The Expected-Length Model of Options

David Abel[*1], **John Winder**[*2], **Marie desJardins**[3] and **Michael Littman**[1]
[1]Brown University
[2]University of Maryland, Baltimore County
[3]Simmons University
david_abel@brown.edu, jwinder1@umbc.edu,
marie.desjardins@simmons.edu, mlittman@cs.brown.edu

Abstract

Effective options can make reinforcement learning easier by enhancing an agent's ability to both explore in a targeted manner and plan further into the future. However, learning an appropriate model of an option's dynamics in hard, requiring estimating a highly parameterized probability distribution. This paper introduces and motivates the Expected-Length Model (ELM) for options, an alternate model for transition dynamics. We prove ELM is a (biased) estimator of the traditional Multi-Time Model (MTM), but provide a non-vacuous bound on their deviation. We further prove that, in stochastic shortest path problems, ELM induces a value function that is sufficiently similar to the one induced by MTM, and is thus capable of supporting near-optimal behavior. We explore the practical utility of this option model experimentally, finding consistent support for the thesis that ELM is a suitable replacement for MTM. In some cases, we find ELM leads to more sample efficient learning, especially when options are arranged in a hierarchy.

1 Introduction

Making accurate long horizon predictions about the effects of an action can improve an agent's ability to make meaningful decisions. With such predictive power, agents can take into account the long-term outcomes of an action, and use this information to make informed plans that account for contingencies, uncertainty, and utility maximization. In the context of reinforcement learning, the well-studied options framework defines behavioral policies that extend actions beyond a single time-step [Sutton *et al.*, 1999]. Options can improve both learning [Konidaris and Barto, 2009; Brunskill and Li, 2014; Bacon *et al.*, 2017; Fruit and Lazaric, 2017; Machado *et al.*, 2017] and planning [Silver and Ciosek, 2012; Mann and Mannor, 2014; Mann and Mannor, 2013] by encoding relevant long-horizon behavior.

Learning models for use in making long horizon predictions has proven challenging. For instance, even ε-accurate one-step models are known to lead to an exponential increase in the error of n-step predictions as a function of the horizon [Kearns and Singh, 2002; Brafman and Tennenholtz, 2002], though recent approaches show how to diminish this error through smoothness assumptions [Asadi *et al.*, 2018]. Composing an accurate one-step model into an n-step model is known to give rise to predictions of states dissimilar to those seen during training of the model, leading to poor generalization [Talvitie, 2017]. Recent work has proposed methods for learning options that alter some aspect of the standard formalism. For example, some variations that have been explored include treating option terminations as off-policy [Harutyunyan *et al.*, 2018], regularizing for longer-duration options [Mankowitz *et al.*, 2014], and composing option models together to be jointly optimized while planning [Silver and Ciosek, 2012]. How to obtain an option model tractably, however, remains an open question.

The work we present here analyzes the problem of efficiently computing option models from experience. We first discuss the sense in which the traditional Multi-Time Model (MTM) of options [Precup and Sutton, 1997; Precup and Sutton, 1998], is highly parameterized, and thus difficult to compute or learn under reasonable constraints. In short, the density computed by MTM relies on modeling the outcome of a given option over all possible time-steps, which can be impractical to compute even in small domains. In light of this difficulty, we introduce an alternate representation, which we call the Expected Length Model (ELM). The main idea behind ELM, and indeed, this paper, is that we need not model the full joint distribution of possible outcomes of an option like MTM. Instead, we can model (1) how long, on average, the option takes to run, and (2) a categorical distribution over states where the option terminates. We analyze ELM and prove that in stochastic shortest path problems the differences in value functions induced by MTM and ELM are bounded. We corroborate these findings in learning experiments and visuals. First, we demonstrate how ELM retains the accuracy of MTM for domains using simple, flat hierarchies of options. We then consider increasingly complex environments while analyzing and visualizing ELM's benefits to both storage and sample complexity. Further, we apply ELM to hierarchies of options, showing the relative benefit over MTM when uniting state abstraction with temporal abstractions under uncertainty.

[*]The first two authors contributed equally.

2 Background

We take the usual treatment of reinforcement learning: an agent interacts with a Markov Decision Process (MDP) [Puterman, 2014], all the while learning to take actions that maximize long-term discounted reward. For further background, see Sutton and Barto [2018].

