The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

On Value Function Representation of Long Horizon Problems

Lucas Lehnert,[1,2] **Romain Laroche,**[1] **Harm van Seijen**[1]
lucas_lehnert@brown.edu, {romain.laroche, harm.vanseijen}@microsoft.com
[1]Microsoft Maluuba, Montreal, QC, Canada
[2]Brown University, Providence, RI, United States

Abstract

In Reinforcement Learning, an intelligent agent has to make a sequence of decisions to accomplish a goal. If this sequence is long, then the agent has to plan over a long horizon. While learning the optimal policy and its value function is a well studied problem in Reinforcement Learning, this paper focuses on the structure of the optimal value function and how hard it is to represent the optimal value function. We show that the generalized Rademacher complexity of the hypothesis space of all optimal value functions is dependent on the planning horizon and independent of the state and action space size. Further, we present bounds on the action-gaps of action value functions and show that they can collapse if a long planning horizon is used. The theoretical results are verified empirically on randomly generated MDPs and on a grid-world fruit collection task using deep value function approximation. Our theoretical results highlight a connection between value function approximation and the Options framework and suggest that value functions should be decomposed along bottlenecks of the MDP's transition dynamics.

1 Introduction

Reinforcement Learning (RL) (Kaelbling, Littman, and Moore 1996; Sutton and Barto 1998) studies how to compute an optimal control strategy for interacting with an environment. This interaction consists of the agent making a decision by choosing an action, observing a reward for selecting these actions, and observing a change in the environment's state. The goal of the agent is to find an action-selection strategy, also called *policy*, which maximizes the overall received rewards.

Typically, an optimal policy is learned by incrementally improving an intermediate policy. To improve a policy, the agent has to estimate its utility, which is expressed by a value function mapping a state to the return of a specific policy. For example, recent algorithms such as DQN (Mnih et al. 2015) learn how to play Atari 2600 games (Bellemare et al. 2013) by using a deep neural network architecture to approximate the value function.

In this paper we consider *Long Horizon Problems* (LHPs) where the agent is required to plan over many time steps into the future. While recent work focused on the benefits of using a shorter than specified planning horizon (Jiang et al. 2015), we focus on the case where using a shorter planning horizon is not possible because the policy's performance would degrade too much. In this case, we show that representing the optimal value function can become challenging, even if the agent was given the optimal state and action values. Previous work focuses on mistake or sample complexity bounds and characterizes how well state and action values can be approximated given a certain amount of data (Strehl et al. 2006; Strehl, Li, and Littman 2009). In contrast to these results, this paper focuses on the structure of the *solution* and considers how difficult representing the optimal value function is.

We present two results. The first result characterizes the value function space given a fixed finite state and action space. We express the complexity of this solution space using the generalized Rademacher complexity (Balcan 2011; Shalev-Shwartz and Ben-David 2014), which is defined on subsets of $\mathbb{R}^n$. This complexity measure of the optimal value function space increases with the planning horizon, and is independent of the state and action space size. Our result suggests that large state spaces only make learning more difficult and do not have an effect on the complexity of the solution space the agent has to search over. Subsequently we present bounds on the action-gaps of the value function. If action-gaps are large, value function approximation becomes easier (Bellemare et al. 2016b). However, if action-gaps collapse, then function approximation methods may not be able to recover the optimal action, because it lacks the necessary "resolution" to distinguish the optimal action from sub-optimal actions. We show that action-gaps can collapse if the planning horizon is long. Further, our analysis suggests that the state space should be partitioned along Bottlenecks (Bacon 2013; Stolle and Precup 2002), and for each partition a separate shorter planning horizon should be used. To support our results we present two sets of experiments. The first set is on randomly generated MDPs showing that action-gaps can collapse depending on the transition structure of the MDP. The second set shows a fruit collection task and compares approximations, made with neural networks, of the optimal value function for different planning horizons. We find that predicting long planning horizons is more challenging, despite the fact that ground truth state values were used for training.

Our results motivate the importance of studying different value function representations in RL and we hope to provide guidance for the design of new algorithms. Even if the learning problem in RL was solved, our results show that representing the optimal value function for a long planning horizon can be challenging. We present theoretical evidence that frameworks such as Options (Sutton, Precup, and Singh 1999) or a Hierarchical decomposition (Dietterich 2000) are beneficial when used with value function approximation.

2 Background

In RL the interaction between the agent and its environment is formalized as a Markov Decision Process (MDP) $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. State transitions follow a transition function p, where $p(s, a, s') = \mathbb{P}\{s'|s, a\}$, the probability of reaching state s' given that action a was selected at state s. Rewards are specified by the expected reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and the discount factor $\gamma \in [0, 1)$ favours immediate rewards over long-term rewards. The value of a policy $\pi : \mathcal{S} \to \mathcal{A}$ is defined by the value function

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \middle| s_1 = s \right], \qquad (1)$$

where the expectation is over all infinite length trajectories that start at state s and select actions according to π. While policies are typically evaluated for infinite-length trajectories, the discount factor $\gamma \in [0, 1)$ is understood as a form of finite horizon because short term rewards are weighted exponentially stronger than long term rewards.

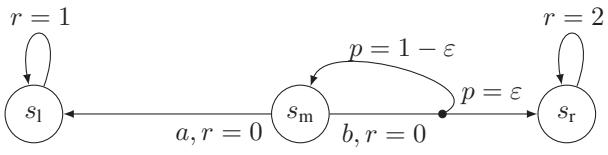
The action-value function, also called Q-function, evaluates choosing a particular action at a given state and then using the policy π afterwards:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \middle| s_1 = s, a_1 = a \right]. \qquad (2)$$

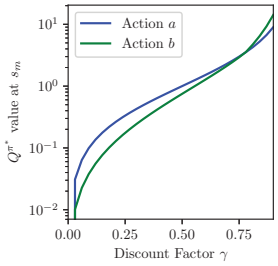
The only difference between the action-value function Q^π and the value function V^π is that the expectation in Eq. (2) considers trajectories that start with a particular action, while Eq. (1) considers trajectories where actions are chosen according to π.

