Contextual Decision Processes with low Bellman rank are PAC-Learnable

Nan Jiang [1] **Akshay Krishnamurthy** [2] **Alekh Agarwal** [3] **John Langford** [3] **Robert E. Schapire** [3]

Abstract

This paper studies systematic exploration for reinforcement learning (RL) with rich observations and function approximation. We introduce *contextual decision processes* (CDPs), that unify most prior RL settings. Our first contribution is a complexity measure, the *Bellman rank*, that we show enables tractable learning of near-optimal behavior in CDPs and is naturally small for many well-studied RL models. Our second contribution is a new RL algorithm that does systematic exploration to learn near-optimal behavior in CDPs with low Bellman rank. The algorithm requires a number of samples that is polynomial in all relevant parameters but independent of the number of unique contexts. Our approach uses Bellman error minimization with optimistic exploration and provides new insights into efficient exploration for RL with function approximation.

1. Introduction

In this paper, we study reinforcement learning (RL) problems where the agent receives rich sensory observations from the environment, forms complex contexts from sensorimotor streams, uses function approximation to generalize to unseen contexts, and must perform *systematic exploration* to learn efficiently. Such problems are at the core of empirical RL research (e.g., Mnih et al., 2015; Bellemare et al., 2016), yet no existing theory provides rigorous and satisfactory guarantees in a general setting. This situation motivates an important question: *how can we solve RL problems where exploration is critical and the agent receives rich observations, in a sample-efficient manner?*

To answer the question, we propose a new formulation, *Contextual Decision Processes* (CDPs), to capture a large class of sequential decision-making problems: CDPs generalize MDPs where the state forms the context (Ex. 1) and POMDPs where the history forms the context (Ex. 2), and can be much more concise than alternative formulations based on sufficient statistics (e.g., Hutter, 2005). We define CDPs in Section 2, and the learning goal is to find a near-optimal policy for a CDP with the help of a value-function approximator in a sample-efficient manner.[1]

A structural assumption: When the context space is very large or infinite, as is common in practice, lower bounds that are *exponential in the problem horizon* preclude efficient learning in CDPs, even when simple function approximators are used. However, RL problems arising in applications are often far more benign than the pathological lower bound instances, and we identify a structural assumption capturing this intuition. As our first major contribution, we define a notion of *Bellman factorization* (Definition 5) in Section 3, and focus on problems with low *Bellman rank*.

At a high level, Bellman rank is an algebraic dimension capturing the interplay between the CDP and the value-function approximator that we show is small for many previously-studied settings. For example, every MDP with a tabular value-function has Bellman rank bounded by the rank of its transition matrix, which is at most the number of states but can be considerably smaller. For a POMDP with reactive value-functions, the Bellman rank is at most the number of hidden states and has no dependence on the observation space. We provide other instances of low Bellman rank including Linear Quadratic Regulators and Predictive State Representations. Overall, CDPs with a small Bellman rank yield a unified framework for a large class of sequential decision making problems.

A new algorithm: Our second contribution is a new algorithm for episodic RL called OLIVE (Optimism Led Iterative Value-function Elimination), detailed in Section 4.1. OLIVE iteratively refines a space of candidate Q-value functions $\mathcal{F}$. At each iteration, it chooses a value function f using an optimistic criterion and collects trajectories from the corresponding greedy policy π_f. If π_f attains a high-value, the algorithm terminates and outputs f. Other-

[1]University of Michigan, Ann Arbor [2]University of Massachusetts, Amherst [3]Microsoft Research, New York. Correspondence to: Nan Jiang <nanjiang@umich.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

[1]Throughout the paper, by sample-efficient we mean a number of trajectories that is polynomial in the problem horizon, number of actions, Bellman rank (to be introduced), and polylogarithmic in the number of candidate value-functions.

Model	Tabular MDP	Low-rank MDP	Reactive POMDP	Reactive PSR	LQR
Bellman rank	# states	rank	# hidden states	PSR rank	# state variables
PAC Learning	known	new	extended	new	known[3]

Table 1. Summary of settings having low Bellman rank, with formal statements in Section 3 (Proposition 1 to 5, from left to right in the table). The 2nd row gives the parameters that bound the Bellman rank. In the 3rd row, "known" means that sample-efficient algorithms already exist (e.g., tabular MDPs), "extended" means our results substantially extend previous work (e.g., (Krishnamurthy et al., 2016) for reactive POMDPs), and "new" means our result gives the first sample-efficient algorithm (e.g., MDPs with low-rank transitions).

wise, it eliminates all $g \in \mathcal{F}$ which violate certain Bellman equations under trajectories generated by π_f and performs the next iteration with this refined class of functions.

A PAC guarantee: We prove that OLIVE performs sample-efficient learning in CDPs with a small Bellman rank (Section 4.2). Concretely, when the $Q^\star$-function for the CDP is contained in $\mathcal{F}$, OLIVE requires $\tilde{\mathcal{O}}(M^2H^3K\log(N/\delta)/\epsilon^2)$ trajectories to find an ϵ-suboptimal policy,[2] where M is the Bellman rank, H is the length of a trajectory, K is the number of actions, N is the cardinality of $\mathcal{F}$, and δ is the failure probability.

Importantly, the sample complexity bound has a logarithmic dependence on N, enabling powerful function approximation, and no direct dependence on the size of the context space, which can be very large or infinite. As many existing models, including the ones highlighted in Table 1, have low Bellman rank, the result immediately implies sample-efficient learning in all of these settings.[3]

The main PAC-guarantee can be extended in several ways, discussed in Appendix A. Specifically, OLIVE is robust to the failure of our assumptions, can adapt to unknown Bellman rank, and can handle infinite function classes with bounded statistical complexity. These extensions demonstrate that the Bellman rank robustly captures the difficulty of exploration in sequential-decision making problems.

To summarize, this work advances our understanding of RL with complex observations where long-term planning and exploration are critical. While OLIVE represents an exponential advance in statistical efficiency, its computational complexity, which is polynomial in N, is intractable for the powerful function classes of interest. This computational issue must be addressed before we can empirically evaluate the effectiveness of the proposed algorithm. We discuss this and other future directions in Section 6.

