Landmark Options Via Reflection (LOVR) in Multi-Task Lifelong Reinforcement Learning

Nicholas Denis
Department of Mathematics
University of Ottawa
ndeni032@uottawa.ca

Maia Fraser
Department of Mathematics
University of Ottawa
mfrase8@uottawa.ca

Abstract

Temporally extended actions such as options have attractive mathematical properties that can lead to accelerated convergence of reinforcement learning algorithms. This paper introduces the Landmark Options Via Reflection (LOVR) framework in which the agent maintains a set of landmark options associated to landmark states and leverages these options with a special action, interpreted as the ability to reflect. Each landmark option defines an efficient traversing of state space to the given landmark. Reflection lets the agent select the option to the landmark “closest" to the current goal state. We study this as a mechanism to guide hierarchical control with otherwise generic RL agents on higher and lower levels. We consider a lifelong learning setting where an agent is assigned a sequence of tasks, represented as episodic MDPs having varying goal states s_g but the same underlying MDP structure. We provide theoretical and empirical results demonstrating how LOVR can provide a drastic reduction in cumulative lifetime regret over baseline approaches.

1 Introduction

The strength of reinforcement learning (RL) in modelling and solving sequential decision problems is apparent in the broad and increasing range of problem-types considered by RL researchers. One example of these are lifelong learning settings [20]. They assume an RL agent is faced with a sequence of MDPs that share certain properties and are thus viewed as coming from a common *environment*. As in transfer learning [5], rather than beginning fresh with each task, an efficient lifelong agent should be able to leverage past knowledge of its experience within the environment in order to solve each new task as quickly as possible. Hierarchical RL and the use of temporally extended actions are essential in overcoming the curse of dimensionality in many settings [2]. How to structure control and learning within a hierarchy are active current research questions [8, 11, 16, 19].

Here we introduce the Landmark Options Via Reflection (LOVR) framework, a meta-algorithm for lifelong-learning settings consisting of a sequence of task MDPs which are episodic, each with its own well-defined goal state, s_g, and the rest of the MDP common across tasks. In its offline form that we present here, LOVR involves two phases. The first pre-learning phase happens "offline" and allows the agent to explore the environment: given a set of landmark states $\mathbb{L} \subseteq \mathcal{S}$, the agent learns accurate policies for arriving at each $l \in \mathbb{L}$. These policies carry over into the second (main) phase as landmark options, o_l, $l \in \mathbb{L}$ which can be accessed via an action $a_{reflect}$.

We give theoretical results for (offline) LOVR in deterministic and finite environments and also empirical results that show LOVR drastically reduces cumulative regret for a lifelong learning agent given a sequence of MDP tasks. This holds even with the number of landmark states being a small proportion of the state space. In follow-up work we address an online form of LOVR where choice of curriculum (and gradual awakening of higher levels) serves instead to provide “phases".

Hierarchical Reinforcement Learning Workshop at the 31st Conference on Neural Information Processing Systems (HRL@NIPS 2017), Long Beach, CA, USA.

2 Background and Assumptions

Multi-task and lifelong reinforcement learning We consider a multi-task RL setting where the agent is confined to a stationary environment, and throughout its lifetime is assigned a sequence of tasks. Tasks are represented as finite sequence of episodic MDPs $\mathcal{M}_i = \langle \mathcal{S}, \mathcal{A}, P, \mathcal{R}, s_{g_i}, \nu \rangle$, $i \in [T]$, with respective terminal states s_{g_i}. In summary: the state space $\mathcal{S}$, set of actions $\mathcal{A}$, transition probability kernel P, reward function $\mathcal{R}$ and initial state distribution ν all remain fixed across tasks and only the goal state varies. The goal state s_{g_i} thus encodes the i'th task, though when speaking of a single task denoted as simply s_g. Our usage of terminal state follows the common formulation [4] for episodic tasks where one wants the agent, such as a robot, to perform a single specific function, and upon completion to become inactive and remain so until a new episode or task is begun. In general one could let K_i be the number of episodes that $\mathcal{M}_i$ is run before the next task is assigned, but we will take $K_i \equiv K$ to be independent of i in the present paper. We consider the action-penalty reward structure [15] of -1 for all transitions. The goal of the lifelong learning agent is to solve for a sequence of policies $\{\pi_i^*\}_{i=1}^T$ which minimize cumulative regret.

