} \text{Uniform}(\mathcal{D})$
        $\nu \leftarrow \nu - \alpha \nabla_\nu \mathcal{L}(\nu, \tilde{\mathcal{D}})$
    end
end
```

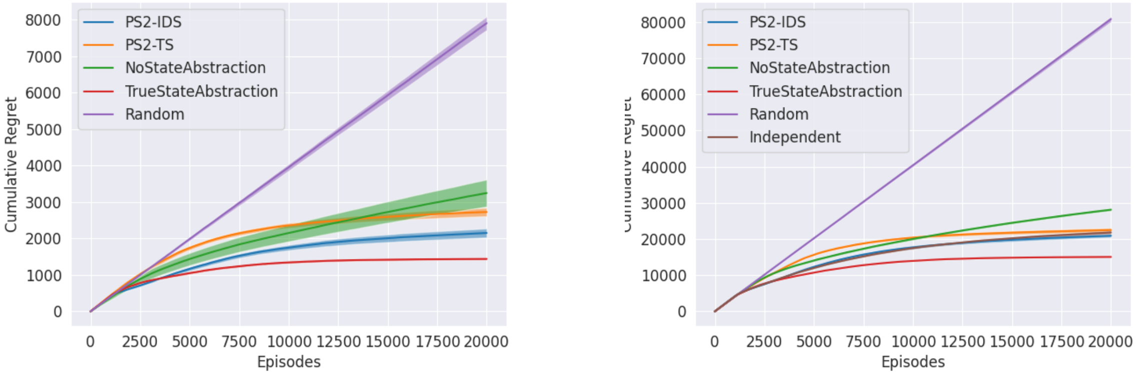
For the multi-task learning setting, the algorithm is nearly identical, with the caveat that index samples must be drawn for each hypermodel over abstract-state values (with one hypermodel per task); the hypermodel over state abstractions is shared across all tasks.

6 Experiments & Discussion

We recall that the primary goal of this paper is twofold, (1) offering PS2 as a practical approach for synthesizing a state abstraction based on structure within $Q^\star$ (as opposed to, for instance, structure in the transition function) and (2) identifying when the pursuit of such latent structure is particularly advantageous. In this section, we outline a recipe for randomly generating contextual bandit problems that allow us to empirically address these goals.

Concretely, given specific values for the number of states (S), actions (A), and abstract states (M), we generate contextual bandit problems by randomly sampling a reward function with sparse latent structure. The factorization $\mathbf{Q}^\star = \mathbf{\Phi}\mathbf{\Psi}^T$ is formed by first generating $\mathbf{\Phi} \in \mathbb{R}^{S \times M}$ through the random sampling of S one-hot vectors of length M. The abstract-state values $\mathbf{\Psi} \in \mathbb{R}^{A \times M}$ are then each drawn uniformly at random from $[0, 1]$. For our multi-task experiments, the procedure for sampling $\mathbf{\Phi}$ remains unchanged and the procedure for sampling $\mathbf{\Psi}_t$ is repeated for each task t. We define instantaneous regret as $\Delta_t = |\max_{a^\star} Q^\star(s_t, a^\star) - Q^\star(s_t, a_t)|$ and define cumulative regret over T episodes or time periods as $\sum_{t=1}^{T} \Delta_t$. For multi-task experiments, instantaneous regret is summed across all tasks. In all experiments we use the Adam optimizer [Kingma and Ba, 2014] with a learning rate of 0.001, a batch size of 1024, 128 index samples per timestep, noise variances of 0.25, and regularization parameter of 0.001. All shading in figures denote 95% confidence intervals computed across five random seeds. All hypermodels are parameterized as Gaussian distributions with index samples of appropriate dimension drawn iid from $\mathcal{N}(0, 1)$. For each contextual-bandit problem, we evaluate the following algorithms:

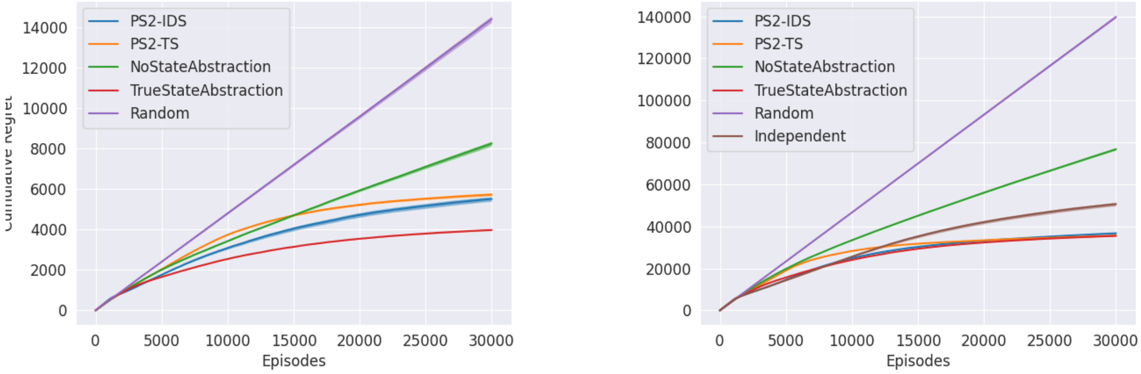
- **PS2-IDS** – Algorithm 1 with two hypermodels over state abstractions and abstract-state values respectively. The former hypermodel is parameterized as a Gaussian distribution with diagonal covariance matrix. The latter hypermodel over values is also represented as a Gaussian distribution with diagonal covariance whose mean is a linear function of an input state abstraction $\mathbf{\Phi} \in \mathbb{R}^{S \times M}$.