Related work. There is rich theoretical literature on RL in tabular settings, including MDPs (Kearns & Singh, 2002; Brafman & Tennenholtz, 2003; Strehl et al., 2006) and POMDPs (Azizzadenesheli et al., 2016) with small state and observation spaces, with an emphasis on sophisticated exploration to find near-optimal policies in a sample-efficient manner. While there have been extensions to large state spaces (Kakade et al., 2003; Jong & Stone, 2007; Pazis & Parr, 2016), these approaches fail to be a good fit for practical scenarios where the environment is typically perceived through complex observations such as image, text, or audio signals. Alternatively, Monte Carlo Tree Search (MCTS) methods can handle large state spaces, but only at the cost of exponential dependence on the planning horizon (Kearns et al., 2002; Kocsis & Szepesvári, 2006).

Closest to our work are the results of Wen & Van Roy (2013) and Krishnamurthy et al. (2016), which also obtain sample complexity independent of the number of unique contexts, but only under deterministic dynamics and other special structures. In contrast, we study a much broader class of problems with relatively mild conditions.

On the empirical side, recent successes on both the Atari platform (Mnih et al., 2015; Wang et al., 2015) and Go (Silver et al., 2016) have sparked a flurry of research interest. These approaches leverage advances in deep learning for powerful function approximation, but typically use simple strategies, such as ϵ-greedy, for exploration. Better exploration strategies, such as pseudo-counts in Bellemare et al. (2016), and combining MCTS with function approximation (e.g., Silver et al. (2016)), typically require strong domain knowledge and large amounts of data to be successful.

Hallak et al. (2015) have proposed Contextual MDPs, where each context parametrizes an MDP. In contrast, our use of contexts is in analogy with contextual bandits (Langford & Zhang, 2008), and is similar to state features in RL.

2. Contextual Decision Processes (CDPs)

In this section, we introduce a new model, called a Contextual Decision Process, as a unified framework for reinforcement learning with rich observations.

2.1. Model and Examples

CDPs make minimal assumptions to capture a general class of RL problems and are defined as follows.

Definition 1 (Contextual Decision Process (CDP))**.** *A*

[2] A logarithmic dependence on a norm parameter ζ is omitted here, as ζ is polynomial in most cases.

[3] Our algorithm requires discrete action spaces and does not immediately apply to LQRs; see more discussion in Section 3.

(finite-horizon) CDP is defined as a tuple $(\mathcal{X}, \mathcal{A}, H, P)$*, where* $\mathcal{X}$ *is the context space,* $\mathcal{A}$ *is the action space, and* H *is the horizon of the problem.* $P = (P_\varnothing, P_+)$ *is the* system descriptor*, where* $P_\varnothing \in \Delta(\mathcal{X})$ *is a distribution over initial contexts, that is* $x_1 \sim P_\varnothing$*, and* $P_+ : (\mathcal{X} \times \mathcal{A} \times \mathbb{R})^* \times \mathcal{X} \times \mathcal{A} \to \Delta(\mathbb{R} \times \mathcal{X})$ *elicits the next reward and context from the interactions so far* $x_1, a_1, r_1, \ldots, x_h, a_h$*:*

$$(r_h, x_{h+1}) \sim P_+(x_1, a_1, r_1, \ldots, x_h, a_h).$$

In a CDP, the agent interacts with the environment in episodes. In an episode, the agent observes a context x_1, takes action a_1, receives reward r_1 and observes x_2, repeating H times. A policy $\pi : \mathcal{X} \to \mathcal{A}$ specifies the agent's decision-making strategy, i.e. $a_h = \pi(x_h)$, $\forall h \in [H]$, and induces a distribution over trajectories $(x_1, a_1, r_1, \ldots, x_H, a_H, r_H, x_{H+1})$ via the system descriptor P. The value of a policy, V^π, is defined as

$$V^\pi = \mathbb{E}_P \left[\sum_{h=1}^H r_H \;\middle|\; a_{1:H} \sim \pi \right], \tag{1}$$

where $a_{1:H} \sim \pi$ abbreviates for $a_1 = \pi(x_1), \ldots, a_H = \pi(x_H)$. Throughout, the expectation is always taken over contexts and rewards drawn according to the system descriptor P, so we suppress the subscript P. The goal of the agent is to find a policy π that attains the largest value.

CDPs capture classical RL models, like MDPs and POMDPs, with appropriately chosen contexts:

Example 1 (*MDPs with states as contexts*)**.** Consider a finite-horizon MDP $(\mathcal{S}, \mathcal{A}, H, \Gamma_1, \Gamma, R)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, H is the horizon, $\Gamma_1 \in \Delta(\mathcal{S})$ is the initial state distribution, $\Gamma : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the state transition function, $R : \mathcal{S} \times \mathcal{A} \to \Delta([0,1])$ is the reward function, and an episode takes the form of $(s_1, a_1, r_1, \ldots, s_H, a_H, r_H)$. We can convert the MDP to a CDP $(\mathcal{X}, \mathcal{A}, H, P)$ by letting $\mathcal{X} = \mathcal{S} \times [H]$ and $x_h = (s_h, h)$, which allows the set of policies $\{\mathcal{X} \to \mathcal{A}\}$ to contain the optimal policy (Puterman, 1994). The system descriptor is $P = (P_\varnothing, P_+)$, where $P_\varnothing(x_1) = \Gamma_1(s_1)$, and $P_+(r_h, x_{h+1} \,|\, x_1, a_1, r_1, \ldots, x_h, a_h) = R(r_h|s_h, a_h)\,\Gamma(s_{h+1}|s_h, a_h)$.

As above, the system descriptor for a model is usually obvious and we omit its specification in the following examples. Turning to POMDPs, it might seem that a CDP limits the agent's decision-making strategies to memoryless (or reactive) policies, as we only consider policies in $\{\mathcal{X} \to \mathcal{A}\}$. This is not true. We clarify this issue by showing that we can use the history as context, and the induced CDP suffers no loss in the ability to represent optimal policies.