Typically the value function V^π is treated as the performance objective and the agent tries to recover the optimal policy π^* that maximizes V^π at every state. This objective also depends on the discount factor γ which controls how far the agent looks ahead for making a decision. If the discount factor γ is increased, then the planning horizon also increases. Hence the agent considers longer trajectories which could generate more reward (because a trajectory can transition through positive reward transitions more often). Jiang et al. (2015) formalize this intuition by considering a lower discount factor for learning only. In this case, the algorithm ignores the discount factor γ of the problem specification and instead searches for a policy optimal for a discount factor $\gamma_{\text{LEARN}} \leq \gamma$. However, using a lower discount factor for planning incurs a loss because

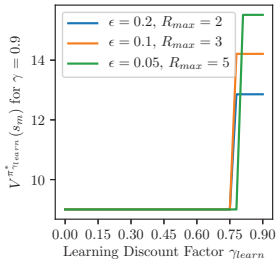
$$\forall s \in \mathcal{S},\ V_\gamma^{\pi^*_{\gamma_{\text{LEARN}}}}(s) \leq V_\gamma^{\pi^*_\gamma}(s), \qquad (3)$$



(a) Three State LHP MDP. The arrows indicate the possible transitions and are labelled with the action they correspond to (either action a or b), the transition probability p, and the reward r. If transitions are deterministic, or if both actions trigger the same transition, the transition probability of action label is omitted. For low values of ε the optimal action at the middle state s_m is b, but if ε is high it becomes unlikely to transition to s_r and choosing action a becomes optimal.



(b) Q-values of the optimal policy for the middle state s_m. The transition probabilities were set using $\varepsilon = 0.1$. One can see how the discount factor influences the optimal policy.



(c) Value loss due to using a shorter horizon for training. The plot shows $V^{\pi^*_{\text{LEARN}}}(s_\text{m})$ for different learning discount factors γ_{LEARN}.

Figure 1: Three State LHP Example.

where values V_γ^π are computed using the discount factor γ specified by the problem. In the remainder of this paper we omit the discount factor subscript if only one discount factor γ is used.

Using a learning objective different than the performance objective to make a certain problem more tractable is common practice in machine learning and RL. Often this is viewed as a form of regularization (e.g. Jiang et al. (2015)). Decomposing the reward function and learning each component independently (van Seijen et al. 2017; Laroche et al. 2017) is another approach to using a more tractable learning objective. Intrinsic motivation (Barto, Singh, and Chentanez 2004) also uses a different learning objective by augmenting the reward function to encourage a certain behaviour, for example efficient exploration (Strehl and Littman 2008; Bellemare et al. 2016a).

3 Long Horizon Problems

For some control problems, a very long planning horizon is needed to find a well performing policy. Figure 1 shows an MDP where the discount factor γ has a direct effect on the performance of the optimal policy. In this three state MDP a reward of 2 can be received by selecting action b at the middle state s_m. However, if ε is low, then with high probability action b will transition back to state s_m. To obtain

the reward of 2, the agent has to repeatedly select action b and incur many zero-reward steps before state s_r is reached. Alternatively, action a could be selected to receive a reward of 1 within two steps. This trade-off between receiving a lower reward quickly versus receiving a higher reward within many time steps can be controlled by the discount factor. Figure 1 shows how changing the discount factor can also change the optimal policy of the MDP and can cause a sudden drop in the value of a policy: if a policy $\pi^*_{\gamma_{\text{LEARN}}}$ is optimal for a lower discount factor γ_{LEARN} but evaluated using a discount factor $\gamma > \gamma_{\text{LEARN}}$, then a significantly lower value can be observed, depending on γ_{LEARN}. Figure 1(b) shows that values of the two actions at state s_m are not far apart, which highlights the importance of action-gaps. If action-gaps are small, then an approximation of the Q-values may not be accurate enough to determine the highest value action and result in a sub-optimal policy of significantly lower value, as shown in Figure 1(c). We call MDPs where learning with a smaller discount factor is not possible *Long Horizon Problems* (LHPs).

Definition 1 (Long Horizon Problem (LHP))**.** An MDP $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ is a Long Horizon Problem (LHP) if

1. the discount factor γ is close to one, and
2. a lower discount factor cannot be used for learning without incurring a significant performance loss, *i.e.* for all states s, $V^{\pi^*_{\gamma_{\text{LEARN}}}}_{\gamma}(s) \ll V^{\pi^*_{\gamma}}_{\gamma}(s)$.

The following sections show that using a high discount factor can make representing the optimal value function difficult. Our analysis suggests that value functions should be decomposed along bottle-necks in the MDP.

3.1 Increased Value Function Complexity

Existing literature on the sample complexity of different RL algorithms shows that learning the optimal policy and value function becomes more difficult as the discount factor γ is increased (Strehl et al. 2006; Strehl, Li, and Littman 2009; Jiang et al. 2015). In contrast to these learning bounds, this section characterizes how difficult it is to represent the optimal value function. First, we formally define the hypothesis space of all possible optimal value functions given a fixed state and action space, discount factor, and reward range, and then we use the generalized Rademacher complexity to quantify the complexity of this hypothesis space.

For finite state and action spaces the value function V^π can be expressed as a vector $\boldsymbol{v}^\pi$ of dimension $|\mathcal{S}|$, where each entry corresponds to the value of each state. Similarly, we define a reward vector $\boldsymbol{r}_a$ where each entry is equal to $r(s,a)$ in expectation. The transition function p is written as a state-to-state transition matrix $\boldsymbol{P}_a$ for action a. Using this vector notation the Q-values corresponding to action a can be expressed as

$$\boldsymbol{q}^\pi_a = \boldsymbol{r}_a + \gamma \boldsymbol{P}_a \boldsymbol{v}^\pi. \tag{4}$$

If we say that a policy π is greedy with respect to some value function vector $\boldsymbol{v}$, then we mean that π selects actions greedily using Q-values that are constructed using Eq. (4). Hence, all arguments that apply to state conditioned value functions also apply to Q-functions.

We make the following assumption.

Assumption 1 (Bounded Non-negative Rewards)**.** For any MDP $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, the reward function satisfies

$$r(s,a) \in [0, R_{\max}],\ \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{5}$$

This assumption is not restrictive because an MDP with negative rewards can always be converted to an MDP with non-negative rewards by shifting the reward function[1] (Ng, Harada, and Russell 1999; Von Neumann and Morgenstern 1945). Further, we consider $R_{\max}$ as a known fixed constant. Note that if negative rewards are present, then an MDP with terminal states can be converted into an MDP satisfying Assumption 1 by also shifting the value of the terminal states.

