Lipschitz Lifelong Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the problem of reusing prior experience when an agent is facing a series of Reinforcement Learning (RL) tasks. We focus on the study and the exploitation of the optimal value function's Lipschitz continuity in the task space with respect to a certain metric. These theoretical results lead us to a value transfer method for Lifelong RL. Based on this, we build a PAC-MDP algorithm exploiting the continuity properties to accelerate learning. We illustrate the benefits of the method in Lifelong RL experiments.

1 Introduction

Lifelong Reinforcement Learning (RL) is a problem where an agent faces a series of RL tasks, drawn sequentially. Transferring the knowledge of prior experience while solving new tasks is a key question in that setting (see Lazaric [2012] or Taylor and Stone [2009] for surveys). We elaborate on the intuitive idea that *similar* tasks should allow a large amount of transfer. Thus, an agent able to measure this similarity online should be able to perform transfer from prior source tasks accordingly. Measuring the amount of inter-tasks similarity is not new [Song et al., 2016, Ammar et al., 2014, Mahmud et al., 2013, Brunskill and Li, 2013, Carroll and Seppi, 2005, Fernández and Veloso, 2006, Lazaric et al., 2008]. Following this idea, we present a novel method for non-negative value transfer practically deployable in the Lifelong RL setting.

Our contributions can be listed as follows. First, we study theoretically the Lipschitz continuity of the optimal value function in the task space (Section 3). Then, we use that continuity to propose a value-transfer method based on a local distance between MDPs (Section 4). Full knowledge of both MDPs is not required and the transfer is non-negative, which makes the method practical and safe. In Section 4.2, we build a PAC-MDP algorithm applying this transfer method online in the Lifelong RL setting. We provide sample and computational complexity bounds accordingly and showcase the algorithm in Lifelong RL experiments.

2 Background

Reinforcement Learning (RL, Sutton et al. [1998]) is a framework for sequential decision making. The problem is typically modelled as a Markov Decision Process (MDP, Puterman [2014]) consisting in a 4-tuple $\langle \mathcal{S}, \mathcal{A}, R, T \rangle$ where $\mathcal{S}$ is a state space, $\mathcal{A}$ an action space, R_s^a is the expected reward of taking action a in state s and $T_{ss'}^a$ is the transition probability of reaching state s' when taking action a in state s. Without loss of generality, we assume that $R_s^a \in [0, 1]$ for all $s, a \in \mathcal{S} \times \mathcal{A}$. The expected cumulative return $\sum_t \gamma^t R_{s_t}^{a_t}$ obtained along a trajectory starting with state s and action a is noted $Q(s, a)$ and called the Q-function. The optimal Q-function Q^* is the highest attainable expected return from (s, a) and $V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a)$ is the optimal value function in s.

Lifelong RL [Silver et al., 2013, Brunskill and Li, 2014, Abel et al., 2018] is the problem of experiencing a series of MDPs drawn from an unknown distribution $\mathcal{D}$. Each time an MDP is

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

encountered, a classical RL problem takes place where the agent is able to interact with its environment to maximize its expected return. In this setting, it is reasonable to think that knowledge gained on previous MDPs could be re-used to improve the performance in new MDPs. In this paper, we provide a novel method for such transfer by characterizing the way the optimal Q-function can evolve across tasks. We restrict the scope of the study to the case where sampled MDPs share the same state-action space $\mathcal{S} \times \mathcal{A}$. These restricted sets of MDPs give a relevant insight on the question we try to answer and are of great importance in the RL literature. We will refer to the reward and transition functions of an MDP as a model. Thus, in the considered Lifelong RL setting, a new model is sampled each time a new task is drawn.

3 Lipschitz continuity of the optimal Q-function

The intuition we build on is that similar MDPs should have similar optimal Q-functions. Formally, this insight can be translated into a continuity property of that function in the MDP space $\mathcal{M}$. We show in this section that the optimal Q-function is Lipschitz continuous with respect to a metric in the MDP space. For that purpose, we introduce a local pseudo-metric characterizing the distance between the models of two MDPs at a particular state-action pair.

Definition 1. *For two MDPs $M = \langle \mathcal{S}, \mathcal{A}, R, T \rangle$ and $\bar{M} = \langle \mathcal{S}, \mathcal{A}, \bar{R}, \bar{T} \rangle$, for any function $f : \mathcal{S} \to \mathbb{R}^+$, we define the **pseudo-metric between their models** at $(s, a) \in \mathcal{S} \times \mathcal{A}$ w.r.t. f as:*

$$D_f\left(\langle R, T \rangle, \langle \bar{R}, \bar{T} \rangle\right)(s, a) \triangleq |R_s^a - \bar{R}_s^a| + \sum_{s' \in \mathcal{S}} f(s') |T_{ss'}^a - \bar{T}_{ss'}^a|. \tag{1}$$

This pseudo-metric is relative to a positive function f, which will be defined later for the needs of our results. It should be noted that the definition holds for finite state-action spaces. We see this restriction as a necessary step in the study of the Lipschitz continuity of the Q-function in the exact case. We now state our main result. We denote by Q_M^* the optimal Q-function of the MDP $M \in \mathcal{M}$. All the proofs of the paper can be found in the Appendix.