Example 2 (*POMDPs with histories as contexts*)**.** Consider a finite-horizon POMDP with a *hidden* state space $\mathcal{S}$, an observation space $\mathcal{O}$, and an emission process D_s specifying a distribution over $\mathcal{O}$. We can convert the POMDP to a CDP $(\mathcal{X}, \mathcal{A}, H, P)$ by letting $\mathcal{X} = (\mathcal{O} \times \mathcal{A} \times \mathbb{R})^* \times \mathcal{O}$ and $x_h = (o_1, a_1, r_1, \ldots, o_h)$ is the observed history at level h.

Our next example considers a POMDP where the context can be substantially more concise than the full history. As will be formalized in Section 2.2, all we need is that the context can express a good value function, which is significantly weaker than requiring it be a sufficient statistic (unlike e.g., Hutter 2005). Therefore, it is important to separate the context in a CDP from any precise notion of state in the process, and instead keep it as a modeling choice.

Example 3 (*POMDPs with sliding windows of observations as contexts*)**.** Sometimes partial observability can be resolved by using a small history: for example, in Atari games, it is common to keep track of the last 4 images (Mnih et al., 2015). In this case, we can represent the problem as a CDP by letting $x_h = (o_{h-3}, o_{h-2}, o_{h-1}, o_h)$.

We hope the above examples demonstrate the generality and flexibility of the CDP framework. Finally, we introduce a regularity assumption on the rewards.

Assumption 1 (Boundedness of rewards)**.** *We assume that regardless of how actions are chosen, for any* $h = 1, \ldots, H$*,* $r_h \geq 0$ *and* $\sum_{h=1}^H r_h \leq 1$ *almost surely.*[4]

2.2. Value-based RL and Function Approximation

A CDP makes no assumptions on the cardinality of the context space, which makes it critical to generalize across contexts, since an agent might not observe the same context twice. Hence, we consider value-based RL with function approximation. That is, the agent is given a set of functions $\mathcal{F} \subseteq \mathcal{X} \times \mathcal{A} \to [0,1]$ and uses it to approximate an *action-value function* (or Q-function). To avoid imposing boundary-conditions, we set $f(x_{H+1}) \equiv 0$ w.l.o.g. For ease of presentation, we assume that $\mathcal{F}$ is finite with $|\mathcal{F}| = N < \infty$ throughout the paper. In Appendix A.3 we allow infinite function classes with bounded complexity.

As in typical value-based RL, the goal is to identify $f \in \mathcal{F}$ which respects a particular set of *Bellman equations* and achieves a high value with its greedy policy $\pi_f(x) = \operatorname{argmax}_{a \in \mathcal{A}} f(x, a)$. We next set up the appropriate extensions of Bellman equations to CDPs and the optimal value $V^\star_{\mathcal{F}}$ through a series of definitions. Unlike MDPs, these involve both the CDP and function approximator $\mathcal{F}$.

Definition 2 (Average Bellman error)**.** *Given a policy* $\pi : \mathcal{X} \to \mathcal{A}$ *and a function* $f : \mathcal{X} \times \mathcal{A} \to [0,1]$*, the* average Bellman error *of* f *under* π *at level* h *is defined as*

$$\mathcal{E}(f, \pi, h) = \mathbb{E}\left[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1}) \;\middle|\; a_{1:h-1} \sim \pi,\ a_{h:h+1} \sim \pi_f \right]. \tag{2}$$

[4] The bound of 1 is w.l.o.g. More generally, we may simply replace ϵ with ϵ/R in all the sample complexity results when the bound is R. See more discussion in Kakade (2003, Section 2.2.3).

The average Bellman error measures the self-consistency of f between its predictions at levels h and $h+1$, when all the previous actions are taken according to some policy π. We now define a set of Bellman equations.

Definition 3 (Bellman equations and validity of f)**.** *Given an (f, π, h) triple, a* Bellman equation *posits $\mathcal{E}(f, \pi, h) = 0$. We say $f \in \mathcal{F}$ is* valid *if the Bellman equation on $(f, \pi_{f'}, h)$ holds for every $f' \in \mathcal{F}, h \in [H]$.*

Note that the validity assumption only considers roll-ins according to the greedy policies π_f, which is the natural policy class given $\mathcal{F}$. In MDPs, each Bellman equation can be viewed as the linear combination of the standard Bellman optimality equations for $Q^\star$,[5] where the coefficients are the probabilities with which the roll-in policy π visits each state. This leads to the following consequence.

Fact 1 ($Q^\star$ is always valid)**.** *Given an MDP and a space of functions $\mathcal{F} : \mathcal{S} \times [H] \times \mathcal{A} \to [0, 1]$, if $Q^\star \in \mathcal{F}$, then in the corresponding CDP with $\mathcal{X} = \mathcal{S} \times [H]$, $Q^\star$ is valid.*

While $Q^\star$ satisfies the Bellman equations and yields the optimal policy $\pi^\star = \pi_{Q^\star}$, there can be other functions which also satisfy the equations while yielding suboptimal policies. For instance, if $f(x, \pi_f(x))$ correctly predicts the long-term reward of π_f, then f is always valid. Since validity alone does not imply that we get a good policy, it is natural to search for a valid value function which also induces a high-value policy. We formalize this goal next.

Definition 4 (Optimal value)**.** *Define*

$$f^\star = \operatorname*{argmax}_{f \in \mathcal{F}:\ f \text{ is valid}} V^{\pi_f}, \text{ and } V^\star_{\mathcal{F}} = V^{\pi_{f^\star}}.$$

Fact 2. *In the setting of Fact 1, we have $f^\star = Q^\star \in \mathcal{F}$, and $V^\star_{\mathcal{F}} = V^\star$, which is the optimal long-term value.*

Definition 4 implicitly assumes that there is at least one valid $f \in \mathcal{F}$. This is weaker than the realizability assumption made in the value-based RL literature, that $\mathcal{F}$ contains $Q^\star$ of an MDP (e.g., Krishnamurthy et al., 2016) (see Facts 1 and 2). While some works only require $Q^\star$ to be *approximately* captured (e.g., Antos et al., 2008), our algorithm can also be adapted to work with an approximate notion of validity as discussed in Appendix A.4.

3. Bellman Factorization and Bellman Rank

CDPs are general models for sequential decision making, but are there efficient RL algorithms for them?