Definition 2 (Value Function Space)**.** For a finite state space $\mathcal{S}$, a finite action space $\mathcal{A}$, and a discount factor $\gamma \in [0,1)$,

$$\mathcal{V}^*_\gamma \overset{\text{def.}}{=} \left\{ \boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}_{\geq 0} \middle| \exists p, r :\ \boldsymbol{v} = \boldsymbol{v}^{\pi^*}_{\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle} \right\}, \tag{6}$$

where $\boldsymbol{v}^{\pi^*}_{\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle}$ is the value function vector of the optimal policy of the MDP $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$.

Typically value functions can be upper bounded with

$$V_{\max,\gamma} = \max_s \mathbb{E}_{\pi^*} \left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_1 = s \right] \leq \frac{R_{\max}}{1-\gamma}, \tag{7}$$

because at time step t the reward $r_t \leq R_{\max}$. Interestingly, given a fixed state and action space, the space of all possible state values is dense and equals exactly the space of all possible real valued vectors with entries that lie in $\left[0, \frac{R_{\max}}{1-\gamma}\right]$.

Lemma 1. For a fixed state space $\mathcal{S}$, an action space $\mathcal{A}$, discount factor $\gamma \in [0,1)$, and reward functions satisfying Assumption 1, the space of optimal value functions

$$\mathcal{V}^*_\gamma = \left\{ \boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}_{\geq 0} \middle| ||\boldsymbol{v}||_\infty \leq \frac{R_{\max}}{1-\gamma} \right\}. \tag{8}$$

The proof is listed in Appendix A. Lemma 1 highlights an interesting connection between the planning horizon and the complexity of value functions, and shows how the discount factor γ controls the hypothesis space the algorithm needs to be able to express and search over. If we pick two different discount factors $\tilde{\gamma}$ and γ such that $\tilde{\gamma} < \gamma$, then $\mathcal{V}^*_{\tilde{\gamma}} \subset \mathcal{V}^*_\gamma$. Intuitively, this can be understood as a hypothesis space that becomes more complex as the planning horizon becomes longer, because the hypothesis space of higher γ contains the hypothesis space of smaller $\tilde{\gamma}$.

To formalize this intuition, we use the generalized Rademacher complexity (Balcan 2011; Shalev-Shwartz and Ben-David 2014) defined on sub-spaces of $\mathbb{R}^n$. Let $\mathcal{X} \subseteq \mathbb{R}^n$, then the generalized Rademacher complexity is defined as

$$\mathfrak{R}(\mathcal{X}) = \frac{1}{n} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{n} \boldsymbol{\sigma}_i \boldsymbol{x}_i \right], \tag{9}$$

[1] Shifting the reward function by a constant results in a constant shift of any value function.

where $\boldsymbol{\sigma}$ is the random Rademacher vector with entries being either +1 or -1 with equal probability. We denote the ith entry of a vector $\boldsymbol{x}$ with $\boldsymbol{x}_i$. Similar to Zhang et al. (2016), we interpret the generalized Rademacher complexity as a measure of how well vectors in a set $\mathcal{X}$ can represent a random noise vector. If this complexity measure is high, then the set $\mathcal{X}$ is expressive enough to capture a random noise vector well and the entries within the vector $\boldsymbol{x}$ are only weakly correlated. Thus, the generalized Rademacher complexity also increases if elements of $\mathcal{X}$ have a high maximum norm.

For fixed finite state and action spaces, a fixed discount factor γ, and a fixed reward upper bound $R_{\max}$ (Assumption 1), an RL algorithm has to be able to represent any value function in the space $\mathcal{V}^*_\gamma$. We quantify the complexity of this hypothesis space with the following theorem.

Theorem 1 (Value Function Hypothesis Space Complexity)**.** The generalized Rademacher complexity of $\mathcal{V}^*_\gamma$ is

$$\mathfrak{R}(\mathcal{V}^*_\gamma) = \frac{R_{\max}}{2(1-\gamma)}. \tag{10}$$

The proof is listed in Appendix A. For a given $R_{\max}$ (Assumption 1), Theorem 1 shows that the Rademacher complexity of the hypothesis space $\mathcal{V}^*_\gamma$ increases with the planning horizon and tends to infinity as γ tends to one. This result shows that the complexity of the hypothesis space an algorithm has to represent increases as the planning horizon increases. Further, the generalized Rademacher complexity of $\mathcal{V}^*_\gamma$ is independent of state and action space sizes. The size of the state and action space becomes relevant if one considers the learning problem, where an algorithm receives a certain amount of training data and then has to estimate the optimal value function. In this case, generalization bounds can be derived (Jiang et al. 2015). Instead, we focus on the complexity of the hypothesis space an algorithm has to represent. Intuitively, this result shows that the state and action space size only makes learning harder, and the planning horizon as well as the reward range make both learning harder as well as representing an optimal solution.

The generalized Rademacher complexity of $\mathcal{V}^*_\gamma$ depends on the reward range because $\mathfrak{R}(\cdot)$ only measures the correlation of the different entries of a vector $\boldsymbol{v} \in \mathcal{V}^*_\gamma$. If a vector $\boldsymbol{v} \in \mathcal{V}^*_\gamma$ is re-scaled by two, for example, then the correlation between the vector entries decreases, and thus the generalized Rademacher complexity increases. Hence, if the reward range increases, then an algorithm also has to represent a much wider range of value functions. This effect is reflected by the $R_{\max}$ factor in Theorem 1.

3.2 Collapsing Action-Gaps

The planning horizon can have a significant impact on the action-gaps of the optimal policy and in some cases can cause action-gaps to collapse completely. At a state $s \in \mathcal{S}$ the *action-gap* is at most $\max_{a\in\mathcal{A}} Q^{\pi^*}(s,a) - \min_{a\in\mathcal{A}} Q^{\pi^*}(s,a)$, and the *maximal action-gap* over the state space $\mathcal{S}$ is defined as

$$\text{MAG}(\mathcal{S}) = \max_{s\in\mathcal{S}} \left[\max_{a\in\mathcal{A}} Q^{\pi^*}(s,a) - \min_{a\in\mathcal{A}} Q^{\pi^*}(s,a) \right]. \tag{11}$$

To bound the range of all action-gaps we focus on the maximal action-gap in an MDP, because depending on the MDPs structure the minimal action-gap can always be zero.

If the value function is approximated, for example with a deep neural network, then a large action-gap is important for being able to reliably choose the action of highest value and being able to reconstruct the optimal policy. If the maximal action-gap is too low, then any approximation method may not have the necessary "resolution" to distinguish optimal from sub-optimal actions. In this case, the algorithm cannot recover the optimal policy.