Options define temporally extended actions, a common constituent of hierarchical decision making [Sutton *et al.*, 1999; Konidaris and Barto, 2007; Konidaris and Barto, 2009; Bacon *et al.*, 2017]. More formally, an option is defined as follows:

Definition 1 (Option): *An option is a triple:* $\langle I, \beta, \pi \rangle$*, where:*

1. $I : \mathcal{S} \to \{0, 1\}$ *is a predicate on states denoting the initiation condition,*
2. $\beta : \mathcal{S} \to [0, 1]$ *is a probability distribution on* $\{0, 1\}$*, denoting the termination probability for each state.*
3. $\pi : S \to \Pr\{\mathcal{A}\}$ *is a stochastic behavioral policy.*

Intuitively, an option expresses a complete pattern of useful behavior—when to start, how to act, when to stop.

Sutton *et al.* [1999] showed that extending an MDP's action set with options results in a *semi-MDP* (SMDP). Learning with an SMDP assumes no direct knowledge of T or R, only what may be learned from experience. Reasoning about the effect of actions or options requires computing an approximate *model* of the environment's dynamics, T and R. Model-based reinforcement learning algorithms do just that—they concentrate on learning these two functions explicitly, enabling agents to predict the outcome of an action. Thus, in possessing models of the options in an SMDP, an agent may create a plan *in terms of options*, indicating how to solve the overall problem. An option's transition and reward models are used as an extension of the Bellman Equation that accommodates the termination condition. This model, originally proposed for options by Precup and Sutton; Precup and Sutton [1997; 1998], is called the Multi-Time Model (MTM), defined as follows:

Definition 2 (Multi-Time Model): *For a given* γ *and option* o*, MTM's transition and reward model are:*

$$T_\gamma(s' \mid s, o) := \sum_{k=0}^{\infty} \gamma^k \Pr(s_k = s', \beta(s_k) \mid s, o), \tag{1}$$

$$R_\gamma(s, o) := \mathop{\mathbb{E}}_{k, s_{1\ldots k}} \left[r_1 + \gamma r_2 \ldots + \gamma^{k-1} r_k \;\middle|\; s, o \right] \tag{2}$$

3 The Expected-Length Model of Options

We here introduce our new option model and analyze its properties. We begin with some intuition.

3.1 Main Idea

Our new option model explicitly models the expected number of time-steps the option will execute, instead of modeling the full distribution over trajectories the option might take. Doing so provides enough information to come up with reasonable plans while not having to learn, compute, or store a complex probability distribution. Specifically, we model the expected number of time-steps (k) taken by an option in a given state as μ_k. Using this quantity, we construct a new transition and reward model that approximates MTM well.

Definition 3 (Expected-length model of options): *The expected length model (ELM) for a given option* o *in state* s *supposes that the distribution of time-steps taken by the option can be well approximated by its expected value,* μ_k*:*

$$T_{\mu_k}(s' \mid s, o) := \gamma^{\mu_k} \Pr(s' \mid s, o), \tag{3}$$

$$R_{\mu_k}(s, o, s') := \gamma^{\mu_k} \mathbb{E}\left[r_1 + r_2 \ldots + r_k \mid s, o \right], \tag{4}$$

where $\Pr(s' \mid s, o)$ *denotes the probability of terminating in* s'*, given that the option was executed in* s*.*

Modeling only the expected number of time-steps throws away information—it ignores, essentially, the particulars of how executing the option can play out. Consider an agent in the classic Four Rooms domain, with an option for moving from the top-left room to the top-right one. Suppose the primitive actions are stochastic, with a small probability of moving the agent in the wrong direction. Due to this chance of slipping, the option may sometimes take five, ten, or even more steps to reach the top-right room. Instead of modeling the full distribution of the number of time-steps taken, ELM averages over these quantities (represented by μ_k), and models the transition as taking place over this expected number of time-steps. We provide additional intuition for ELM in Section 3.2 by working through a concrete example.