Unfortunately, without further assumptions, learning in CDPs is generally hard, since they subsume MDPs and POMDPs with arbitrarily large state/observation spaces. Formally, the sample complexity of learning CDPs in the worst-case is $\Omega(K^H)$ when $K = |\mathcal{A}|$, even when the complexity of the function class, measured by $\log |\mathcal{F}|$, is small. The result is due to Krishnamurthy et al. (2016) and is included in Appendix F.1 for completeness.

Of course the lower bound instances are quite pathological and devoid of any structure that is often present in real problems. To capture these realistic scenarios, we propose a new complexity measure and restrict our attention to settings where this measure is low. As we will see, this measure is naturally small for many existing models, and, when it is small, efficient reinforcement learning is possible.

The complexity measure we propose is a structural characterization of the set of Bellman equations induced by the CDP and the class $\mathcal{F}$ (recall Definitions 2 and 3), that we need to check to find valid functions. Checking validity by enumeration is statistically intractable for large $\mathcal{F}$, since it requires $\Omega(|\mathcal{F}|)$ samples to perform all roll-ins. However, observe that the Bellman equations are structured in tabular MDPs: the average Bellman error under any roll-in policy is a stochastic combination of the single-state errors, and checking the single-state errors (which is tractable) is sufficient to guarantee validity. This observation hints toward a more general phenomenon: whenever the collection of Bellman errors across all roll-in policies can be concisely represented, we may be able to check the validity of all functions in a tractable way.

This intuition motivates a new complexity measure that we call the *Bellman rank*. Define the Bellman error matrices, one for each h, to be $|\mathcal{F}| \times |\mathcal{F}|$ matrices where the $(f, f')^{\text{th}}$ entry is the Bellman error $\mathcal{E}(f, \pi_{f'}, h)$. Informally, the Bellman rank for a CDP and a given value-function class $\mathcal{F}$ is a uniform upper bound on the rank of these H Bellman error matrices.

Definition 5 (Bellman factorization and Bellman rank)**.** *We say that a CDP $(\mathcal{X}, \mathcal{A}, H, P)$ and $\mathcal{F} \subset \mathcal{X} \times \mathcal{A} \to [0, 1]$ admit Bellman factorization with Bellman rank M and norm parameter ζ, if there exists $\nu_h : \mathcal{F} \to \mathbb{R}^M, \xi_h : \mathcal{F} \to \mathbb{R}^M$ for each $h \in [H]$, such that for any $f, f' \in \mathcal{F}, h \in [H]$,*

$$\mathcal{E}(f, \pi_{f'}, h) = \langle \nu_h(f'), \xi_h(f) \rangle, \tag{3}$$

and $\|\nu_h(f')\|_2 \cdot \|\xi_h(f)\|_2 \le \zeta < \infty$.

The exact factorization in Eq. (3) can be relaxed to an approximate version as is discussed in Appendix A.4. Unlike rank-based notions in PSRs (Littman et al., 2001) and multiplicity automata (Schützenberger, 1961), Bellman rank depends both on the process and the class $\mathcal{F}$. In the remainder of this section we showcase the generality of Definition 5 by describing a number of common RL settings that have a small Bellman rank. Throughout, we see how

[5] We refer the readers who are not familiar with the definition of $Q^\star$ to standard texts, such as (Sutton & Barto, 1998).

the Bellman rank captures the process-specific structures that allow for efficient exploration. Proofs of all claims in this section are deferred to Appendix B.

We start with the tabular MDP setting, and show that the Bellman rank is at most the number of states.

Proposition 1 (Bellman rank bounded by number of states in MDPs)**.** *Consider the setting of Example 1 with the corresponding CDP. With any class $\mathcal{F}$, this model admits a Bellman factorization with $M = |\mathcal{S}|$ and $\zeta = 2\sqrt{M}$.*

The MDP example is particularly simple as each coordinate of the M-dimensional space corresponds to a state, which is observable. Our next few examples show that this is not necessary, and that the Bellman factorization can be based on latent properties of the process. We next consider large MDPs whose transition dynamics have a low-rank structure. A closely related setting has been considered by Barreto et al. (2011; 2014) where the low-rank structure is exploited to speed up MDP planning, but no sample-efficient RL algorithms were previously known for this setting.

Proposition 2 (Bellman rank in low-rank MDPs, informally)**.** *Consider the setting of Example 1 with a transition matrix Γ having rank at most M. The induced CDP along with any $\mathcal{F} \subset \mathcal{X} \times \mathcal{A} \to [0, 1]$ admits a Bellman factorization with Bellman rank M.*

The next example considers POMDPs with large observations spaces and reactive value functions, where the Bellman rank is at most the number of hidden states.

Proposition 3 (Bellman rank bounded by hidden states in reactive POMDPs)**.** *Consider the setting of Example 3 with $|\mathcal{S}| < \infty$ and a sliding window of size 1. Given any $\mathcal{F} \subset \mathcal{X} \times \mathcal{A} \to [0, 1]$, this model admits a Bellman factorization with $M = |\mathcal{S}|$ and $\zeta = 2\sqrt{M}$.*

Propositions 2 and 3 can be proved under a unified model that generalizes POMDPs by allowing the transition and reward functions to depend on the observation (Figure 1). This model captures the experimental settings considered in state-of-the-art empirical RL work, where agents act in a grid-world ($|\mathcal{S}|$ is small) and receives complex and rich observations such as raw pixel images ($|\mathcal{O}|$ is large); see e.g., Johnson et al. (2016). The model also subsumes and generalizes the setting of Krishnamurthy et al. (2016) which requires deterministic transitions in the underlying MDP.

Next, we consider Predictive State Representations (PSRs), which are models of partially observable systems with parameters grounded in observable quantities (Littman et al., 2001). Similar to the case of POMDPs, we can bound the Bellman rank in terms of the rank of the PSR[6] when the candidate value functions are reactive.

[6]Every POMDP has an equivalent PSR whose rank is bounded by the number of hidden states (Singh et al., 2004).

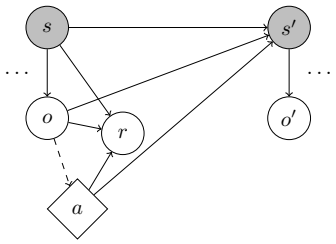

Figure 1. A unified model that subsumes MDPs and reactive POMDPs and has a low Bellman rank. Gray nodes represent unobservable quantities while diamonds are actions controlled by the agent. The dashed arrow indicates that the action is a function only of the current observation, so value functions are reactive.