Theorem 1. ***Local Lipschitz continuity.*** *For any two MDPs M and $\bar{M}$, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$,*

$$|Q_M^*(s, a) - Q_{\bar{M}}^*(s, a)| \leq d_M^{\bar{M}}(s, a) \tag{2}$$

*where **the local MDP distance** [1] $d_M^{\bar{M}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined with the following fixed-point equation:*

$$d_M^{\bar{M}}(s, a) = D_{\gamma V_{\bar{M}}^*}\left(\langle R, T \rangle, \langle \bar{R}, \bar{T} \rangle\right)(s, a) + \gamma \sum_{s' \in \mathcal{S}} T_{ss'}^a \max_{a' \in \mathcal{A}} d_M^{\bar{M}}(s', a'). \tag{3}$$

This result establishes that the distance between the optimal Q-functions of two MDPs at $(s, a) \in \mathcal{S} \times \mathcal{A}$ is controlled by a local distance between the MDPs. The latter can be computed by Dynamic Programming (DP, Bellman [1957]). For the result to hold, the function $f : s \mapsto \gamma V_{\bar{M}}^*(s) \in \mathbb{R}^+$ was selected in the definition of the model's pseudo-metric (Equation 1). Note that the local MDP distance $d_M^{\bar{M}}$ is asymmetric. Similar results for the value function of a fixed policy and the optimal value function V_M^* can easily be derived (see Appendix, Section 4). A consequence of Theorem 1 is the extension to the global Lipschitz continuity result whose statement follows.

Corollary 1. ***Global Lipschitz continuity.*** *For any two MDPs M and $\bar{M}$, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$,*

$$|Q_M^*(s, a) - Q_{\bar{M}}^*(s, a)| \leq d_M^{\bar{M}} \triangleq \frac{1}{1 - \gamma} \max_{s,a \in \mathcal{S} \times \mathcal{A}} \left[D_{\gamma V_{\bar{M}}^*}\left(\langle R, T \rangle, \langle \bar{R}, \bar{T} \rangle\right)(s, a) \right]. \tag{4}$$

The **global MDP pseudo-metric** $d_M^{\bar{M}}$ is also asymmetric. Although this result is interesting from a theoretical perspective, we will not use it for transfer because it is impractical to compute. Indeed, estimating the maximum in Equation 4 is as hard as solving both MDPs. Thus, in an online setting, the moment we can estimate $d_M^{\bar{M}}$ corresponds to the moment we have solved both MDPs, which is too late for transfer to be useful.

[1] The term distance used here is not a mathematical distance, see Appendix, Section 2 for a discussion.

4 Transfer using the Lipschitz continuity property

We use the theoretical results from Section 3 to introduce a transfer method and build an algorithm applying this method in the Lifelong RL setting. From Theorem 1, we can naturally define a local upper bound on the optimal Q-function of an MDP given the optimal Q-function of another MDP.

Definition 2. *Given two MDPs M and $\bar{M}$, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, the **Lipschitz bound on** Q^*_M **induced by** $Q^*_{\bar{M}}$ is defined by:*

$$Q^*_M(s,a) \leq U_{\bar{M}}(s,a) \triangleq Q^*_{\bar{M}}(s,a) + \min\left\{d^{\bar{M}}_M(s,a), d^M_{\bar{M}}(s,a)\right\}. \tag{5}$$

Notice the use of the $\min$ operator to account for the fact that the asymmetric distance of Theorem 1 yields two valid upper bounds. This Lipschitz bound allows shrinking the upper bound on the optimal Q-function of an MDP. In the Lifelong RL setting, we aim to exploit this property in a method guaranteeing the three conditions that

C1 the resulting algorithm is PAC-MDP [Strehl et al., 2009];

C2 the transfer accelerates learning;

C3 the transfer is non-negative.

To that end, we propose to build on the R-MAX algorithm [Brafman and Tennenholtz, 2002], which satisfies the C1 condition. This choice is motivated by the fact that R-MAX relies on an optimistic exploration of an MDP, assuming that each unknown state-action pair yields the maximum return. Thus, shrinking the optimistic upper bound with Equation 5 is expected to improve the learning rate. In R-MAX, during the solving of an MDP, $\mathcal{S} \times \mathcal{A}$ is split into a subset of known state-action pairs K and a subset of the unknown pairs. A state-action pair is known if the number of collected reward and transition samples allows estimating an ϵ-accurate model in $\mathcal{L}_1$-norm with high probability [Strehl et al., 2009]. Given the experience of a set of MDPs $\{\bar{M}_1, \bar{M}_2, \cdots\}$, we define the total bound as the concatenation of all the Lipschitz bounds induced by each previous MDP.