Landmark options The options framework is a mathematically principled approach for temporally extended actions [23]. An option $o \in \mathcal{O}$ is a triple, $o = \langle I_o, \pi_o, \beta_o \rangle$, where $I_o \subseteq \mathcal{S}$ is the initiation set. Here $I_o = \mathcal{S}$. $\forall o \in \mathcal{O}$, π_o is the option policy that maps states to primitive actions, and β_o is the termination function which controls when to terminate π_o and return control back to π. The inclusion of options in a MDP results in a Semi-MDP (SMDP), where standard RL algorithms apply in the SMDP setting [23]. Landmark options were first introduced in [26, 23] as policies leading towards "landmark" states. We retain this aspect but the way they are employed in our framework is different.

Hierarchical control LOVR seeks to address hierarchical control in RL with a simple generic approach where RL agents of each level can be arbitrary as long as they provide value functions, and higher-level agents access the options produced by lower-level agents through *reflection*. This paper is intended as a first step towards eventually replacing phases by curriculum, with suitable maturing of higher levels. Aggregation of lower-agent knowledge and control of lower agent actions are key issues in biological hierarchies and AI in general [17, 10, 18]. Our future use of curriculum and gradual wakening to train lower then higher levels is partly motivated by neuroscience research showing bottom-up learning in development vs. top-down learning in plasticity [1, 7].

Related work In the Separation of Concerns (SoC) framework [19, 25] multiple agents estimate the value of a given state, and an aggregator function (higher-level agent) makes decisions by combining input of lower agents. Reflection via $a_{reflect}$ plays a similar aggregation role in our framework. Hierarchical RL approaches such as SoC or Feudal RL [19, 9, 24] allow for temporal abstraction at different levels; this is achieved at two levels by our use of landmark options.

Both MLSH [11] and LOVR are meta-algorithms for multi-task lifelong settings that allow various underlying RL agents. MLSH performs joint optimization over two sets of parameters, one for a higher-level agent which learns commonalities among tasks, and one for each task. Per task optimization is done in a reduced space of *sub*-policies which speeds discovery of a sequence leading to reward. Reducing learning problems is a motivation in our work as well (see Section 3). The contrast in approach is similar to the above-mentioned contrast between learning in development vs. plasticity. Plasticity in mature nervous systems involves online training in both directions simultaneously [13] - similar to MLSH - while LOVR is a development-style agent.

Brunskill et al [8] studied multi-task lifelong learning and provided PAC-MDP theoretical guarantees. Similar to LOVR, their framework involves two phases. Clustering of sub-policies provides the structure by which the higher-level agent accesses them. In contrast to LOVR, [8] allow a more general setting and different reward structures. Designing complex task-dependent reward functions is often prohibitive, moreover, such rewards may induce unexpected behaviours. This is a safety concern, where the more predictable action-penalty reward structure used by LOVR may be preferable.

3 Proposed Framework and Results

LOVR framework The basic version of the LOVR meta-algorithm has two phases: a pre-learning phase which is offline, and a subsequent main learning phase where the agent solves episodic MDP's in sequence. At the start of the pre-learning phase the agent is provided with a set of landmark states $\mathbb{L} \subseteq \mathcal{S}$, and by the end of the phase it has learned landmark options for all of them. In the second phase the agent can use the acquired landmark options in addition to primitive actions (details below).

Phase I: Pre-learning phase. In this phase the LOVR agent explores state space by invoking an RL agent which we purposely leave undefined. Our only assumption is that the agent learn a landmark-centric value function, V^l, and option (optimal) policy o_l for every $l \in \mathbb{L}$ with regret at most ϵ when run for a full episode. This can be done with PAC-MDP algorithms such as Delayed Q-learning [21] or through building a model (i.e. E^3 [14]). Given action-penalty reward structure, and $\gamma = 1$, $V^{l^*}(s) = -\mathbb{E}_{o_l}[\#$ of steps from s to $l]$.