Proposition 4 (Bellman rank in PSRs, informally)**.** *Consider a partially observable system with observation space $\mathcal{O}$ and the induced CDP $(\mathcal{X}, \mathcal{A}, H, P)$ with $x_h = (o_h, h)$. If the linear dimension of the system (i.e., rank of its PSR model) is at most L, then given any $\mathcal{F} : \mathcal{X} \times \mathcal{A} \to [0, 1]$, the Bellman rank is bounded by LK.*

The last example considers a class of linear control problems called Linear Quadratic Regulators (LQRs). We show that the Bellman rank in LQRs is bounded by the dimension of the state space. Unlike previous examples, here we crucially use structure of the quadratic value functions, which is the form $Q^\star$ takes. Exploration in this class of problems has been previously considered by Osband & Van Roy (2014). Note that the algorithm to be introduced in the next section does not directly apply to LQRs due to the continuous action space, and adaptations that exploit the structure of the action space may be needed.

Proposition 5 (Bellman rank in LQRs, informally)**.** *An LQR can be viewed as an MDP with continuous state space $\mathbb{R}^d$ and action space $\mathbb{R}^K$, where the dynamics are described by some linear equations. Given the function class $\mathcal{F}$ which consists of non-stationary quadratic functions of the state, the Bellman rank is bounded by $d^2 + 1$.*

4. Algorithm and Main Results

In this section we present our algorithm for learning CDPs that have a Bellman factorization with small Bellman rank, along with the main sample complexity guarantee. To aid presentation and help convey the main ideas, we make three simplifying assumptions. We assume that (1) the agent knows the Bellman rank M and the corresponding norm bound, (2) the function class $\mathcal{F}$ is finite with cardinality N, and (3) the validity and Bellman factorization conditions (Definitions 3 and 5) hold exactly. We relax these assumptions in Section 5 and Appendix A.

We are interested in designing an algorithm for *PAC Learning CDPs*. We say that an algorithm PAC learns a

CDP if given $\mathcal{F}$, two parameters $\epsilon, \delta \in (0,1)$, and access to the CDP, the algorithm outputs a policy $\hat{\pi}$ with $V^{\hat{\pi}} \geq V^{\star}_{\mathcal{F}} - \epsilon$ with probability at least $1-\delta$. The sample complexity is the number of episodes needed to achieve such a guarantee, and is typically expressed in terms of ϵ, δ, and other relevant parameters. The goal is to design an algorithm with sample complexity that is $\text{Poly}(M, K, H, 1/\epsilon, \log(N), \log(1/\delta))$ where M is the Bellman rank, K is the number of actions, and H is the time horizon. Importantly, the bound allows no dependence on the number of unique contexts $|\mathcal{X}|$.

4.1. Algorithm

Pseudocode for our algorithm, OLIVE (Optimism Led Iterative Value-function Elimination), is displayed in Algorithm 1. Theorem 1 describes how to set the parameters $n_{\text{est}}, n_{\text{eval}}, n$, and ϕ. For brevity, we introduce a shorthand for empirical Bellman errors given a tuple (x, a, r, x'):

$$\sigma(f, x, a, r, x') := f(x, a) - r - f(x', \pi_f(x')). \tag{4}$$

At a high level, the algorithm aims to eliminate functions $f \in \mathcal{F}$ that fail to satisfy the validity condition in Definition 3. This is done by Lines 13 and 14 inside the loop of the algorithm. Line 13 uses importance weighting to get an unbiased estimate of $\mathcal{E}(f, \pi_t, h_t)$, the average Bellman error for function f on roll-in policy π_t at time h_t. Thus, Line 14 eliminates functions that have high average Bellman error under π_t and hence are not valid.

The other major component of the algorithm involves choosing the roll-in policy π_t and level h_t on which to do the learning step. At iteration t, we choose the roll-in policy π_t *optimistically*, by choosing f_t that predicts the highest value at the starting context distribution and setting $\pi_t = \pi_{f_t}$. To pick h_t, we compute f_t's average Bellman error on its own roll-in distribution (Line 7), and set h_t to be any level for which this average Bellman error is high (See Line 11). As we will show, these choices ensure that substantial learning happens on each iteration, guaranteeing that the algorithm uses polynomially many episodes.

The last component is the termination criterion. The algorithm terminates if f_t has small average Bellman error on its own roll-in distribution at all levels. This criteria guarantees that π_t is near optimal.

Computationally, the algorithm requires enumeration of the value-function class, which we expect to be extremely large or infinite in practice. A computationally efficient implementation is essential for a practical algorithm, which remains an open question. We focus on the sample efficiency of the algorithm in this paper.

Intuition for OLIVE. To convey intuition, it is helpful to ignore any sampling effects by replacing all empirical estimates with population values and set ϵ to 0. The first important fact is that the algorithm never eliminates a valid function, since the learning step in Line 14 only eliminates a function f if we can find a distribution on which it has a large average Bellman error. If f is valid, then $\mathcal{E}(f, \pi, h) = 0$ for all π, h, so f is never eliminated.

The second fact is that if a function f is valid, then its predicted value is exactly the value achieved by the greedy policy π_f, that is $V_f = \mathbb{E}[f(x_1, \pi_f(x_1))] = V^{\pi_f}$. This is based on the following lemma.

Lemma 1 (Value-function error decomposition)**.** *Define* $V_f = \mathbb{E}[f(x_1, \pi_f(x_1))]$. *Then* $\forall f : \mathcal{X} \times \mathcal{A} \to [0,1]$,

$$V_f - V^{\pi_f} = \sum_{h=1}^{H} \mathcal{E}(f, \pi_f, h). \tag{5}$$

Therefore, since f_t is chosen *optimistically* as the maximizer of the value prediction among the surviving functions, and since we never eliminate valid functions, if OLIVE terminates, it must output a policy with value $V^{\star}_{\mathcal{F}}$. In the analysis, we incorporate sampling effects to derive robust versions of these facts so the algorithm always outputs a policy that is at most ϵ-suboptimal.