If the state space is fully connected then there always exists some policy π which can navigate from any arbitrary start state s to any other state s'. The maximum number of time steps needed to transition between any pairs of states is called the *stochastic diameter*. We consider a variation of the stochastic diameter that only considers any pairs of states that lie in a fully connected subset $\mathcal{S}_C \subseteq \mathcal{S}$. Note that if a state subset is not fully connected, then the diameter is not well defined, because the walking time between states need not be finite if they are not reachable.

Definition 3 (Subset Stochastic Diameter)**.** The *subset stochastic diameter* of an MDP $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ with a fully connected state subset $\mathcal{S}_C \subseteq \mathcal{S}$ is defined as

$$D_{\mathcal{S}_C} = \max_{s,\tilde{s}\in\mathcal{S}_C\times\mathcal{S}_C} \min_{\pi:\mathcal{S}\to\mathcal{A}} \mathbb{E}_\pi \left[\inf\{t \in \mathbb{N}, s_t = \tilde{s}\} | s_0 = s \right].$$

Note that $D_{\mathcal{S}}$ is equivalent to the usual definition of a stochastic diameter, and we omit the subscript in this case. Generalizing the definition of diameter to fully connected subsets allows us to consider MDPs whose entire state space is not fully connected (and contains terminal states), but there still exist fully connected subsets $\mathcal{S}_C \subset \mathcal{S}$. If an MDP has a fully connected state subset $\mathcal{S}_C$ with diameter $D_{\mathcal{S}_C}$, we can prove an upper bound on the maximal action-gap.

Lemma 2 (Action-gap)**.** Consider an MDP $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ and a fully connected subset $\mathcal{S}_C \subseteq \mathcal{S}$ with diameter $D_{\mathcal{S}_C}$. Then,

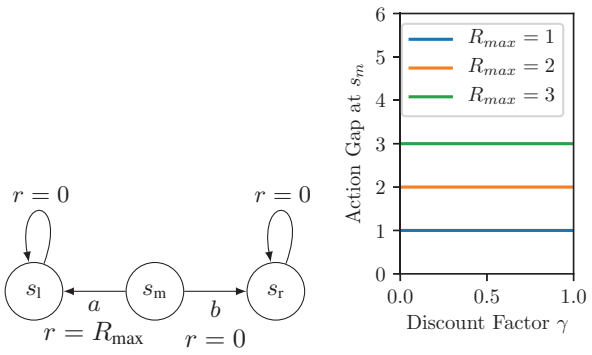
$$\text{MAG}(\mathcal{S}_C) \leq (1 - \gamma^{D_{\mathcal{S}_C}+1}) V_{\max,\gamma}, \tag{12}$$

where $V_{\max,\gamma} = \max_{s\in\mathcal{S}_C} V^{\pi^*}(s)$.

The proof is listed in Appendix B. Lemma 2 shows that the maximal action-gap depends on the diameter and the upper bound of the value function. This means that if $V_{\max,\gamma}$ remains bounded as γ tends to one, then all action-gaps in the MDP collapse. Using the value function bound Eq. (7) the maximal action-gap of a fully connected MDP with diameter D is bounded by

$$\text{MAG}(\mathcal{S}) \leq (1 - \gamma^{D+1}) \frac{R_{\max}}{1-\gamma}. \tag{13}$$

Depending on the transition dynamics of the MDP, action-gaps can become independent of the discount factor γ if the state space is not fully connected. Figure 2 shows one such example. This MDP is not fully connected and the two outer states s_l and s_r are terminal states. Because the agent can only transition out of the middle state s_m once and collect



(a) Disconnected Three State MDP.

(b) Action-gap at s_m.

Figure 2: Disconnected Three State MDP Example. In this MDP, the maximal action-gap does not depend on the discount factor γ. Instead, it depends on the reward function only. One can also come up with counter examples showing the dependency on the transition function.

the one step return once, the planning horizon becomes independent of the action-gap at the middle state s_m.

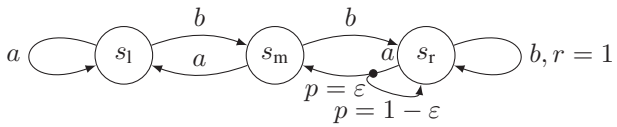
For not fully connected state spaces the maximum state value $V_{\max,\gamma}$ may remain bounded. Consider the three state chain MDP in Figure 3(a), which is fully connected if $\varepsilon > 0$. If $\varepsilon = 0$, then the subset $\mathcal{S}_C = \{s_l, s_m\} \subset \mathcal{S}$ is a fully connected component with diameter $D_{\mathcal{S}_C}$, and Lemma 2 can be applied for $\mathcal{S}_C$. However, the optimal policy may take a trajectory started in $\mathcal{S}_C$ outside the connected subset to a state in $\mathcal{S} \setminus \mathcal{S}_C$. In this case trajectories spend only a finite amount of time in $\mathcal{S}_C$ which allows us to find a much tighter bound on the value function. This bound explains the vanishing action-gap shown in Figure 3(b).

Let N be the random variable indicating the number of time steps a trajectory started in $\mathcal{S}_C$ spends in the subset $\mathcal{S}_C$. Further, assume that $\mathbb{E}[N] < \infty$[2]. Then the return generated by a trajectory started at $s \in \mathcal{S}_C$ can be split into two terms:

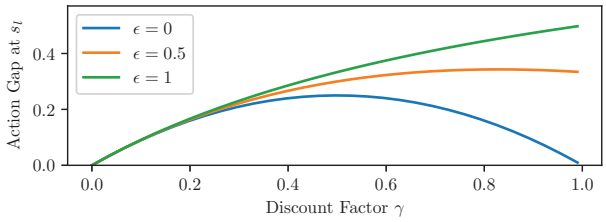
$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \middle| s_1 = s\right] \\ &= \mathbb{E}_\pi\Bigg[\sum_{t=1}^{N} \gamma^{t-1} r(s_t, a_t) \\ &\qquad + \underbrace{\sum_{t=N+1}^{\infty} \gamma^{t-1} r(s_t, a_t)}_{\leq F_\mathrm{max}} \middle| s_1 = s\Bigg]. \end{aligned} \tag{14}$$

If the return a trajectory generates outside of $\mathcal{S}_C$ is upper bounded with F_max, then a tighter upper bound on the action-gap can be derived.