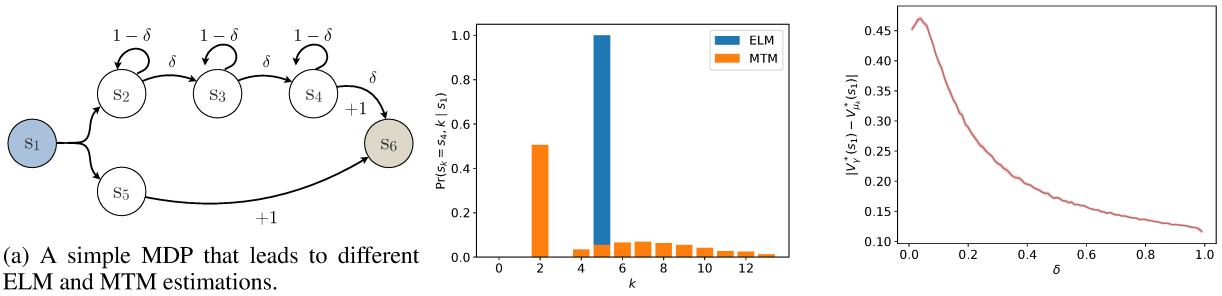
The core contribution of the paper is to show that this process of distillation is acceptable and desirable, leading to simpler models and often improving the rate at which models are learned. Specifically, we discuss two properties of ELM: (1) it is easier to estimate MTM, and (2) we prove that, under mild assumptions, it induces similar value functions to MTM, where the bound depends on primarily on the amount of stochasticity in the MDP (and option's trajectory). In experiments, we report that ELM options perform competitively to MTM, offering further support that it is a suitable option model.

3.2 Intuition

We first develop intuition behind ELM through an example, concentrating on the transition model.

Example 1. *Consider the six-state MDP in Figure 1a, chosen to accentuate the differences in ELM and MTM. Suppose an option initiates in* s_1 *(shown in blue), and terminates in* s_6 *(shown in tan). To retain the simplicity of the example, we suppose* $\beta(s_i) = 0$ *for all* $s_i \neq s_6$*. The option policy is depicted by the arrows—when the option executes its policy in* s_1*, it lands in* s_2 *with probability* $\frac{1}{2}$ *and* s_5 *with probability* $\frac{1}{2}$*. In* s_2*, when the option executes its policy, the agent stays in* s_2 *with probability* $1 - \delta$*, and transitions to* s_3 *with probability* δ *(and so on for* s_3 *and* s_4*). Conversely, in* s_5*, the option deterministically transitions to* s_6*.*

Consider now estimation of the transition into s_6: $T_\gamma(s_6 \mid s_1, o)$ under MTM. To construct a proper estimate, MTM must estimate the probability of termination in each state over all possible time-steps to determine $\Pr(s^{(1)} = s_6 \mid s_1, o), \Pr(s^{(2)} = s_6 \mid s_1, o), \ldots$. This computation involves



(a) A simple MDP that leads to different ELM and MTM estimations.

(b) Model Diff. $\delta = 0.4$

(c) Value Diff. vs. δ

Figure 1: An example illustrating the key difference between ELM and MTM. Consider the six-state MDP presented in (a), parameterized by a slip probability δ. We consider an option that initiates at s_1 and terminates in s_6, with the transition $T(s_2 \mid s_1, \cdot) = T(s_5 \mid s_1, \cdot) = 0.5$. The middle plot (b) indicates the modeling difference (for $\delta = 0.4$): ELM only models the expected number of time-steps taken by the option, whereas MTM models the full distribution over possible time-steps taken by the option. The right plot (c) presents the value function difference between the two models with respect to choices of $\delta \in [0.01 : 1.0]$, reported with almost invisible 95% confidence intervals.

estimation over arbitrarily many time-steps; in some cases, like this one, we might find a closed form based on convergence of the geometric series, but agents cannot always intuit this fact from limited data. In contrast, ELM models this distribution according to μ_k, the average number of time-steps.

Given the true MDP transition function T, we run n rollouts of the option to termination. Supposing each rollout reports (s, o, r, s', k), with r the cumulative reward received and k the number of time-steps taken, we can trivially estimate μ_k with the maximum likelihood estimator (MLE) $\hat{\mu}_k = \frac{1}{n}\sum_{i=1}^{n} k_i$. We can also estimate $\Pr(s' \mid s, o)$, the probability that o terminates in s', by modeling it as a categorical distribution with $\ell = |\mathcal{S}|$ parameters. Then, we estimate each ℓ_i with an MLE.