Property 1. *Given a partially known MDP $M = \langle \mathcal{S}, \mathcal{A}, R, T\rangle$, the set of known state-action pairs K and a set of Lipschitz bounds $\left\{U_{\bar{M}_1}, U_{\bar{M}_2}, \cdots\right\}$, the **total upper bound** U defined below is an upper bound on Q^*_M for all $s, a \in \mathcal{S} \times \mathcal{A}$*

$$U(s,a) \triangleq \begin{cases} R^a_s + \gamma \sum_{s' \in \mathcal{S}} T^a_{ss'} \max_{a'} U(s',a') & \text{if } (s,a) \in K, \\ \min\left\{\frac{1}{1-\gamma}, U_{\bar{M}_1}(s,a), U_{\bar{M}_2}(s,a), \cdots\right\} & \text{otherwise.} \end{cases} \tag{6}$$

This total upper bound is an upper bound on Q^*_M a valid heuristic (provable upper bound on Q^*_M) for the exploration of the new MDP and can be computed online with DP.

4.1 Estimating an upper bound on Q^*_M

Consider two MDPs $M = \langle \mathcal{S}, \mathcal{A}, R, T\rangle$ and $\bar{M} = \langle \mathcal{S}, \mathcal{A}, \bar{R}, \bar{T}\rangle$ with their respective sets of known state-action pairs $K, \bar{K} \subset \mathcal{S} \times \mathcal{A}$. Property 1 can be used to transfer knowledge from $\bar{M}$ to M. Let $\bar{U}$ be the total upper bound on $Q^*_{\bar{M}}$[2]. Solving Equation 6 requires the knowledge of $U_{\bar{M}}(s,a)$, which in turn requires the computation of $d^{\bar{M}}_M(s,a)$ and $d^M_{\bar{M}}(s,a)$, which are themselves computed via Equation 3. Solving Equation 3 requires computing $D_{\gamma V^*_{\bar{M}}}\left(\langle R, T\rangle, \langle \bar{R}, \bar{T}\rangle\right)(s,a)$ in all pairs (s,a) which demands the full knowledge of both MDPs models. But, those are only known in K and $\bar{K}$. We propose a method to upper bound this distance between models, based on what is known of $\langle R, T\rangle$ and $\langle \bar{R}, \bar{T}\rangle$. This allows computing a provable and usable upper bound on Q^*_M (Equation 5). This method is based on the idea of maximizing over the possible models in state-action pairs where

[2] In the worst case, if $\bar{M}$ has not been solved at all, it is the R-MAX optimistic upper bound $\frac{1}{1-\gamma}$.

they are unknown. The following quantity is an upper bound on $D_{\gamma V^*_{\bar{M}}}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a)$:

$$\hat{D}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a) \triangleq$$
$$\begin{cases} D_{\gamma U_{V^*_{\bar{M}}}}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a) & \text{if } (s,a)\in K\cap \bar{K} \\ \max_{\bar{R},\bar{T}\in\mathcal{M}} D_{\gamma U_{V^*_{\bar{M}}}}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a) & \text{if } (s,a)\in K\cap \bar{K}^c \\ \max_{R,T\in\mathcal{M}} D_{\gamma U_{V^*_{\bar{M}}}}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a) & \text{if } (s,a)\in \bar{K}\cap K^c \\ \max_{R,T,\bar{R},\bar{T}\in\mathcal{M}^2} D_{\gamma U_{V^*_{\bar{M}}}}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a) & \text{if } (s,a)\in \mathcal{S}\times\mathcal{A}\cap\left(K\cup\bar{K}\right)^c \end{cases} \tag{7}$$

with $U_{V^*_{\bar{M}}}(s)=\max_a \bar{U}(s,a)$ and K^c refers to the complement of K in $\mathcal{S}\times\mathcal{A}$. In the extreme case where $\bar{M}$ is fully explored, $\bar{U}$ is an ϵ-accurate estimate in $\mathcal{L}_1$-norm of $Q^*_{\bar{M}}$ with high probability.