[2]This assumption is not restrictive if transitions are stochastic and if the probability of leaving the fully connected subset $\mathcal{S}_C$ is greater than zero. If $\mathbb{E}[N]$ were unbounded, then one could equivalently consider a smaller MDP with a fully connected state space $\mathcal{S}_C$.



(a) Three State Chain MDP. All transitions are indicated with an arrow and labelled with their corresponding action and reward. The only stochastic transition occurs when action b is selected at state s_2, where transition probabilities are indicated with p.



(b) Action-gap of the Three State Chain MDP at state s_0 for different ε settings. In the fully connected case ($\varepsilon > 0$) the action-gap is strictly increasing, as predicted by before. For the disconnected case ($\varepsilon = 0$) the action-gap decreases for very high γ values.

Figure 3: Three State Chain MDP

Lemma 3 (Finite Trajectory Value Bound)**.** Let $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ be an MDP with a fully connected subset of the state space $\mathcal{S}_C \subset \mathcal{S}$, and let N be a random variable indicating the number of time steps a trajectory spends in $\mathcal{S}_C$ until a state outside of $\mathcal{S}_C$ is reached. If $\mathbb{E}[N] < \infty$, then

$$\max_{s \in \mathcal{S}_C} V^{\pi^*}(s) = V_{\max,\gamma} \leq \frac{1 - \gamma^{\mathbb{E}[N]}}{1 - \gamma} R_\mathrm{max} + F_\mathrm{max}. \tag{15}$$

The proof of Lemma 3 is listed in Appendix B. For a trajectory leaving a fully connected subset $\mathcal{S}_C$, the first state in $\mathcal{S} \setminus \mathcal{S}_C$ can be understood as reaching a terminal state (from the perspective of the subset $\mathcal{S}_C$). In this sense, the random variable N can be also thought of as the termination time or length of a trajectory in $\mathcal{S}_C$. Hence, a tighter value function upper bound can be found, because rewards can only be collected for a finite number of time steps in $\mathcal{S}_C$. The following theorem summarizes our results.

Theorem 2 (Action-gap Bounds)**.** Let $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ be a fully connected MDP with diameter D. Then,

$$\mathrm{MAG}(\mathcal{S}) \leq (1 - \gamma^{D+1}) \frac{R_\mathrm{max}}{1 - \gamma}. \tag{16}$$

Let $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ be an MDP with a fully connected state subset $\mathcal{S}_C \subset \mathcal{S}$. Further, assume that $\mathcal{S}_C$ has a diameter $D_{\mathcal{S}_C}$, and that for the optimal policy π^* the expected number of time steps spent in $\mathcal{S}$ is bounded, i.e. $\mathbb{E}[N] < \infty$. Then,

$$\mathrm{MAG}(\mathcal{S}_C) \leq (1 - \gamma^{D_{\mathcal{S}_C}+1}) \left(\frac{1 - \gamma^{\mathbb{E}[N]}}{1 - \gamma} R_\mathrm{max} + F_\mathrm{max} \right), \tag{17}$$

where F_max is the maximum value of the states outside $\mathcal{S}_C$ but reachable from $\mathcal{S}_C$ in one step.

Proof. The first bound restates Eq. (13). The second bound follows by first applying Lemma 2 for a subset $\mathcal{S}_C$ and then bounding the maximum value using Lemma 3. In this case we can also only upper bound the action-gaps for $\mathcal{S}_C$. □

Figure 4 plots the bounds of Theorem 2 for different parameter settings. If the optimal policy takes any trajectory out of a state sub-set $\mathcal{S}_C$, the value function across $\mathcal{S}_C$ can be bounded by a constant and we observe a reduced action-gap for high γ values. Our bound suggests a discount factor setting that allows for the largest possible action-gaps.

Vanishing action-gaps can become problematic if Q-values are only approximated, especially when using function approximation methods such as deep neural networks. Given some amount of data, suppose the function approximation method used can only capture an ϵ-close approximation to the true Q-function. If a high discount factor setting is used, then the maximal action-gap may fall below ϵ and the used algorithm cannot recover the optimal policy anymore. Depending on the connectivity of the state space, value function approximation methods can become very difficult to use if the discount factor γ is set too high.

4 Empirical Results

We conduct two sets of experiments: The first experiment verifies the dependence of the discount factor γ on the maximal action-gap on randomly sampled MDPs. The second experiment approximates the ground truth value function of a grid-world fruit collection task with a deep neural network.

4.1 Randomly Generated MDPs

In this experiment, transition functions are randomly sampled such that the state space is partitioned into four subsets $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, and $\mathcal{S}_4$, and transitions occur within the same subset with $1-\delta$ probability. We consider two cases: (1) transitioning between different components are possible (except for the terminal state), (2) only transitions to components with a higher index are possible. Further, one state in component $\mathcal{S}_4$ is terminal and transitioning into it results in a +1 reward. All other rewards are set to zero.

Figure 5(a) shows the maximal action-gap for the first three components (the reward state is in $\mathcal{S}_4$) in the partially connected case. For components $\mathcal{S}_1$ and $\mathcal{S}_2$ the maximal action-gap has a shape similarly to the predicted bound in Figure 4. For component $\mathcal{S}_3$ no optimal discount factor less than one is observed because high value states in $\mathcal{S}_4$ are reached with higher probability. If the F_{max} term from Theorem 2 is high enough, then this term has a stronger effect on the action-gap bound and can cause the maximal action-gap to be monotonically increasing. These empirical results verify our analysis. For the fully connected case, Figure 5(b) shows the same maximal action-gaps for all three components. This is due to the fact that all three components $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ can reach component $\mathcal{S}_4$ equally quickly. Interestingly, the same drop in the maximal action-gap can be observed even though the state space is fully connected (excluding the terminal state). The fact that transitions between components occur with very low probability is enough to cause the maximal action-gap to drop for high γ settings.

4.2 Value Function Approximation

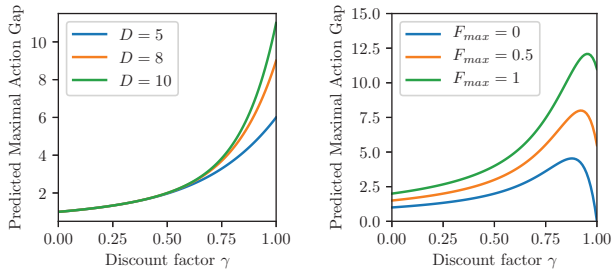
We consider a fruit collection task where the agent navigates through a 5×5 grid and receives a +1 reward when visiting a fruit cell. To support our theoretical results, a deep neural network (DNN) is fitted to the ground-truth value function $V_\gamma^{\pi^*}$ for various γ values. This learning problem is fully supervised allowing us to show how well a DNN can capture $V_\gamma^{\pi^*}$ while ignoring the problem of finding the optimal policy and estimating its value function.