To summarize:

- ELM estimates μ_k and $\Pr(s' \mid s, o)$, for each s' of relevance, by using an MLE based on data collected from rollouts of the option.
- MTM must estimate the probability of terminating in each state, at each time-step. It is unclear how to capture this infinite set of probabilities of value economically.

We present their differences in the quantity $\Pr(s_k = s_6 \mid s_1, o)$, for each k, in Figure 1b. MTM (in orange) distributes the transition probability across many step lengths k. Approximately half of the time, s_6 is reached in two steps via s_5; the rest of the probability mass is spread across higher values, reflecting longer paths (via s_2). ELM (in blue) instead assumes the option takes $\mu_k = 5$ steps. For both models, each non-zero bar represents a parameter that needs to be estimated, giving a sense of the difficulty in estimating each distribution.

We also present the value difference under each model in Figure 1c, which decreases to around 0.15 as δ tends to 1 (with $\text{VMAX} = 1.0$). This trend is predicted by the analysis we conduct in Section 3.4, which suggests that the higher the variance over expected number of time-steps, the more the ELM deviates from MTM.

The example is intended to highlight the following intuition: we need not decompose future plans into the probabilities over all possible actions, over all possible time-steps; such reasoning can actually be counterproductive to the purpose of temporal abstraction.

3.3 Difficulty of Finding Option Models

The goal of ELM is to simplify MTM to be able to estimate and compute the model of a given option more efficiently.

Estimation. Learning an option's MTM involves estimating infinitely many probability distributions. Specifically, the general case would require parameters for the (potentially unbounded) number of time-steps taken to reach a given s' conditioned on initiating o in s. For such cases, a common assumption to make in analyzing complexity is to model the process only out to some finite horizon. Thus, a reasonable approximation might involve limiting the sum inside MTM to the first $\lambda = (1-\gamma)^{-1}$ steps as an artificial horizon, thereby yielding $\lambda|\mathcal{S}|^2$ parameters to estimate. In contrast, ELM requires learning the parameters of a categorical distribution indicating the probability of terminating in each state. With one multinomial for each state, any learning algorithm must estimate $2|\mathcal{S}|^2$ total parameters. Depending on the stochasticity inherent in the environment, option policy, and option-termination condition, estimating this smaller number of parameters is likely to be considerably easier ($\lambda \gg 2$).

Computation. The MTM requires performing the equivalent computation of a Bellman backup until the option is guaranteed to have terminated *just to compute the option's reward function* (Equation 2). Due to the decreasing relevance of future time-steps from γ, one might again only compute out to λ time-steps to determine R_γ and T_γ. Thus, computing R_γ is roughly as hard as computing the value function of the option's policy (at least out to λ time-steps), requiring computational hardness similar to that of an algorithm like Value Iteration, which is known to be $O(|\mathcal{S}|^2|\mathcal{A}|)$ per iteration, with a rough convergence rate of $\tilde{O}(\lambda|T|)$ for $|T|$ as a measure of the complexity of the true transition function [Littman *et al.*, 1995; Tseng, 1990]. Conversely, ELM is well suited to construction via Monte Carlo methods. Consider a single *simulated experience* $e = (s, o, r, s', t)$, of the initial state, the

option, termination state, cumulative reward, and time taken. This experience contains each data point needed to compute the components of option o's model (Equations 3 and 4), all sampled directly from the appropriate distributions. We highlight this property of ELM as desirable when the acquisition of samples is costly, as in robotics domains. With ELM, option models can be learned from these simulations, $\mathcal{E}$, with each $e \in \mathcal{E}$ needing only labels of where the option began, where it ended, how much reward it received, and how long it took. It is therefore sufficient to run a number of rollouts proportional to the desired accuracy when using ELM. Relying on such methods for computing MTM again requires estimating an arbitrarily large number of parameters, which is clearly untenable.

We note that these are not conclusive analyses of the computational and statistical difficulty of obtaining each model, but take the insights discussed to serve as sufficient motivation for further exploration of ELM. For instance, there is some similarity in determining MTM and TD(λ) when $\lambda = 1$ [Sutton, 1988], so such estimation can be feasible (see, for instance, Chapter 4 of Parr [1998]).