Finally, to compute an upper bound $\hat{d}(s,a)$ on $d^{\bar{M}}_{M}(s,a)$ we need to upper bound the right-hand side term of Equation 3. The first term is bounded by $\hat{D}$. When (s,a) is in K, the second term can be computed exactly. Otherwise, it can be upper bounded by $\gamma\max_{s',a'}\hat{d}^{\bar{M}}_{M,n}(s',a')$. Overall, the following DP sequence of functions $\left(\hat{d}^{\bar{M}}_{M,n}\right)_{n\in\mathbb{N}}$ converges to this upper bound $\hat{d}$:

$$\hat{d}^{\bar{M}}_{M,0}(s,a)=0$$
$$\hat{d}^{\bar{M}}_{M,n+1}(s,a)=\begin{cases}\hat{D}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a)+\gamma\sum_{s'\in\mathcal{S}}T^a_{ss'}\max_{a'\in\mathcal{A}}\hat{d}^{\bar{M}}_{M,n}(s',a') \text{ if } s,a\in K,\\ \hat{D}\left(\langle R,T\rangle,\langle \bar{R},\bar{T}\rangle\right)(s,a)+\gamma\max_{s',a'\in\mathcal{S}\times\mathcal{A}}\hat{d}^{\bar{M}}_{M,n}(s',a') \text{ otherwise.}\end{cases}$$

4.2 Lipschitz R-MAX algorithm

In the Lifelong RL setting, MDPs are encountered sequentially. Each interaction with a new task yields a set of known state-action pairs with their learned models and a learned upper bound on the optimal Q-function. We propose to save this learned information to compute the total upper bound of Equation 6 and use it for transfer. Practically, the method presented in Section 4.1 is used to compute the local MDP distance of Equation 3. The resulting algorithm, Lipschitz R-MAX, is presented in Algorithm 1.

Algorithm 1 Lipschitz R-MAX algorithm

```
for each sampled MDP do
  for t = 1, 2, ... do
    s = current state
    a = arg max_a' U(s, a')
    Observe reward r and next state s'
    if enough observations for (s, a) then
      Update model at (s, a)
      for each known MDP M̄ do
        Update U_M̄    # Dynamic Programming
      end for
      Update U with Equation 6    # Dynamic Programming
    end if
  end for
  Save learned model
end for
```

Lipschitz R-MAX measures the local distance to each one of the MDPs in memory to refine its upper bound on the optimal Q-function of the current MDP. As new samples are gathered, the overestimation of the Lipschitz bounds of Equation 5 decreases and the total upper bound of Property 1 becomes more accurate. Lipschitz R-MAX is PAC-MDP (Condition C1) as stated in Properties 2 and 3 below. The sample complexity of vanilla R-MAX is $\mathcal{O}(SA/(\epsilon^3(1-\gamma)^3))$, which is improved by Lipschitz R-MAX in Property 2 and meets Condition C2. The upper bound used by Lipschitz R-MAX is provably overestimating the optimal Q-function of the current MDP as stated in Property 1, which avoids negative transfer and meets Condition C3.

Property 2. ***Sample complexity.*** *[Strehl et al., 2009] With probability $1-\delta$, Lipschitz R-MAX algorithm achieves an ϵ-optimal return in the MDP M for all but*

$$\mathcal{O}\left(\frac{|\{s,a \in \mathcal{S}\times\mathcal{A} \mid U(s,a) \geq V_M^*(s)-\epsilon\}|}{\epsilon^3(1-\gamma)^3}\right)$$

time steps, with U defined in Equation 6.

Property 3. ***Computational complexity.*** *The total computation complexity of Lipschitz R-MAX is*

$$\mathcal{O}\left(B + \frac{S^2A^2(S+\ln(A))(N+1)}{(1-\gamma)}\ln\frac{1}{\epsilon(1-\gamma)}\right)$$

with B the number of time steps, ϵ the precision of value iteration and N the memory size.

Compared to R-MAX, Lipschitz R-MAX achieves potentially better sample complexity (Property 2) to the cost of N Dynamic Programming computations (Property 3), N being the number of previously seen MDPs in the Lifelong RL setting.

4.3 Improving Lipschitz R-MAX using knowledge of the maximum distance

Lipschitz R-MAX relies on upper bounds on the local distances between the models of both MDPs (Equation 7). The quality of the Lipschitz bound on Q_M^* greatly depends on the quality of those estimates and can be improved accordingly. We discuss two methods to provide finer estimates.