Phase II: Main learning phase, with reflection. LOVR augments the set of actions with a special action, $a_{reflect}$, which allows the agent to reflect upon landmark options. We denote $\mathcal{A}_+ := \mathcal{A} \cup \{a_{reflect}\}$. An arbitrary RL agent, e.g. a form of Q-learning, is assumed on this level. Upon receiving a new task $\mathcal{M}_i$, during episode 1, with no knowledge of s_g, the LOVR agent selects actions according to $\pi{:}\mathcal{S}{\to}\mathcal{A}$. Upon arrival at the goal state s_g, the first episode terminates, and the agent has knowledge of s_g. For all subsequent episodes of this task the agent selects actions according to $\pi{:}\mathcal{S}{\to}\mathcal{A}_+$. $a_{reflect}$ can be seen as an aggregator function from the Separation of Concerns framework, in that the agent can only access the landmark options $o_l \in \mathcal{O}$ via reflection. If at time t $\pi(s_t) = a_{reflect}$, the agent seeks the landmark state l that can reach s_g in the fewest number of steps possible: $\underset{l \in \mathbb{L}}{\operatorname{argmax}} \, V^{s_g}(l)$, and then selects the corresponding landmark option o_l. If $s_g \notin \mathbb{L}$, then $\underset{l \in \mathbb{L}}{\operatorname{argmax}} \, V^l(s_g)$ is used as a proxy. β_l terminates the landmark option o_l when the agent has taken strictly more than $|V^l(s_{t'})|$ # steps, where t' is the time when $a_{reflect}$ was selected, signifying the agent has taken more time (actions) in attempting to reach l from $s_{t'}$ than originally expected. Note $a_{reflect}$ is treated as any other action and is subject to learning updates. For our experiments we also compare to a LOVR agent using $\pi{:}\mathcal{S}{\to}\mathcal{A}_{\mathcal{O}} := \mathcal{A}_+ \cup \mathcal{O}$, which may access the landmark options o_l directly without reflection.

Theoretical results We now state our main theoretical observation when $\mathbb{L} = \mathcal{S}$. The analysis of the setting $\mathbb{L} \neq \mathcal{S}$ is out of the scope of the present paper, and will be addressed in future work.

Proposition 1. *Let $\mathcal{M}$ be an MDP as defined in Section 2 with action-penalty reward structure, and P deterministic, $\mathcal{M}$ of finite diameter D and $\mathbb{L} = \mathcal{S}$. Suppose $\forall l \in \mathbb{L}$, o_l is ϵ-accurate, then given a sequence of episodic tasks in $\mathcal{M}$ parametrized by goal state, the lifelong cumulative regret of the LOVR agent after T tasks each lasting K episodes is $\leq T\big(2\mathcal{S}\mathcal{A}D + (K-1)\big)$.*

Connection with Curriculum Learning. It is known [3] that a supervised learning problem can be converted to an RL problem by mapping smaller losses to larger rewards. We remark moreover that this conversion (to MDP's of a restricted type) can be done so as to carry over accuracy statements, i.e., so that an RL agent that achieves ϵ-optimal average reward with probability $1 - \delta$ after time T_0 can be used to define a supervised learning algorithm that given a finite sample of size T_0 from p produces with probability $1 - \delta$ an ϵ-optimal $h \in \mathcal{H}$. Results in the supervised learning setting [12] establish reduced overall sample-size requirements when a complex problem is broken into a sequence of two simpler ones according to a factorization in the statistical model for the original problem. These immediately give theoretical statements on the RL side such that there is a provable benefit in terms of decreased # steps when curriculum learning is used in corresponding multi-task RL settings: for example training an agent first to produce invariant features, then to perform a classification. LOVR is a first step in this direction - with options learned in one phase being invoked in the next. We aim to use more sophisticated curricula instead in subsequent work.

Experimental results: Grid world We test LOVR in a 15x15 gridworld domain using tabular Q-learning, and allow the agent to explore randomly throughout the environment for the pre-learning phase. During the second phase of learning we sample 100 tasks randomly, each task represented by a random goal and initialization state, ensuring that $s_0 \neq s_g$. Each task is run for 1000 episodes where after every training episode (using ϵ-greedy, ϵ := 0.1) a test episode is carried out (greedy policy). Greedy evaluation episodes run until the agent arrives at s_g, or a maximum of 5000 time steps has passed. We compare LOVR+Q-learning to baseline Q-learning. A grid search on learning rate and Q function initialization was performed. We report on the highest performing parameters, though results were quite similar across all parameters. In this environment we have $|\mathcal{S}| = 225$. We performed experiments with $|\mathbb{L}| \in \{9, 25, 225\}$. We compare two action sets, $\mathcal{A}_+$ and $\mathcal{A}_{\mathcal{O}} := \mathcal{A}_+ \cup \mathcal{O}$. Both deterministic and stochastic gridworlds were tested. In the stochastic setting, with $p = 0.8$ the agent takes the action it selected, and with $1 - p$ the agent moves to one of the states adjacent to its current state with uniform probability. All experiments were repeated for n=100 trials.