We now turn to our primary analysis, which illustrates the mathematical deviation between MTM and ELM for each of the transition dynamics, reward function, and value function.

3.4 Analysis

Our main theorem bounds the value difference between ELM and MTM in stochastic shortest path problems (SSPs). To prove this theorem, we make the following two assumptions, which simplifies the analysis.

Assumption 1. *All MDPs we consider are SSPs.*

We make this assumption to achieve a sharp bound in the difference of the ELM and MTM reward models.

Assumption 2. *Every option-termination condition is non-zero in every state, lower bounded by* $\beta_{\min} \in (0, 1]$.

Indeed, while these assumptions slightly limit the scope of the analysis of ELM, we take the setting to still be sufficiently interesting to offer insights about learning and using option models. We take the relaxation of each assumption as a direction for future work.

We begin with two lemmas that show the transition and rewards of ELM are reasonable approximations of MTM. All proofs are presented in the appendix.

Lemma 1. *Under Assumption 2, the ELM transition model is sufficiently close to the expected transition model of the multi-time model.*

More formally, for any option $o \in \mathcal{O}$*, for some real* $\tau > 1$*, for* $\delta = \frac{\sigma^2_{k,o}}{\tau^2}$*, and for any state pair* $(s, s') \in \mathcal{S} \times \mathcal{S}$*, with probability* $1 - \delta$*:*

$$|T_\gamma(s' \mid s, o) - T_{\mu_k}(s' \mid s, o)| \leq \gamma^{\mu_{k,o} - \tau}(2\tau + 1)e^{-\beta_{\min}}. \quad (5)$$

Lemma 2. *Under Assumptions 1 and 2, ELM's reward model is similar to MTM's reward model.*

More formally, for a given option o, for $\delta = \frac{\sigma^2_{k,o}}{\tau^2}$*, for some* $\tau > 1$*, for any state s:*

$$|R_\gamma(s, o) - R_{\mu_k}(s, o)| = |T_\gamma(s_g \mid s, o) - T_{\mu_k}(s_g \mid s, o)|. \quad (6)$$

And, thus, with probability $1 - \delta$*:*

$$|R_\gamma(s, o) - R_{\mu_k}(s, o)| \leq \gamma^{\mu_{k,o} - \tau}(2\tau + 1)e^{\beta_{\min}}. \quad (7)$$

Notably, Lemma 1 does not depend on Assumption 1—it applies to any MDP. We suspect that the reward function can also be bounded in a more general class of MDPs than SSPs, but leave such a direction open for future work. In short, the naïve method for bounding the two in non-SSPs yielded a vacuous bound larger than $\text{RMAX}/(1 - \gamma)$. With these lemmas in place, we now present our main result.

Theorem 1. *In SSPs, the value of any policy over options under ELM is bounded relative to the value of the policy under the multi-time model, with high probability.*

More formally, under Assumptions 1 and 2, for any policy over options π_o*, some real valued* $\tau > 1$*,* $\varepsilon = \gamma^{\mu_{k,o} - \tau}(2\tau + 1)e^{-\beta_{min}}$*,* $\delta = \frac{\sigma^2}{\tau^2}$*, for any state* $s \in \mathcal{S}$*, with probability* $1 - \delta$*:*

$$|V^{\pi_o}_\gamma(s) - V^{\pi_o}_{\mu_k}(s)| \leq \frac{\varepsilon(1 - \gamma^{\mu_k}) + \gamma^{\mu_k}\frac{\varepsilon}{2}\text{RMAX}}{(1 - \gamma^{\mu_k})(1 - \gamma^{\mu_k} + \frac{\varepsilon}{2}\gamma^{\mu_k})}.$$

Thus, in SSPs, the value of the two models is bounded. The dominant terms in the bound are τ and $\gamma^{\mu_k - \tau}$, which roughly capture the variance over the number of time-steps taken by the option and the length of the option's execution. We highlight this dependence in the following remark:

Remark 1. *When the option's execution is nearly deterministic,* τ *is close to* 1*, and the bound collapses to* $3\gamma^{\mu_k}$*. Therefore, the bound is tightest when 1) the option/MDP is not very stochastic, and 2) the option executes for a long period of time.*

Further, the bound is quite loose; the proof of Lemma 1 uses Chebyshev's inequality, which does not sharply characterize concentration of measure, and relies on at least one other major approximation. Hence, in practice, we expect the two models to be closer; our experiments provide further support for the closeness of the two models in a variety of traditional MDPs.