1) In a lifelong RL setting, one can define the **maximum model distance at a particular s,a pair** over the whole MDP space $\mathcal{M}$: $D_{\max}(s,a) \triangleq \max_{M,\bar{M}\in\mathcal{M}^2}(D_{\gamma V_{\bar{M}}^*}\left(\langle R,T\rangle,\langle\bar{R},\bar{T}\rangle\right)(s,a))$. The **maximum model distance** is defined by $D_{\max} = \max_{s,a} D_{\max}(s,a)$. It is reasonable to think that some prior knowledge can be available to the user over the possible range of $D_{\max}$. Interestingly, such an assumption provides a way to quantify the subset of MDPs wherein one is able to perform efficient transfer. We write $U_{D_{\max}}$ the upper bound on $D_{\max}$ and will call it the **prior knowledge**. It can then be used by Lipschitz R-MAX to refine the upper bounds on the model's pseudo-distance as described in Equation 8

$$\hat{D}(\langle R,T\rangle,\langle\bar{R},\bar{T}\rangle)(s,a) \leftarrow \min\{U_{D_{\max}}, \hat{D}(\langle R,T\rangle,\langle\bar{R},\bar{T}\rangle)(s,a)\}. \tag{8}$$

It should be noted that $D_{\max} \leq (1+\gamma)/(1-\gamma)$.

2) It is also possible to estimate online the value of $D_{\max}(s,a)$ for each known s,a in the Lifelong RL setting, freeing us from making the previous hypothesis. As new MDPs are sampled, we gain new estimates of the possible models at s,a. Thus, we can build an estimate of the maximum model distance at s,a: $\hat{D}_{\max}(s,a) \triangleq \max_{M,\bar{M}\in\hat{\mathcal{M}}^2}(D_{\gamma V_{\bar{M}}^*}\left(\langle R,T\rangle,\langle\bar{R},\bar{T}\rangle\right)(s,a))$ where $\hat{\mathcal{M}}$ is the set of experienced MDPs. The pitfall is that, with few MDPs, $\hat{D}_{\max}(s,a)$ could be an underestimate of the real value. Theorem 2 provides the probability for $\hat{D}_{\max}(s,a)$ not to be an under estimate of $D_{\max}(s,a)$, allowing us to use it with high confidence. In turn, Algorithm 2 provides a practical method for estimating $D_{\max}(s,a)$ with probability $1-\delta$ and can be combined with Algorithm 1 to improve the performance.

Theorem 2. *Consider an algorithm producing ϵ-accurate model estimates for a subset of $\mathcal{S}\times\mathcal{A}$ after interacting with an MDP. For all $s,a \in \mathcal{S}\times\mathcal{A}$, after sampling m MDPs, the probability of successfully estimating $D_{\max}(s,a)$ is:*

$$Pr(\hat{D}_{\max}(s,a) \geq D_{\max}(s,a)) \geq 1 - 2(1-p_{\min})^m + (1-2p_{\min})^m$$

where $p_{\min} = \min_{M\in\mathcal{M}} Pr(M)$ is a lower bound on the sampling probability of an MDP.

The assumption of having a lower bound $p_{\min}$ on the sampling probability of an MDP is commonly seen as a non-adversarial MDP sampling strategy [Abel et al., 2018].

5 Experiments

We evaluate the Lipschitz R-MAX algorithm within three types of experiments[3]: 1) We quantify the use of the Lipschitz upper bound of Equation 5 w.r.t. the magnitude of the known $U_{D_{\max}}$ bound

[3] A link to the code will be made available before publication.

Algorithm 2 Maximum distance estimation

Input: $p_{\min}, \delta, \hat{\mathcal{M}}$
Output: $\hat{D}(s,a)$
for each known s, a **do**
  if $1-2(1-p_{\min})^{|\mathcal{M}|} + (1-2p_{\min})^{|\mathcal{M}|} \geq 1-\delta$ **then**
    $\hat{D}(s,a) \leftarrow \max_{M_1,M_2 \in \hat{\mathcal{M}}^2} (D_{\gamma V^*_{M_1}}(\langle R_1, T_1\rangle, \langle R_2, T_2\rangle)(s,a))$
  else
    $\hat{D}(s,a) \leftarrow (1+\gamma)/(1-\gamma)$
  end if
end for

(Section 4.3); 2) We show the decay of the use of $U_{D_{\max}}$ during the learning of the MDPs; 3) We showcase the performance of the algorithm in three Lifelong RL experiments. Each curve of Figures 1–6 displays values with 95% their confidence interval. Informations for the compliance with the Machine Learning Reproducibility Checklist can be found in Appendix, Section 10.

In Experiment 1, we trained R-MAX and Lipschitz R-MAX on a 4-state MDP with a specific reward model (Appendix, Section 9, Figure 3). The reward is 0 everywhere except for the unique terminal state where it is 0.8. We then repeated the experience, changing the reward of the terminal state to 1, creating a maximum gap of $D_{\max} = |R^a_s - \bar{R}^a_s| = 0.2$. We measured 1) the number of times the Lipschitz bound (Equation 5) was tighter than the optimistic bound $1/(1-\gamma)$; and 2) the time to convergence of Lipschitz R-MAX vs R-MAX. Time to convergence is understood as the number of steps until the algorithm achieves an ϵ-optimal return. The results are reported in Figure1 for various values of prior knowledge $U_{D_{\max}}$. As expected, more prior knowledge of the maximum model's distance results in a tighter Lipschitz bound which, in turn, is used more often than the optimistic bound, up to 100% of the time after $U_{D_{\max}} \geq 0.5$ in this case. It is interesting to see that the prior does not need to be the smallest upper bound ($U_{D_{\max}} = 0.2$) for the Lipschitz bound to be better. In parallel, sufficiently accurate Lipschitz bounds result in improved convergence to the optimal policy as illustrated by the speed-up curve. Concretely, this means that Lipschitz R-MAX explores fewer state-action pairs than R-MAX with the guarantees of a PAC-MDP optimistic algorithm.