The more challenging component is bounding the number of iterations of the algorithm, which is critical for obtaining a polynomial sample complexity bound. This argument crucially relies on the Bellman factorization (Definition 5), which enables us to embed the distributions over contexts for any roll-in policy into M dimensions and measure progress in this low-dimensional space.

For now, fix some h and focus on the iterations where $h_t = h$. If we ignore sampling effects we can set $\phi = 0$. By using the Bellman factorization to write $\mathcal{E}(f, \pi_{f_t}, h)$ as an inner product, we can think of the learning step in Line 14 as introducing a homogeneous linear constraint on the set of $\xi_h(f)$ vectors: $\langle \nu_h(f_t), \xi_h(f) \rangle = 0$. Now, if we execute the learning step at h again in a later iteration t', we have $\langle \nu_h(f_{t'}), \xi_h(f_{t'}) \rangle \neq 0$ from Line 11. Importantly, this means that $\nu_h(f_{t'})$ must be linearly independent from previous $\nu_h(f_t)$ since $\langle \nu_h(f_t), \xi_h(f_{t'}) \rangle = 0$. Since every time $h_t = h$, the number of linearly independent constraints increases by 1, the number of iterations where $h_t = h$ is at most M, the dimension of the space. Thus the Bellman rank (times H) upper-bounds the number of iterations.

The above heuristic reasoning, despite relying on the brittle notion of linear independence, can be made robust. With sampling effects, rather than homogeneous linear equalities, the learning step for level h introduces linear inequality constraints to the $\xi_h(f)$ vectors. But if f' is a surviving function that forces us to train at h, it means that $\langle \nu_h(f'), \xi_h(f') \rangle$ is very large, while $\langle \nu_h(\cdot), \xi_h(f') \rangle$ is very small for all previous $\nu_h(\cdot)$ vectors used in the learning

Algorithm 1 OLIVE $(\mathcal{F}, M, \zeta, \epsilon, \delta)$ – **O**ptimism **L**ed **I**terative **V**alue-function **E**limination

Collect n_{est} trajectories with actions taken in an arbitrary manner; save initial contexts $\{x_1^{(i)}\}_{i=1}^{n_{\text{est}}}$.
Estimate the predicted value for each $f \in \mathcal{F}$: $\hat{V}_f = \frac{1}{n_{\text{est}}} \sum_{i=1}^{n_{\text{est}}} f(x_1^{(i)}, \pi_f(x_1^{(i)}))$.
$\mathcal{F}_0 \leftarrow \mathcal{F}$.
for $t = 1, 2, \ldots$ **do**
  Choose policy $f_t = \operatorname{argmax}_{f \in \mathcal{F}_{t-1}} \hat{V}_f$, $\pi_t = \pi_{f_t}$.
  Collect n_{eval} trajectories by following π_t (i.e. $a_h^{(i)} = \pi_t(x_h^{(i)})$ for all h and $i = 1, \ldots, n_{\text{eval}}$).
  Estimate $\forall h \in [H],\ \tilde{\mathcal{E}}(f_t, \pi_t, h) := \frac{1}{n_{\text{eval}}} \sum_{i=1}^{n_{\text{eval}}} \sigma(f, x_h^{(i)}, a_h^{(i)}, r_h^{(i)}, x_{h+1}^{(i)})$.
  if $\sum_{h=1}^{H} \tilde{\mathcal{E}}(f_t, \pi_t, h) \leq 5\epsilon/8$ **then**
    Terminate and output π_t.
  end if
  Pick $h_t \in [H]$ such that $\tilde{\mathcal{E}}(f_t, \pi_t, h_t) \geq 5\epsilon/(8H)$.
  Collect n trajectories where $a_h^{(i)} = \pi_t(x_h^{(i)})$ for all $h \neq h_t$ and $a_{h_t}^{(i)}$ is drawn uniformly at random.
  Estimate $\forall f \in \mathcal{F},\ \widehat{\mathcal{E}}(f, \pi_t, h_t) := \frac{1}{n} \sum_{i=1}^{n} \frac{\mathbf{1}[a_{h_t}^{(i)} = \pi_f(x_{h_t}^{(i)})]}{1/K} \sigma(f, x_{h_t}^{(i)}, a_{h_t}^{(i)}, r_{h_t}^{(i)}, x_{h_t+1}^{(i)})$. (see Eq. (4))
  Learn $\mathcal{F}_t = \left\{ f \in \mathcal{F}_{t-1} : \left| \widehat{\mathcal{E}}(f, \pi_t, h_t) \right| \leq \phi \right\}$.
end for

step. Intuitively this means that the new $\nu_h(f')$ vector is quite different from all of the previous ones. Our proof uses a volumetric argument to show that this suffices to guarantee substantial learning takes place. In more detail, we track the volume of an enclosing ellipsoid of the surviving $\xi_h(f)$ functions and show that each time we learn at level h this volume shrinks multiplicatively, which results in an iteration complexity that is linear in MH.

The optimistic choice for f_t is critical for driving the agent's exploration. With this choice, if f_t is valid, then the algorithm terminates correctly, and if f_t is not valid, then substantial progress is made. Thus the agent does not get stuck exploring with many valid but suboptimal functions, which could result in exponential sample complexity.

4.2. Sample Complexity

We now turn to the main result, which guarantees that OLIVE PAC-learns Contextual Decision Processes with polynomial sample complexity.

Theorem 1. *For any $\epsilon, \delta \in (0, 1)$, any CDP and function class $\mathcal{F}$ that admit a Bellman factorization with parameters M and ζ, run* OLIVE *with the following parameters:*

$$\phi = \frac{\epsilon}{12H\sqrt{M}}, \qquad n_{est} = \frac{32}{\epsilon^2} \log(6N/\delta),$$

$$n_{eval} = \frac{288H^2}{\epsilon^2} \log\left(\frac{12H^2 M \log(6H\sqrt{M}\zeta/\epsilon)}{\delta} \right),$$

$$n = \frac{4608H^2 MK}{\epsilon^2} \log\left(\frac{12NHM \log(6H\sqrt{M}\zeta/\epsilon)}{\delta} \right).$$

Then, with probability at least $1-\delta$, OLIVE *returns a policy $\hat{\pi}$ that satisfies $V^{\hat{\pi}} \geq V^\star_{\mathcal{F}} - \epsilon$ (recall Definition 3 for $V^\star_{\mathcal{F}}$), and the number of episodes required is at most*[7]

$$\tilde{\mathcal{O}}\left(\frac{M^2 H^3 K}{\epsilon^2} \log(N\zeta/\delta) \right). \tag{6}$$

Thus, if a CDP and function class $\mathcal{F}$ admit a Bellman factorization with small Bellman rank and $\mathcal{F}$ contains valid functions, OLIVE is guaranteed to find a near optimal valid function using only polynomially many episodes. To our knowledge, this is the most general polynomial sample complexity bound for RL with rich observations and function approximation, as many popular models are shown to admit small Bellman rank (see Section 3, Table 1). The result also certifies that the notion of Bellman factorization, which is quite general, is sufficient for efficient exploration and learning in sequential decision making problems.