Finally, for clarity, we note that the typical convergence guarantees of the Bellman Operator are preserved under ELM. The property follows naturally from the main result of Littman and Szepesvári [1996], since ELM is still a well-formed transition model, and $\gamma^{\mu_k} \in (0, 1)$:

Remark 2. *The Bellman Operator using ELM (in place of MTM) converges to a fixed point* $V^*_{\mu_k}$*.*

4 Related Work

We now discuss other relevant literature that explores options, their models, and their use in learning and planning. We concentrate only on those methods that focus on aspects of learning the model of an option (possibly in the context of a hierarchy), or propose deviations from the usual option formalism.

Most similar to our agenda are those works that change the termination condition of the option, as proposed by Harutyunyan *et al.* [2018]. In their work, the core idea is to terminate options in an "off-policy" way, enabling unification of typical off-policy TD updates and option updates. This

gives rise to a new option learning algorithm, $Q(\beta)$, that enables faster convergence by learning β in an off-policy manner. Similarly, Mankowitz *et al.* [2014] study *interrupting* options, a means of improving a given set of options during planning. Their idea is to alter a given option's predefined termination condition based on information computed during planning. In this way, options can be iteratively improved via a Bellman like update (with interruption added). They demonstrate that these new options also lead to a contraction-mapping that ensures convergence of the option value function to a fixed point. Their main contribution is to build *regularization* into this framework by encouraging their operator to choose longer options. Silver and Ciosek [2012] develop *compositional* option models, which enable recursive nesting of option models through a generalization of the Bellman operator. Our work differs from each of the above three methods in that we propose a new transition model and reward model to be used for planning and learning with options—naturally, combinations of ELM with the above variants may yield suitable algorithms for option discovery, model computation, and planning, which we leave for future work.

We also highlight the exciting, growing literature on option discovery, as explored by Şimşek and Barto [2004], Konidaris and Barto [2009], Mankowitz *et al.* [2016], Machado *et al.* [2017], and Bacon *et al.* [2017]; options for transfer, as developed by Konidaris and Barto [2007], Brunskill and Li [2014], and Topin *et al.* [2015]; and options as generalized reinforcement learning tasks [White, 2017].

ELM is in part inspired by the use of options in the context of hierarchical reinforcement learning, when estimating nested option models becomes increasingly challenging. MAXQ [Dietterich, 2000] is a classic approach to decomposing value functions of MDPs into smaller pieces, according to a task hierarchy. Considering its model-based extension, R-MAXQ [Jong and Stone, 2008], each subtask model is initially unknown and approximated via R-MAX [Brafman and Tennenholtz, 2002] under MTM, relying upon a modified Bellman update recursively dependent on its subtasks. R-MAXQ is thus akin to our experimental methodology (Section 5.1), where we employ R-MAX with MTM or ELM to guide the intra-option learning of models. An approach similar to MAXQ's task hierarchies plans instead over hierarchies of abstract Markov decision processes, or AMDPs [Gopalan *et al.*, 2017]. AMDPs act as a bridge between MAXQ and options, differing from both by treating each decision point in a hierarchical plan as a completely separate MDP, with its own state abstraction and local model of reward and transitions. In this sense, an AMDP serves an SMDP relative to the ground MDP, with its actions functioning like options; to learn an AMDP model, thus, is to learn an option model. In our experiments, we use AMDPs as the underlying representation for specifying and learning option models.

5 Experiments

We now explore the utility of ELM through experiments. The main hypothesis we investigate is how ELM compares to MTM for learning and exploiting option models in SSPs.