Experiment 2 is designed to measure to what extent is the prior knowledge $U_{D_{\max}}$ used during learning. We trained Lipschitz R-MAX on a heat-map grid world MDP (Appendix, Section 9, Figure 4) and repeated the experience on a different MDP (Appendix, Section 9, Figure 5). Both MDPs have different models and different optimal policies. Each time a new s, a pair is learned (enough samples collected), Lipschitz R-MAX updates the Lipschitz bound. The prior knowledge is used by Lipschitz R-MAX to estimate the local model's pseudo-metric (Equation 1) as $\min\{U_{D_{\max}}, \hat{D}(\langle R, T\rangle, \langle \bar{R}, \bar{T}\rangle)(s,a)\}$ where $\hat{D}$ is the estimate of Equation 7. At each update of the Lipschitz bound, we recorded the number of times $U_{D_{\max}}$ was selected instead of $\hat{D}$ in the $\min$ operator of Equation 8 We call **use ratio** the ratio of the number of use of $U_{D_{\max}}$ vs the number of use of $\hat{D}$, displayed in Figure 2. As expected, maximum prior knowledge $U_{D_{\max}} = 0$ results in a 100% use ratio, whereas no prior knowledge, $U_{D_{\max}} = 19$, results in a 0% use ratio. With an intermediate value, $U_{D_{\max}} = 10$, Lipschitz R-MAX uses $U_{D_{\max}}$ exactly for all the unknown s, a pairs as it can be seen with the linear decay. Indeed, each time an s, a pair is updated, $U_{D_{\max}}$ is only used one fewer time. Reducing $U_{D_{\max}}$ further ($U_{D_{\max}} = 15$ and $U_{D_{\max}} = 17$ in Figure 2) produces an interesting result: the decay of the use ratio of $U_{D_{\max}}$ being faster than the linear one ($U_{D_{\max}} = 10$), we can deduce that $\hat{D}$ is tighter than $U_{D_{\max}}$ for more than one s, a pair each time a single s, a pair is updated. This property follows from the fact that Equation 7 requires the upper bound on the value function, which is not only refined for the updated s, a pair but for other pairs thanks to the propagation in the Bellman equation. As a conclusion of this last result, it can be said that Lipschitz R-MAX does not only rely on the hypothesis made on $U_{D_{\max}}$. The structure of the algorithm makes it possible to increasingly commit to the local model's estimates.

Experiment 3 aims to evaluate the performance gain of Lipschitz R-MAX in Lifelong RL experiments. We selected three MDPs represented in Appendix, Section 9, Figure 6, Figure 7 and Figure 8. The corridor task is similar to the random-walk task [Sutton et al., 1998] except that the reward at the terminal state located at the extreme right side is sampled uniformly from $[0.8, 1.0]$. The transition

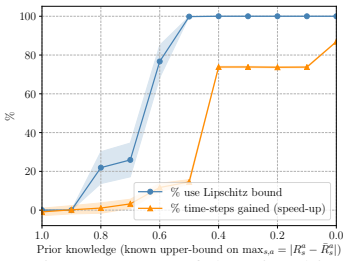

Figure 1: Use of the Lipschitz bound vs prior knowledge.

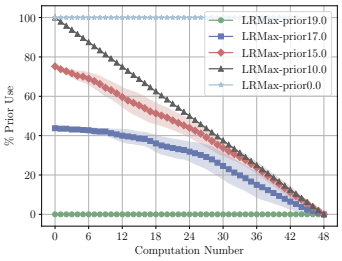

Figure 2: Prior use per update of the Lipschitz bound.

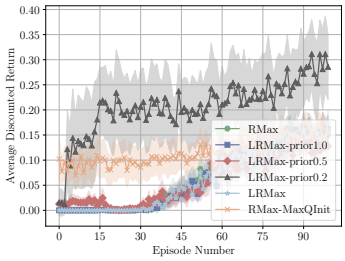

Figure 3: Maze A, average discounted return vs episode number.