- **PS2-TS** – Identical parameterization to PS2-IDS except, instead of using variance-IDS, applies Thompson sampling for action selection by drawing a single $Q^\star$ and then acting greedily with respect to the sample.
- **NoStateAbstraction** – An implementation of variance-IDS that, instead of learning a state abstraction, directly maintains a single hypermodel of the agent's beliefs over $\mathbf{Q}^\star \in \mathbb{R}^{S \times A}$.
- **TrueStateAbstraction** – An implementation of variance-IDS that is given the true state abstraction $\mathbf{\Phi}$ a priori and only maintains a hypermodel for learning the corresponding abstract-state values.
- **Independent** – For multi-task experiments with T distinct tasks, this algorithm maintains T instances of PS2-IDS.
- **Random** – Selects actions at each timestep uniformly at random.



(a) Single-task setting

(b) Multi-task setting with 10 tasks.

Figure 1: Contextual bandit with $S = 10, A = 10, M = 5$.



(a) Single-task setting

(b) Multi-task setting with 10 tasks.

Figure 2: Contextual bandit with $S = 30, A = 30, M = 5$.

Figures 1 and 2 show results for randomly generated contextual bandit problems of two different sizes. Firstly, in the single-task setting, the relationship between the algorithm that has the true state abstraction computed a priori vs. the algorithm that does not pursue any abstraction at all reaffirms existing empirical results that highlight the advantages of leveraging structure in the value function [Abel et al., 2016]. Notably, both PS2 methods are able to achieve performance between these two extremes, identifying the underlying abstraction of the environment to more efficiently arrive at optimal behavior. This point becomes even more apparent in the multi-task setting (Figures 1b and 2b) where interaction with multiple tasks amplifies the signal provided to the agent for distilling the underlying state abstraction. Due to the small problem size, Figure 1b shows little improvement between PS2 and the algorithm that attempts to solve each task in isolation. With a slightly larger problem in Figure 2b, however, we observe a substantial improvement in PS2 as it is able to better exploit information from all tasks to capture shared structure. Finally, it is theoretically know that IDS has a stronger performance guarantee than Thompson sampling [Russo and Van Roy, 2018]. Our experiments confirm this relationship with PS2-IDS matching or outperforming Thompson sampling.

7 Conclusion

We have examined state abstractions as a mechanism for facilitating sample-efficient reinforcement learning. While various forms of state abstraction model structure in different components of a MDP, this work places particular focus on those which attempt to leverage structure in the optimal value function. While prior works that study this abstraction type have often been unrealistic in their acquisition of the abstraction itself, we take a Bayesian perspective and leverage insights from past work on provably-efficient, deep exploration. Concretely, we introduce an algorithmic design principle wherein an agent's beliefs over the optimal value function factor into separate posterior distributions over abstract states and abstract-state values, respectively. Empirically, we demonstrate that an agent whose beliefs have been factored in this manner can learn more efficiently than those that attempt to directly estimate values and ignore latent structure altogether. The experiments in this work have been limited to the contextual bandit setting; identifying the right sampling procedure for generating random MDPs with the appropriate structural properties that are conducive for any algorithm, including PS2, to learn state abstractions is an active direction for future work.

References

David Abel, D Ellis Hershkowitz, and Michael L Littman. Near optimal behavior via approximate state abstraction. In *Proceedings of the 33rd International Conference on International Conference on Machine*

Learning-Volume 48, pages 2915–2923, 2016.

David Abel, Dilip Arumugam, Lucas Lehnert, and Michael Littman. State abstractions for lifelong reinforcement learning. In *International Conference on Machine Learning*, pages 10–19, 2018.

David Abel, Dilip Arumugam, Kavosh Asadi, Yuu Jinnai, Michael L Littman, and Lawson LS Wong. State abstraction as compression in apprenticeship learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3134–3142, 2019.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*, 2020.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1, 2012.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107, 2013.

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pages 1184–1194, 2017.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In *Advances in neural information processing systems*, pages 89–96, 2009.