Similar to the Taxi Domain (Dietterich 2000), we incorporate the location of the fruits into the state representation using a bit vector, where the first 25 entries are used for fruit positions, and the last 25 entries are used for the agent's position. The small (resp. large) DNN feeds this bit-vector as the input layer into one (resp. two) dense hidden layers with 50 (resp. 100 and then 50) units. The output is a single state-value estimate. In order to assess the value function complexity, we train for each discount factor setting a DNN of fixed size on all the 1,386,375 possible states with their ground truth values. Each DNN is trained over 500 epochs using the Adam optimizer (Kingma and Ba 2014) with default parameters. To evaluate the DNN's performance, actions are selected greedily by moving the agent up, down, left, or right to the neighbouring grid cell of highest value.

Figure 6 compares the performance of the trained DNNs with the ground truth solution. We also included the Travelling Salesman Problem (TSP) solution, which is the minimum number of steps needed to collect all fruits. The performance curves of the policies greedy with respect to the trained DNNs present a pronounced U-shape. For small γ values, the range of the value function (and thus the action-gap) collapses quickly as one moves away from fruit locations. In this case, both models cannot reach a high enough precision and hence perform worse for low γ values. Further, for very high γ values action-gaps collapse, because the Fruit Collection Task contains terminal states, which we believe also results in reduced performance of both DNNs. This effect on action-gaps aligns with our predictions presented in Section 3.2. The larger network performs better for a much wider range of γ values than the smaller network, whose performance degrades much more quickly for $\gamma \geq 0.8$. If larger DNNs have greater representational power than smaller ones, this can be explained by the fact that a larger DNN can better capture the more complex structure of a value function with a high discount factor. This pattern in Figure 6 corresponds to our complexity results presented in Section 3.1. The poor performance for low and high γ values can be explained by various other factors, such as requiring a high precision approximation for all inputs, or a difficult to optimize loss function where only poor local optima are found. Increasing the number of training epochs or adjusting the learning rate did not significantly improve performance.

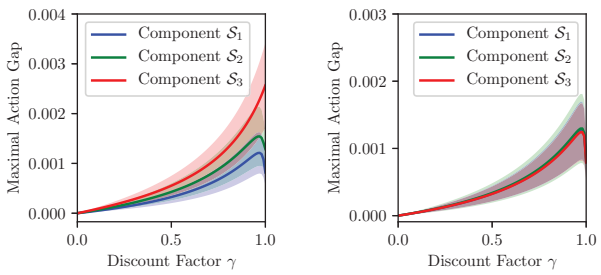
5 Discussion

This paper presents a new approach to understanding the representational complexity of value functions. Previous work analyzed the learning problem in RL and provides mistake or sample bounds when an algorithm learns through in-



(a) Bound Eq. (16) (b) Bound Eq. (17)

Figure 4: Plot of the Maximal action-gap bounds presented in Theorem 2. For both plots $R_{\max} = 1$. The right plot uses a subset diameter of 10 and $\mathbb{E}[N] = 10$.



(a) Partially Connected (b) Fully Connected

Figure 5: Four Component MDP. The MDPs have four fully connected state components of size 5 each, and two actions. The plots show averages over 100 MDPs, using $\delta = 0.01$.

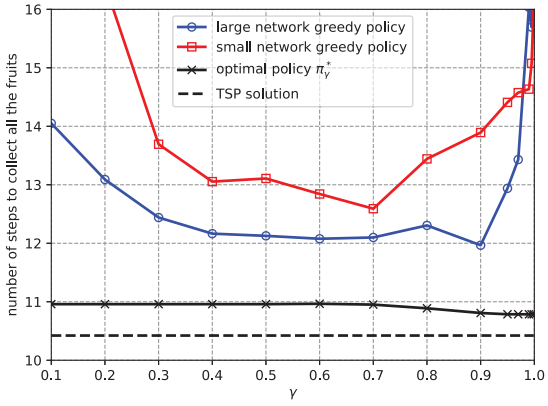


Figure 6: Value function approximation performance for different γ values. The optimal policy solution is greedy with respect to the ground-truth discounted return, which only improves for $\gamma \geq 0.7$.

teractions with its environment (Strehl et al. 2006; Strehl, Li, and Littman 2009; Jiang et al. 2015). While these bounds all depend on the discount factor and show that learning becomes harder for longer planning horizons, we show that representing the optimal solution also becomes more difficult for longer planning horizons.

Current algorithms mostly focus on learning efficiently, either through efficient exploration (Strehl and Littman 2008; Bellemare et al. 2016a), by using options (Sutton, Precup, and Singh 1999; Bacon, Harb, and Precup 2017), hierarchies (Dietterich 2000; Kulkarni et al. 2016; Gopalan et al. 2017), or by decomposing the learning problem into a multi-agent setting (van Seijen et al. 2017; Sunehag et al. 2017; Russell and Zimdars 2003). We focus on how difficult representing or approximating the optimal value function is. In this context, action-gaps play an important role, because when action-gaps are small accurately approximating action-values and recovering the highest valued action can become intractable. Approximating the value function becomes easier if action-gaps are large (Bellemare et al. 2016b). However, Bellemare et al. present new Bellman operators to increase the action-gap. In contrast to their work, we analyze the dependency of an MDP's action-gaps on the planning horizon and look at the structure of the transition dynamics. Our results indicate that the state space should be partitioned into fully connected subsets along bottle-necks (Şimşek and Barto 2004; Stolle and Precup 2002; Bacon 2013) and for each partition a separate lower discount factor should be used to allow for the largest possible action-gaps. This aligns well with our value function complexity result: If such a partitioning would be present, then the optimal policy for each state partition could be solved with a separate value function that has a lower complexity than a single value function that solves the entire problem.

While well chosen options can make planning easier and reduce the sample complexity of a learning algorithm (Brunskill and Li 2014), we present theoretical evidence suggesting the benefits of options or value function decomposition methods in terms of representing the solution to a Long Horizon Problem. Further, we present first results indicating how options can be beneficial for value function approximation, which to our knowledge is an open question (Bacon and Precup 2015; Bacon, Harb, and Precup 2017).