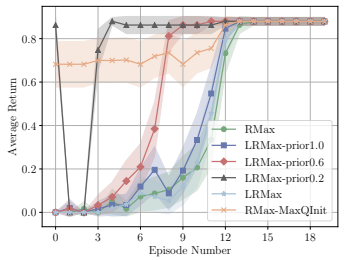

Figure 4: Corridor, average return vs episode number.

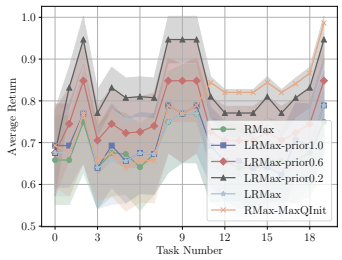

Figure 5: Corridor, average return vs task number.

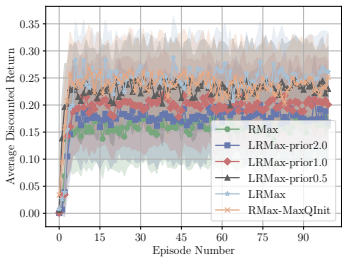

Figure 6: Maze B, average discounted return vs episode number.

model is kept constant. The maze A task is a grid world representing a maze where a slip probability (probability of applying an action other than the desired one) is sampled uniformly from $[0, 0.1]$. The maze B task is a grid world representing mazes with several different walls configurations. Thus, the set of tasks consists of different transition models and optimal policies. In mazes A and B, the goal is to reach a terminal state yielding reward 1.0 while reward is 0 elsewhere. We compared Lipschitz R-MAX using different values of $U_{D_{\max}}$ with Lipschitz R-MAX learning the local maximum model's distances with high probability (Theorem 2) and vanilla R-MAX [Brafman and Tennenholtz, 2002] learning each task from scratch. We also compare those algorithms to R-MAX with the MaxQInit algorithm from Abel et al. [2018], which constitutes a state-of-the-art modification of R-MAX to achieve efficient transfer in a Lifelong RL experiment while preserving PAC-MDP guarantees. Similarly to Theorem 2, MaxQInit relies on an estimation of the maximum Q-value of each s, a pair with high probability in order to refine the upper bound on the optimal Q-function of the current task.

Figure 4 displays the averaged return per task vs the episode number in the corridor task. Without prior knowledge ($U_{D_{\max}} = 1$), Lipschitz R-MAX achieves comparable performance to vanilla R-MAX. Increasing prior knowledge (decrease of $U_{D_{\max}}$) results in increasing performance. An interesting behaviour of Lipschitz R-MAX is observed with the maximum prior $U_{D_{\max}} = 0.2$: the first episode systematically corresponds to the optimal policy while the second one corresponds to the exploration of the *no-reward* part of the corridor. This is due to the fact that Lipschitz R-MAX starts reproducing the optimal behaviour learned on previous MDPs due to the small value of the prior upper bound $U_{D_{\max}}$. The corresponding region of $\mathcal{S} \times \mathcal{A}$ is thus explored, which refines / decreases the upper bound on the Q-function. Consequently, unexplored regions of $\mathcal{S} \times \mathcal{A}$ have higher optimistic upper bounds, which triggers their exploration thereafter. Figure 5 displays the averaged return per episode vs the task number in the corridor task. "The same performance ordering as in Figure 4 can be observed. An interesting fact is that the MaxQInit algorithm requires sampling 10 tasks before using its estimates, after which it matches the performance of R-MAX and optimal performance with high probability.

Figure 3 displays the averaged discounted return per task vs the episode number in the maze A task. In this environment, high prior (small upper bound $U_{D_{\max}} = 0.2$) was required by Lipschitz R-MAX to perform efficient transfer, which allowed the algorithm to outperform the baseline. The performance of MaxQInit is lowered due to the aforementioned need for pre-sampling of tasks. Lipschitz R-MAX using learned distances via Theorem 2 could not catch up with the other algorithms and matches the performance of the R-MAX baseline. Figure 6 displays the averaged discounted return per task vs the

episode number in the maze B task. In this task, Lipschitz R-MAX using learned distances with high probabilities outperformed the baseline and had a comparable performance to high prior Lipschitz R-MAX ($U_{D_{\max}} = 0.5$) and MaxQInit.

As a conclusion, Lipschitz R-MAX can improve the performance within Lifelong RL experiments using efficient transfer. Is it comparable and sometimes outperforms the state-of-the-art algorithm MaxQInit. Use of prior knowledge on the task distribution greatly improves the performance of the algorithm. Alternatively, learning the maximum model's distance with high probability can allow Lipschitz R-MAX to achieve good results. It should be noted that under any configurations of our experiments, like MaxQInit, Lipschitz R-MAX never yields negative transfer.