André MS Barreto, Rafael L Beirigo, Joelle Pineau, and Doina Precup. Incremental stochastic factorization for online reinforcement learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39, 1997.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

Bahram Behzadian and Marek Petrik. Feature selection by singular value decomposition for reinforcement learning. In *Proceedings of the ICML Prediction and Generative Modeling Workshop*, 2018.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

Dimitri P Bertsekas, David A Castanon, et al. Adaptive aggregation methods for infinite horizon dynamic programming. 1988.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Daniele Calandriello, Alessandro Lazaric, and Marcello Restelli. Sparse multi-task reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 819–827, 2014.

Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Alexander L Chistov and D Yu Grigor'Ev. Complexity of quantifier elimination in the theory of algebraically closed fields. In *International Symposium on Mathematical Foundations of Computer Science*, pages 17–31. Springer, 1984.

Thomas Dean and Robert Givan. Model minimization in markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 106–111. AAAI Press, 1997.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Sharing knowledge in multi-task deep reinforcement learning. In *International Conference on Learning Representations*, 2019.

Shi Dong, Benjamin Van Roy, and Zhengyuan Zhou. Provably efficient reinforcement learning with aggregated states. *arXiv preprint arXiv:1912.06366*, 2019.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, pages 1141–1148, 2004.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674, 2019.

Yaqi Duan, Tracy Ke, and Mengdi Wang. State aggregation learning from markov transition data. In *Advances in Neural Information Processing Systems*, pages 4488–4497, 2019.

Vikranth Dwaracherla, Xiuyuan Lu, Morteza Ibrahimi, Ian Osband, Zheng Wen, and Benjamin Van Roy. Hypermodels for exploration. In *International Conference on Learning Representations*, 2020.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. 2004.

Norman Ferns, Pablo Samuel Castro, Doina Precup, and Prakash Panangaden. Methods for computing state similarity in markov decision processes. *arXiv preprint arXiv:1206.6836*, 2012.

Suriya Gunasekar, Ayan Acharya, Neeraj Gaur, and Joydeep Ghosh. Noisy matrix completion using alternating minimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 194–209. Springer, 2013.

Moritz Hardt. Understanding alternating minimization for matrix completion. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 651–660. IEEE, 2014.

Moritz Hardt, Raghu Meka, Prasad Raghavendra, and Benjamin Weitz. Computational limits for matrix completion. In *Conference on Learning Theory*, pages 703–725, 2014.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691, 2019.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 179–188, 2015.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. *arXiv preprint arXiv:2002.02794*, 2020.

Nicholas K Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 752–757, 2005.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning.* PhD thesis, 2003.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE transactions on information theory*, 56(6):2980–2998, 2010a.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

George Konidaris. On the necessity of abstraction. *Current opinion in behavioral sciences*, 29:1–7, 2019.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive sampling. In *Advances in neural information processing systems*, pages 836–844, 2013.

Akshay Krishnamurthy and Aarti Singh. On the power of adaptivity in matrix completion and approximation. *arXiv preprint arXiv:1407.3619*, 2014.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2006.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.

Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. In *Advances in neural information processing systems*, pages 3258–3266, 2017.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. *arXiv preprint arXiv:1911.05815*, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, pages 2701–2710, 2017.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016a.

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pages 2377–2386, 2016b.

Ian Osband, Benjamin Van Roy, Daniel J Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Brendan O'Donoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. The uncertainty bellman equation and exploration. In *International Conference on Machine Learning*, pages 3836–3845, 2018.

Mark Semenovich Pinsker. Information and information stability of random variables and processes. 1960.

Martin L. Puterman. *Markov Decision Processes—Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, 1994.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec): 3413–3430, 2011.

Benjamin Recht and Christopher Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2):201–226, 2013.

Daniel Russo. Approximation benefits of policy gradient methods with aggregated states. *arXiv preprint arXiv:2007.11684*, 2020.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018.

Devavrat Shah, Dogyoon Song, Zhi Xu, and Yuzhe Yang. Sample efficient reinforcement learning via low-rank matrix estimation. *arXiv preprint arXiv:2006.06135*, 2020.

Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec., March 1959*, 4:142–163, 1959.

Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. Reinforcement learning with soft state aggregation. In *Advances in neural information processing systems*, pages 361–368, 1995.

Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Richard S Sutton and Andrew G Barto. Introduction to reinforcement learning. 1998.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768, 2011.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1-3):59–94, 1996.

Benjamin Van Roy. Performance loss bounds for approximate value iteration with state aggregation. *Mathematics of Operations Research*, 31(2):234–244, 2006.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Ward Whitt. Approximations of dynamic programs, i. *Mathematics of Operations Research*, 3(3):231–243, 1978.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004, 2019a.

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

Anru Zhang and Mengdi Wang. Spectral state compression of markov processes. *IEEE Transactions on Information Theory*, 66(5):3202–3231, 2019.