It is worth briefly comparing this result with prior work.

1. The most closely related result is the recent work of Krishnamurthy et al. (2016), who also consider episodic RL with infinite observation spaces and function approximation. The model studied there is a CDP with Bellman rank M, so our result applies as is to that setting. Importantly, we eliminate the need for deterministic transitions in that work, while improving the dependence on H and ϵ, although with worse scaling in M. We emphasize that our result applies to a much more general class of models.
2. Several works provide sample complexity bounds for fitted value/policy iteration methods (e.g., Munos

[7] We use $\tilde{\mathcal{O}}(\cdot)$ notation to suppress poly-logarithmic dependence on everything except N and δ.

(2003); Antos et al. (2008); Munos & Szepesvári (2008)). While these results are relevant, they do not address the exploration issue, which is our main focus, and circumvent it by impliciting assuming an exploratory policy for data collection.

3. Ng & Jordan (2000) proposed a policy search method for POMDPs called PEGASUS, with a sample complexity that scales polynomially with the statistical complexity of the policy class and the horizon. Despite the powerful result, the algorithm requires careful control over the random numbers that determine the state transitions. While the assumption can hold for certain simulated environments, the scope of applications is relatively limited.
4. Since CDPs include small-state MDPs (Kearns & Singh, 2002; Brafman & Tennenholtz, 2003; Strehl et al., 2006), the algorithm can be applied as is to these problems. Unfortunately, our sample complexity is polynomially worse than the state of the art $\tilde{\mathcal{O}}(\frac{M\text{poly}(H)K}{\epsilon^2}\log(1/\delta))$ bounds for PAC-learning MDPs (Dann & Brunskill, 2015). On the other hand, the algorithm also applies to MDPs with infinite state spaces with Bellman factorizations, which cannot be handled by tabular approaches.
5. Finally, Contextual Decision Processes also encompass contextual bandits, where the sample complexity is $\Theta(K\log(N)/\epsilon^2)$ (Agarwal et al., 2014). As contextual bandits have $M = H = 1$, OLIVE achieves the optimal sample complexity in this case.

Turning briefly to lower bounds, since the CDP setting with Bellman factorization is new, general lower bounds for the broad class do not exist. However, we can use MDP lower bounds for guidance on the question of optimality, since the small-state MDPs in Example 1 are a special case. While no existing MDP lower bounds apply as is (because formulations vary), in Appendix F.2 we adapt ideas from Auer et al. (2002) to obtain a $\Omega(MKH/\epsilon^2)$ sample complexity lower bound for learning the MDPs in Example 1.

In comparison, the sample complexity in Theorem 1 is worse in M, H, and $\log(N)$ factors, but of course the small-state MDP is a significantly simpler special case. We leave as future work the question of optimal sample complexity for learning CDPs with low Bellman rank.

5. Extensions

The basic result presented here is quite robust and admits many extensions, some of which we briefly describe here; the details are deferred to Appendix A.

1. Handling infinite function classes with dependence on VC-dimension like quantities. This result uses a context-value function class $\mathcal{G} \subset \mathcal{X} \to [0,1]$ and a policy class $\Pi \subset \mathcal{X} \to \mathcal{A}$ instead of a context-action value class as in OLIVE, with sample complexity depending on the pseudo-dimension of $\mathcal{G}$ and the Natarajan dimension of Π. These are standard measures for regression and multi-class classification, and several natural classes have known bounds.
2. Competing with approximately valid value-functions with inexact Bellman factorization. For this result, we extend the definition of validity and $V^\star_{\mathcal{F}}$ (Defs. 3 and 4) to allow small but non-zero Bellman errors, and also only require that the Bellman error matrices have a low rank approximation with small ℓ_∞ error.
3. Adapting to unknown Bellman rank. Here we run OLIVE with choices of M growing at a doubling schedule and show that the PAC-guarantee is preserved without loss in sample complexity.

6. Discussion

In this paper, we presented a new model for RL with rich observations, called Contextual Decision Processes, and a structural property, the Bellman factorization, of these models that enables sample-efficient learning. The unified approach allows us to address several settings of practical interest that have largely eluded RL theory to date. Our work also elicits several further questions:

1. Can we obtain a computationally efficient algorithm for some form of this setting? Prior related work (for instance in contextual bandits (Dudik et al., 2011; Agarwal et al., 2014)) used supervised learning oracles for computationally efficient approaches. Is there a suitable oracle for this setting?
2. The sample complexity depends polynomially on the cardinality of the action space. Can we extend the results to handle large or continuous action spaces (e.g., by incorporating concepts such as Eluder dimension (Russo & Van Roy, 2013))?
3. Can we address sample-efficient RL given only a policy class rather than a value function class? Empirical approaches often rely on policy gradients, which are subject to local optima. Are there parallel results to this work, without access to value functions?

Resolutions to these questions are important for further connecting RL theory with practice.

Acknowledgements

Part of this work was completed while NJ and AK were at Microsoft Research. NJ was partially supported by Rackham Predoctoral Fellowship in University of Michigan.

References

Agarwal, Alekh, Hsu, Daniel, Kale, Satyen, Langford, John, Li, Lihong, and Schapire, Robert E. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, 2014.

Anderson, Brian D.O. and Moore, John B. *Optimal Control: Linear Quadratic Methods*. Courier Corporation, 2007.

Antos, András, Szepesvári, Csaba, and Munos, Rémi. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 2008.