6 Related work

The idea of using a metric MDP space is not new. It has the appealing characteristic of quantifying the amount of similarity between tasks, which intuitively should be linked to the amount of transfer achievable. Song et al. [2016] define a metric based on the bi-simulation metric introduced by Ferns et al. [2004] and the Wasserstein metric [Villani, 2008]. Value transfer is performed between states with low bi-simulation distances. However, this metric requires knowing both MDPs completely and is thus unusable in the Lifelong RL setting where we expect to perform transfer before having learned the current MDP. Further, the transfer technique they propose does allow negative transfer (see Appendix, Section 1). Carroll and Seppi [2005] also define a value-transfer method based on a measure of similarity between tasks. However, this measure is not computable online and thus not applicable to the Lifelong RL setting. Mahmud et al. [2013] and Brunskill and Li [2013] propose MDP clustering methods respectively using a metric quantifying the regret of running the optimal policy of one MDP in the other MDP and the $\mathcal{L}_1$ norm between the MDP models. An advantage of clustering is to prune the set of possible source tasks. They use their approach for policy transfer, which differs from the value-transfer method proposed in this paper. Ammar et al. [2014] use a Restricted Boltzmann Machine to learn the model of a source MDP and view the prediction error on a target MDP as a dissimilarity measure in the task space. Their method makes use of samples from both tasks and is not readily applicable to the online setting considered in this paper. Lazaric et al. [2008] provide a practical method for sample transfer, computing a similarity metric reflecting the probability of the models to be identical. Their approach is applicable in a batch RL setting as opposed to our online setting. Value is a common kind of knowledge to transfer between tasks. The approach developed by Sorg and Singh [2009] is very similar to ours in the sense that they prove bounds on the optimal Q-function for new tasks, assuming that both MDPs are known and that a soft homomorphism exists between the state spaces. Brunskill and Li [2013] also provide a method that can be used for Q-function bounding in the multi-task RL setting. Abel et al. [2018] present the MaxQInit algorithm, providing transferred bounds on the Q-function with high probability while preserving PAC-MDP guarantees. Thus, their approach is very similar to ours and differs in the way the bounds are computed. They rely on statistics on the number of samples for an estimated Q-function whereas we rely on the Lipschitz continuity property of the Q-function. The high similarity between the approaches made MaxQInit an ideal candidate for comparison in our experiments.

7 Conclusion

We have studied theoretically the Lipschitz continuity property of the optimal Q-function in the MDP space. This led to a local Lipschitz continuity result, establishing that the distance between the optimal Q-functions of two MDPs at the same state-action pair is upper bounded by a local (state-action dependent) distance between MDPs. This local distance can be computed by dynamic programming. A consequence of this result is a global Lipschitz continuity property of the optimal Q-function in the MDP space w.r.t. a pseudo metric between MDPs. We then proposed a value-transfer method using the local continuity property with the Lipschitz R-MAX algorithm; practically implementing this approach in the Lifelong RL setting. The algorithm preserves PAC-MDP guarantees, accelerates the learning in subsequent tasks and performs non-negative transfer. Potential improvements of the algorithm were discussed in the form of prior knowledge introduction on the maximum distance between models and online estimation with high probability of this distance. We showcased the algorithm in lifelong RL experiments and demonstrated empirically its ability to accelerate learning. The results also confirm that no negative transfer occurs, regardless of parameter settings.

References

David Abel, Yuu Jinnai, Sophie Yue Guo, George Konidaris, and Michael Littman. Policy and Value Transfer in Lifelong Reinforcement Learning. In *International Conference on Machine Learning*, pages 20–29, 2018.

Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of MDP similarity for transfer in reinforcement learning. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

Richard Bellman. *Dynamic programming*. Princeton, USA: Princeton University Press, 1957.

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

Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. *arXiv preprint arXiv:1309.6821*, 2013.

Emma Brunskill and Lihong Li. Pac-inspired option discovery in lifelong reinforcement learning. In *International Conference on Machine Learning*, pages 316–324, 2014.

James L. Carroll and Kevin Seppi. Task similarity measures for transfer in reinforcement learning task libraries. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 803–808. IEEE, 2005.

Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 720–727. ACM, 2006.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 162–169. AUAI Press, 2004.

Alessandro Lazaric. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, pages 143–173. Springer, 2012.

Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 544–551. ACM, 2008.

MM Mahmud, Majd Hawasly, Benjamin Rosman, and Subramanian Ramamoorthy. Clustering Markov decision processes for continual transfer. *arXiv preprint arXiv:1311.3959*, 2013.

Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Daniel L. Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*, 2013.

Jinhua Song, Yang Gao, Hao Wang, and Bo An. Measuring the distance between finite Markov decision processes. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*, pages 468–476. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

Jonathan Sorg and Satinder Singh. Transfer via soft homomorphisms. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 741–748. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

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

Richard S. Sutton, Andrew G. Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.