6 Conclusion

We presented a novel perspective on why LHPs can be hard learning problems by only considering the complexity of representing the optimal solution and isolating the representation problem from the learning problem. Our analysis indicates that in order to allow the highest possible action-gaps (to make value function approximation easier), the state space should be partitioned along bottle-necks and each partition should use its own reduced discount factor. We hope that our results can guide the design of novel options and value function decomposition algorithms.

A Value Function Complexity Theorem

Proof of Lemma 1. The set equality is proven by first showing $\boldsymbol{v} \in \mathcal{V}^*_\gamma \implies \boldsymbol{v} \in \left\{ \boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}_{\geq 0} \middle| ||\boldsymbol{v}||_\infty \leq \frac{R_{\max}}{1-\gamma} \right\}$ and then

$\boldsymbol{v} \in \left\{ \boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}_{\geq 0} \middle| ||\boldsymbol{v}||_\infty \leq \frac{R_{\max}}{1-\gamma} \right\} \implies \boldsymbol{v} \in \mathcal{V}^*_\gamma$.

For the first direction, we observe that the value of each state lies in $\left[0, \frac{R_{\max}}{1-\gamma}\right]$. The bound $\frac{R_{\max}}{1-\gamma}$ is also tight because we can choose a reward function that always returns $R_{\max}$. Further, if only zero rewards are given, then $\boldsymbol{v} = \mathbf{0}$.

For the reverse direction, we choose an arbitrary $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}_{\geq 0}$ with $||\boldsymbol{v}||_\infty \leq \frac{R_{\max}}{1-\gamma}$ and construct an MDP M such that $\boldsymbol{v}$ is the value function of the optimal policy in M. We assume that M only contains self transitions and that for every action a the state-to-state transition matrix $\boldsymbol{P}_a$ equals the identity matrix. The reward function is set to

$$r(s,a) = \begin{cases} (1-\gamma)\boldsymbol{v}_s & \text{if } a = a^* \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where $\boldsymbol{v}_s$ is the entry of the vector $\boldsymbol{v}$ corresponding to state s and $a^* \in \mathcal{A}$ is fixed but arbitrary. By construction, the optimal policy π^* for M is to select action a^* at every state. Because all transitions are deterministic self-loops, we have that $V^{\pi^*}(s) = \sum_{t=1}^{\infty} \gamma^{t-1}(1-\gamma)\boldsymbol{v}_s = \boldsymbol{v}_s$, and $\boldsymbol{v} \in \mathcal{V}^*_\gamma$. □

Theorem 1 presents an identity for the Rademacher complexity of the value function hypothesis space $\mathcal{V}^*_\gamma$. The empirical Rademacher complexity (Shalev-Shwartz and Ben-David 2014) can be generalized to vector spaces $\mathcal{X} \subseteq \mathbb{R}^n$ (Balcan 2011) and is then defined as

$$\mathfrak{R}(\mathcal{X}) = \frac{1}{n}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n} \sigma_i x_i\right], \tag{19}$$

which measures the ability of $\mathcal{X}$ to fit a random Rademacher noise vector $\boldsymbol{\sigma}$ whose entries σ_i either equal +1 or -1. Intuitively, the generalized Rademacher complexity measures the correlation between entries in the same vector for some subspace of $\mathbb{R}^n$. The generalized Rademacher complexity decreases as the entries of a vector become more correlated.

Proof of Theorem 1. The generalized Rademacher complexity of $\mathcal{V}^*_\gamma$ can be computed by simplifying definition (19). By Lemma 1 we can assume that the entries of every vector $\boldsymbol{v} \in \mathcal{V}^*_\gamma$ lies in the interval $\left[0, \frac{R_{\max}}{1-\gamma}\right]$, hence

$$\begin{aligned}
\mathfrak{R}(\mathcal{V}^*_\gamma) &= \frac{1}{|\mathcal{S}|}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{v}\in\mathcal{V}^*_\gamma} \sum_{i=1}^{|\mathcal{S}|} \sigma_i v_i\right] \\
&= \frac{1}{|\mathcal{S}|}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{i=1}^{|\mathcal{S}|} \mathbf{1}_{[\sigma_i=1]} \frac{R_{\max}}{1-\gamma}\right] \qquad (20) \\
&= \frac{1}{|\mathcal{S}|} \underbrace{\mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{i=1}^{|\mathcal{S}|} \mathbf{1}_{[\sigma_i=1]}\right]}_{=|\mathcal{S}|/2} \frac{R_{\max}}{1-\gamma} = \frac{R_{\max}}{2(1-\gamma)}. \qquad (21)
\end{aligned}$$

Line (20) follows by $v_i \in \left[0, \frac{R_{\max}}{1-\gamma}\right]$ and $\sum_{i=1}^{|\mathcal{S}|} \sigma_i v_i$ can only be maximized by setting all $v_i = \frac{R_{\max}}{1-\gamma}$ where $\sigma_i = 1$ and $v_i = 0$ where $\sigma_i = 0$. Line (21) follows by σ_i being +1 or -1 with equal probability, so in expectation half of the entries of the vector $\boldsymbol{\sigma}$ will be +1 and the other half -1. □

B Action-gap Theorem

Proof of Lemma 2. First, we observe that

$$\begin{aligned}
\text{MAG}(\mathcal{S}_C) &= \max_{s\in\mathcal{S}_C}\left[\max_{a\in\mathcal{A}} Q^{\pi^*}(s,a) - \min_{a\in\mathcal{A}} Q^{\pi^*}(s,a)\right] \\
&= \max_{s\in\mathcal{S}_C}\left[V^{\pi^*}(s) - \min_{a\in\mathcal{A}} Q^{\pi^*}(s,a)\right] \\
&\leq V_{\max,\gamma} - \min_{s_C\in\mathcal{S}_C, a\in\mathcal{A}} Q^{\pi^*}(s_C,a), \qquad (22)
\end{aligned}$$

where $V_{\max,\gamma} = \max_{s\in\mathcal{S}_C} V^{\pi^*}(s)$. Using the non-negative reward assumption, the second term can be bounded with

$$\begin{aligned}
\min_{s_C,a} Q^{\pi^*}(s_C,a) &= \min_{s_C,a}\left[r(s_C,a) + \gamma\mathbb{E}_{s'}\left[V^{\pi^*}(s')\right]\right] \\
&\geq \gamma \min_{s_C,a}\mathbb{E}_{s'}\left[V^{\pi^*}(s')\right] = \gamma \min_{s_C\in\mathcal{S}_C} V^{\pi^*}(s_C) \qquad (23)
\end{aligned}$$