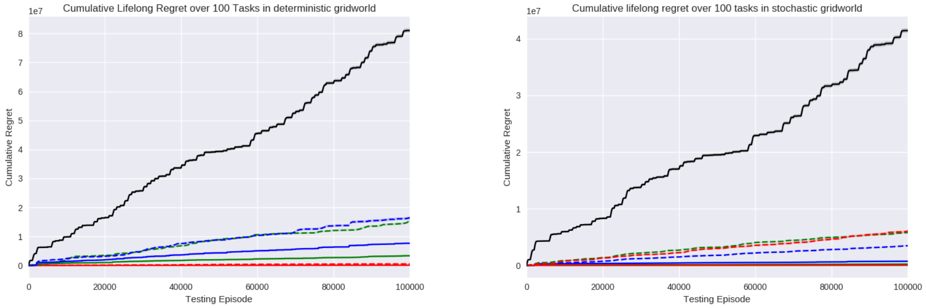

Figure 1: Cumulative lifelong regret. Left: Deterministic, Right: Stochastic gridworld. Black-baseline Q-learning, Blue-L = 9, Green-L = 25, Red-L = 225 landmarks. Dashed: $\mathcal{A}_{\mathcal{O}}$, solid: $\mathcal{A}_{+}$.

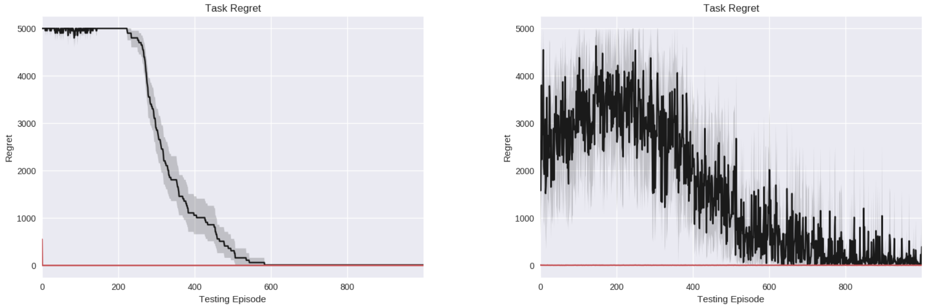

Figure 2: Single Task Regret (Task 1). Left: Deterministic, Right: Stochastic gridworld. Black-baseline Q-learning, Red-L = 225 landmarks with $\mathcal{A}_{+}$

Figure 1 demonstrates that the LOVR implementations drastically reduce the regret over the lifetime of the agent in both the stochastic and deterministic settings, as compared to baseline. We observe also that implementations of LOVR using $\mathcal{A}_{+}$ consistently outperform those using $\mathcal{A}_{\mathcal{O}}$. These empirical findings are a nice proof of concept, since using $\mathcal{A}_{+}$ only augments the set of actions by a single element, $a_{reflect}$, while adding the landmark options as well would significantly increase the number of actions, hence computations.

Table 1: Mean fold reduction in cumulative lifelong regret over 100 tasks as compared to baseline.

L, $\mathcal{A}$	9, $\mathcal{A}_{+}$	9, $\mathcal{A}_{\mathcal{O}}$	25, $\mathcal{A}_{+}$	25, $\mathcal{A}_{\mathcal{O}}$	225, $\mathcal{A}_{+}$	225, $\mathcal{A}_{\mathcal{O}}$
Deterministic	10.6	4.9	24.3	5.2	1148.3	159.9
Stochastic	56.9	12.0	186.9	7.2	1491.4	6.8

To demonstrate the regret experienced during a single task, the regret from the first task was plotted for both deterministic and stochastic settings with $L = 225$, using $\mathcal{A}_{+}$ (Fig. 2). Note these results are quite typical across all tasks. In the deterministic setting, the agent experiences regret in the first episode, but following reflection, follows a landmark option that has zero regret for the remaining 999 episodes, whereas for the baseline Q-learning agent with no landmarks, it must solve each task separately, requiring a few hundred episodes before finding the optimal policy. Similar results can be seen in the stochastic setting. Though the stochastic plot appears to receive no regret for the LOVR agent, non-zero regret is accumulated, however due to the scale of the plot is hard to see.

Data entry using a software GUI We applied the LOVR framework on a Google questionnaire form domain, where each MDP task is identified with a possible way to fill out the three-question form, followed by clicking submit. The primitive actions were moving the mouse in cardinal directions, or left-clicking. A landmark option was learned for every form-possibility. A video (available in Supplementary Material) demonstrates the LOVR agent's performance on this problem - as an applied example of highly sequential tasks within a constrained environment.

Discussion We introduced the LOVR framework as a general approach that uses reflection to accomplish hierarchical control for multi-task lifelong RL problems. We gave theoretical results for simple settings, as well as empirical results. The LOVR framework is amenable to refining an agent over early parts of its lifetime so it can solve more complex tasks with time.