Auer, Peter, Cesa-Bianchi, Nicolo, Freund, Yoav, and Schapire, Robert E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 2002.

Azizzadenesheli, Kamyar, Lazaric, Alessandro, and Anandkumar, Animashree. Reinforcement learning of POMDPs using spectral methods. *Conference on Learning Theory*, 2016.

Barreto, André da Motta Salles, Pineau, Joelle, and Precup, Doina. Policy iteration based on stochastic factorization. *Journal of Artificial Intelligence Research*, 2014.

Barreto, Andre S, Precup, Doina, and Pineau, Joelle. Reinforcement learning using kernel-based stochastic factorization. In *Advances in Neural Information Processing Systems*, 2011.

Bellemare, Marc G., Srinivasan, Sriram, Ostrovski, Georg, Schaul, Tom, Saxton, David, and Munos, Remi. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2016.

Ben-David, Shai, Cesa-Bianchi, Nicolo, and Long, Philip M. Characterizations of learnability for classes of $\{0,\ldots, n\}$-valued functions. In *Conference on Learning Theory*, 1992.

Bland, Robert G, Goldfarb, Donald, and Todd, Michael J. The ellipsoid method: A survey. *Operations research*, 1981.

Boots, Byron, Siddiqi, Sajid M., and Gordon, Geoffrey J. Closing the learning-planning loop with predictive state representations. *International Journal of Robotics Research*, 2011.

Brafman, Ronen I. and Tennenholtz, Moshe. R-max – a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 2003.

Dann, Christoph and Brunskill, Emma. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, 2015.

Devroye, Luc, Györfi, László, and Lugosi, Gábor. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

Dudik, Miroslav, Hsu, Daniel, Kale, Satyen, Karampatziakis, Nikos, Langford, John, Reyzin, Lev, and Zhang, Tong. Efficient optimal learning for contextual bandits. In *Uncertainty in Artificial Intelligence*, 2011.

Hallak, Assaf, Di Castro, Dotan, and Mannor, Shie. Contextual Markov decision processes. *arXiv:1502.02259*, 2015.

Haussler, David. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and computation*, 1992.

Haussler, David. Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 1995.

Haussler, David and Long, Philip M. A generalization of Sauer's lemma. *Journal of Combinatorial Theory, Series A*, 1995.

Hutter, Marcus. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, 2005.

Johnson, Matthew, Hofmann, Katja, Hutton, Tim, and Bignell, David. The Malmo Platform for artificial intelligence experimentation. In *International Joint Conference on Artificial Intelligence*, 2016.

Jong, Nicholas K. and Stone, Peter. Model-based exploration in continuous state spaces. In *Abstraction, Reformulation, and Approximation*, 2007.

Kakade, Sham. *On the sample complexity of reinforcement learning*. PhD thesis, University College London, 2003.

Kakade, Sham, Kearns, Michael, and Langford, John. Exploration in metric state spaces. In *International Conference on Machine Learning*, 2003.

Kearns, Michael and Singh, Satinder. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 2002.

Kearns, Michael, Mansour, Yishay, and Ng, Andrew Y. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 2002.

Kocsis, Levente and Szepesvári, Csaba. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning*, 2006.

Krishnamurthy, Akshay, Agarwal, Alekh, and Langford, John. PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, 2016.

Langford, John and Zhang, Tong. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, 2008.

Li, Lihong. *A unifying framework for computational reinforcement learning theory*. PhD thesis, Rutgers, The State University of New Jersey, 2009.

Littman, Michael L., Sutton, Richard S., and Singh, Satinder. Predictive representations of state. In *Advances in Neural Information Processing Systems*, 2001.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, and Hassabis, Demis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Munos, Rémi. Error bounds for approximate policy iteration. In *International Conference on Machine Learning*, 2003.

Munos, Rémi and Szepesvári, Csaba. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 2008.

Natarajan, Balas K. On learning sets and functions. *Machine Learning*, 1989.

Ng, Andrew Y and Jordan, Michael. Pegasus: A policy search method for large MDPs and POMDPs. In *Uncertainty in Artificial Intelligence*, 2000.

Osband, Ian and Van Roy, Benjamin. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, 2014.

Panchenko, Dmitriy. Some extensions of an inequality of Vapnik and Chervonenkis. *Electronic Communications in Probability*, 2002.

Pazis, Jason and Parr, Ronald. Efficient PAC-optimal exploration in concurrent, continuous state MDPs with delayed updates. In *Conference on Artificial Intelligence*, 2016.

Pollard, David. *Convergence of Stochastic Processes*. Springer Science & Business Media, 2012.

Puterman, Martin. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

Russo, Dan and Van Roy, Benjamin. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, 2013.

Schützenberger, M.P. On the definition of a family of automata. *Information and Control*, 1961.

Silver, David, Huang, Aja, Maddison, Chris J., Guez, Arther, Sifre, Laurent, van den Driessche, George, Schrittwieser, Julian, Antonoglou, Ioannis, Penneershelvam, Veda, Lanctot, Marc, Dieleman, Sander, Grewe, Dominik, Nham, John, Kalchbrenner, Nal, Sutskever, Ilya, Lillicrap, Timothy, Leach, Madeleine, Kavukcuoglu, Koray, Graepel, Thore, and Hassabis, Demis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016.

Singh, Satinder and Yee, Richard C. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 1994.

Singh, Satinder, James, Michael R., and Rudary, Matthew R. Predictive state representations: A new theory for modeling dynamical systems. In *Uncertainty in Artificial Intelligence*, 2004.

Strehl, Alexander L., Li, Lihong, Wiewiora, Eric, Langford, John, and Littman, Michael L. PAC model-free reinforcement learning. In *International Conference on Machine Learning*, 2006.

Sutton, Richard S and Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Todd, Michael J. On minimum volume ellipsoids containing part of a given ellipsoid. *Mathematics of Operations Research*, 1982.

Todd, Michael J and Yıldırım, E Alper. On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics*, 2007.

Wang, Ziyu, de Freitas, Nando, and Lanctot, Marc. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, 2015.

Wen, Zheng and Van Roy, Benjamin. Efficient exploration and value function generalization in deterministic systems. In *Advances in Neural Information Processing Systems*, 2013.