To lower bound the minimal state value, we repeat a similar argument presented in the proof of (Jiang, Singh, and Tewari 2016, Proposition 4): Since V^{π^*} evaluates the optimal policy, we can lower bound V^{π^*} with the value function of the policy $\pi_{\text{to-V-max}}$ which minimizes the distance to the state of value $V_{\max,\gamma}$. Then,

$$\begin{aligned}
\min_{s\in\mathcal{S}_C} V^{\pi^*}(s) &\geq \mathbb{E}_{\pi_{\text{to-V-max}}}\left[\gamma^T V_{\max,\gamma}\right] \\
&\geq \gamma^{\mathbb{E}[T]} V_{\max,\gamma} \qquad \text{(by Jensen's Ineq.)} \\
&\geq \gamma^{D_{\mathcal{S}_C}} V_{\max,\gamma}. \qquad (24)
\end{aligned}$$

The last step follows by the Diameter Definition 3, because $\mathbb{E}[T] \leq D_{\mathcal{S}_C}$. Substituting (24) into (23) and the result into (22) results in $\text{MAG}(\mathcal{S}_C) \leq (1-\gamma^{D_{\mathcal{S}_C}+1})V_{\max,\gamma}$. □

Proof of Lemma 3. It suffices to show a bound on $V^{\pi^*}(s)$ for arbitrary $s \in \mathcal{S}_C$. Let N be a random variable describing the number of time steps a trajectory started in $\mathcal{S}_C$ spends in $\mathcal{S}_C$ and assume that $\mathbb{E}[N] < \infty$. For any state $s \in \mathcal{S}_C$,

$$\begin{aligned}
V^{\pi^*}(s) &= \mathbb{E}_{\pi^*}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \middle| s_1 = s\right] \\
&\leq \mathbb{E}_{\pi^*}\left[\sum_{t=1}^{N} \gamma^{t-1} R_{\max} + \gamma^N F_{\max} \middle| s_1 = s\right] \\
&\leq \mathbb{E}_{\pi^*}\left[\frac{1-\gamma^N}{1-\gamma} R_{\max} + \gamma^N F_{\max} \middle| s_1 = s\right] \\
&\leq \frac{1-\gamma^{\mathbb{E}[N]}}{1-\gamma} R_{\max} + \mathbb{E}_{\pi^*}\left[\gamma^N F_{\max} | s_1 = s\right] \qquad (25) \\
&\leq \frac{1-\gamma^{\mathbb{E}[N]}}{1-\gamma} R_{\max} + F_{\max}, \qquad \text{(Using } \gamma < 1\text{)}
\end{aligned}$$

where (25) follows using Jensen's Inequality and $F_{\max}$ is an upper bound on the value of all states outside of $\mathcal{S}_C$ that are reachable within one time step. □

References

Bacon, P.-L., and Precup, D. 2015. Learning with options: Just deliberate and relax. In *NIPS Bounded Optimality and Rational Metareasoning Workshop*.

Bacon, P.-L.; Harb, J.; and Precup, D. 2017. The option-critic architecture. In *AAAI*, 1726–1734.

Bacon, P.-L. 2013. On the Bottleneck Concept for Options Discovery: Theoretical Underpinnings and Extension in Continuous State Spaces. Master's thesis, McGill University, Montreal, Canada.

Balcan, M.-F. 2011. Machine learning theory - rademacher complexity. http://www.cs.cmu.edu/~ninamf/ML11/lect1117.pdf. Accessed: 2017-08-21.

Barto, A. G.; Singh, S.; and Chentanez, N. 2004. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, 112–19.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47:253–279.

Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016a. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 1471–1479.

Bellemare, M. G.; Ostrovski, G.; Guez, A.; Thomas, P. S.; and Munos, R. 2016b. Increasing the action gap: New operators for reinforcement learning. In *AAAI*, 1476–1483.

Brunskill, E., and Li, L. 2014. Pac-inspired option discovery in lifelong reinforcement learning. In Xing, E. P., and Jebara, T., eds., *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 316–324. Bejing, China: PMLR.

Dietterich, T. G. 2000. State abstraction in maxq hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 994–1000.

Gopalan, N.; Littman, M. L.; MacGlashan, J.; Squire, S.; Tellex, S.; Winder, J.; and Wong, L. L. 2017. Planning with abstract markov decision processes. In *ICAPS*.

Jiang, N.; Kulesza, A.; Singh, S.; and Lewis, R. 2015. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 1181–1189. International Foundation for Autonomous Agents and Multiagent Systems.

Jiang, N.; Singh, S. P.; and Tewari, A. 2016. On structural properties of mdps that bound loss due to shallow planning. In *IJCAI*, 1640–1647.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4:237–285.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Kulkarni, T. D.; Narasimhan, K.; Saeedi, A.; and Tenenbaum, J. B. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *CoRR* abs/1604.06057.

Laroche, R.; Fatemi, M.; Romoff, J.; and van Seijen, H. 2017. Multi-advisor reinforcement learning. *CoRR* abs/1704.00756.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, 278–287.

Russell, S. J., and Zimdars, A. 2003. Q-decomposition for reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 656–663.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Şimşek, Ö., and Barto, A. G. 2004. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 95. ACM.

Stolle, M., and Precup, D. 2002. Learning options in reinforcement learning. In *International Symposium on Abstraction, Reformulation, and Approximation*, 212–223. Springer.

Strehl, A. L., and Littman, M. L. 2008. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences* 74(8):1309–1331.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, 881–888. ACM.

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

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V. F.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2017. Value-decomposition networks for cooperative multi-agent learning. *CoRR* abs/1706.05296.

Sutton, R. S., and Barto, A. G. 1998. *Introduction to reinforcement learning*, volume 135. MIT Press Cambridge.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.

van Seijen, H.; Fatemi, M.; Romoff, J.; Laroche, R.; Barnes, T.; and Tsang, J. 2017. Hybrid reward architecture for reinforcement learning. *CoRR* abs/1706.04208.

Von Neumann, J., and Morgenstern, O. 1945. *Theory of games and economic behavior*. Princeton University Press Princeton, NJ.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.