Acknowledgements The authors are grateful to Doina Precup for her helpful discussions.

References

[1] M. Ahissar and S. Hochstein, *The reverse hierarchy theory of visual perceptual learning*. Trends Cogn Sci., 8(10): 457–464, 2004.

[2] A. G. Barto and S. Mahadevan, *Recent Advances in Hierarchical Reinforcement Learning*. Discrete Event Dynamic Systems 13(4):341–379, 2003.

[3] A. G. Barto and T. G. Dietterich, *Reinforcement learning and Its relationship to supervised learning* In Handbook of Learning and Approximate Dynamic Programming, Chapter 2, J. Si, A. Barto, W. Powell, and D. Wunsch (Eds.), Wiley-IEEE Press, 47-64, 2004.

[4] A. Barto and R. Sutton, *Reinforcement Learning: And Introduction*. 2nd edition (draft), A Bradford Book, MIT Press, 2016.

[5] J. Baxter, *A model of Inductive Bias Learning*. Journal of Artificial Intelligence Research 12:149-198, 2000.

[6] Y. Bengio, D.-H. Lee, J. Bornschein, and Z. Lin, *Towards biologically plausible deep learning*. arXiv preprint arXiv:1502.04156, 2015.

[7] J. Bourne and M. Rosa, *Hierarchical development of the primate visual cortex, as revealed by neurofilament immunoreactivity: early maturation of the middle temporal area (MT)*. Cereb Cortex, 16(3): 405–514, 2006.

[8] E. Brunskill and L. Li, *Sample Complexity of Multi-task Reinforcement Learning*. Conference on Uncertainty in Artificial Intelligence (UAI), 2013.

[9] Dayan, P., Hinton, G.E., *Feudal reinforcement learning*. NIPS. 5: 271-278, 1998.

[10] J.-P. Forestier and P. Varaiya, *Multilayer control of large Markov chains*. IEEE Transactions on Automatic Control 23(2): 298–305, 1978.

[11] K. Frans, J. Ho, X. Chen, P. Abbeel, J. Schulman, *Meta Learning Shared Hierarchies*. arXiv preprint arXiv:1710.09767, 2017.

[12] M. Fraser, *Multi-step learning and underlying structure in statistical models*. NIPS 29: 4815–4823, 2016.

[13] J. Guergiuev, T. Lillicrap, B. Richards, *Towards deep learning with segregated dendrites*. arXiv preprint arXiv:1610.00161, 2016.

[14] M. Kearns and S. Singh, *Near-Optimal Reinforcement Learning in Polynomial Time*. Machine Learning, 49, 209–232, 2002

[15] S. Koenig and R.G. Simmons, *Complexity Analysis of Real-Time Reinforcement Learning*. AAAI, 99–105, 1993.

[16] R. Laroche, M. Fatemi, J. Romoff, H. van Seijen, *Multi-Advisor Reinforcement Learning*. arXiv preprint arXiv:1704.00756, 2017.

[17] M. Minsky, *The Society of Mind*. Simon & Schuster, 1986.

[18] R. Parr and S. Russell, *Reinforcement Learning with Hierarchies of Machines*. NIPS 10:1043-1049, 1997

[19] H. van Seijen, M. Fatemi, J. Romoff, R. Laroche, *Separation of Concerns in Reinforcement Learning*. arXiv preprint arXiv:1612.05159, 2016

[20] D. Silver, Q. Yang, L. Li, *Lifelong machine learning systems: Beyond learning algorithms.* AAAI Spring Symposium: Lifelong Machine Learning, 49-55.

[21] A. Strehl, L. Li, E. Wiewiora, J. Langford, M. Littman, *PAC model-free reinforcement learning*. ICML 23, 881–888, 2006.

[22] Sutton, R.S., Precup, D., Singh, S., *Intra-option learning about temporally abstract actions*. ICML 556-564, 1998.

[23] Sutton, S. R., Precup, D., Singh, S., *Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning*. Artificial Intelligence 112: 181-211, 1999.

[24] Vezhnevets, A.S., et al. *FeUdal networks for hierarchical reinforcement learning*. arXiv: 1703.01161, 2017.

[25] van Seijen, H., et al. *Hybrid Reward Architecture for Reinforcement Learning* arXiv: 1706.04208, 2017.

[26] Mann, T.A., Mannor, S., Precup, D., *Approximate value iteration with temporally extended actions*. Journal of Artificial Intelligence Research. 53, 375-438, 2015.