5.1 Methodology

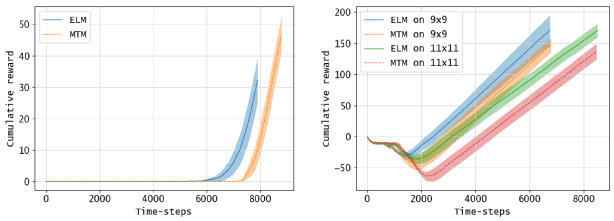
We frame each experiment as a hierarchical model-based reinforcement-learning problem. In this paradigm, an agent reasons with a collection of primitive actions and options, or a hierarchy of options. All models are initially unknown; or equivalently, the agent is only given an initiation predicate and termination probability, but no policy, $\langle I, \beta, \cdot \rangle$. Thus, the agent must estimate each option model through experience—we use R-MAX to guide learning [Brafman and Tennenholtz, 2002]. R-MAX counts transition visitations and total rewards as they are observed. Crucially, unknown transitions are treated as providing maximum reward until they become "known" by being visited beyond some m threshold. It is here that MTM and ELM differ in application: a transition under MTM requires adding and updating as many parameters as needed across all k possible time-steps, while a transition under ELM needs only update its running average, μ_k. Once a transition is known, its respective values in T and R are computed by R-MAX to be the observed totals divided by the state–action count. An option policy is then generated by running a planning algorithm in the subtask's AMDP with the R-MAX-approximated model; we use value iteration.

Our experiments each consists of 30 independent trials. Every trial, we sample a new MDP from the given domain (all MDPs in the same domain share the same actions, transition rules, and underlying representation of state space). Each MDP uses a goal-based reward function, providing the greatest reward at goal states, adhering to the properties of SSPs, and yielding the most negative reward at any failure states. A trial consists of 300 episodes, terminating at either a goal state, a failure state, or upon reaching a maximum number of steps. The AMDP hierarchies are expert-defined and, for the cited domains, are based on options or MAXQ task hierarchies in existing literature. We set $m = 5$ for the confidence parameter in R-MAX. Across all MDPs, $\gamma = 0.99$, and all transitions are stochastic with probability $4/5$ of an action "succeeding," otherwise transitioning with probability $1/5$ to a different adjacent state (as if another action had been selected).

We experiment with the following domains: Four Rooms, a small gridworld with walled rooms and hallways from Sutton *et al.* [1999]; Bridge Room, a gridworld with a large central room containing pits (failure states) spanned by a bridge, with two longer safe corridors on either side; the Taxi domain [Dietterich, 2000], for which tasks are defined by hierarchical options composed of other options; and, the discrete Playroom domain [Singh *et al.*, 2005; Konidaris *et al.*, 2018], also using a hierarchy of options, but requiring an even more complex interlaced sequence of specific actions that must be performed before reaching the goal. For more details, we refer readers to our appendix, or the original papers cited, as we follow their definitions precisely.

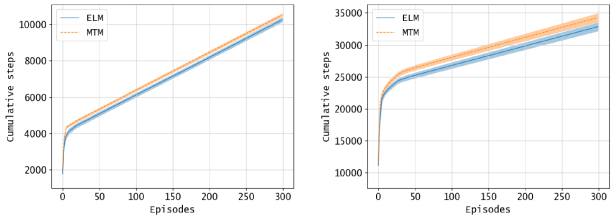
5.2 Results

We conduct experiments focusing on the speed and quality of learning ELM options models, in terms of discounted cumulative reward (performance) and time-steps (sample complexity), compared to MTM. Figures 2 and 3 present performance curves with 95% confidence intervals for the domains that we



(a) Four Rooms. (b) Bridge Room.

Figure 2: Learning flat hierarchies of option models.



(a) One passenger. (b) Two passengers.

Figure 3: Learning options for Taxi task hierarchies.

discuss shortly in more detail. Overall, we observe that ELM and MTM attain the same asymptotic performance across every example, reflecting the fact that they both eventually converge to similar value policies for each task. Further, the results suggest that ELM often requires fewer absolute samples to achieve the same behavior.

In general, we find that, with all else being equal, ELM requires fewer samples to reach near-optimal behavior. This fact is reflected by the graph of ELM terminating earlier than MTM when plotted over time-steps in Figure 2a, given both are run for a consistent number of episodes. ELM more efficiently achieves the same trend. This result reveals how, under ELM, plans reaching the goal are formed earlier, how the agent more quickly finds a good policy. Consider the difference of the value functions learned under these models (Figure 6). The image displays the error that arises from the assumption ELM makes when planning over options, relative to MTM, while reflecting some noise due to stochasticity in the domain. However, upon inspection of this and all other trials, the overall shape of the value function for ELM and MTM is approximately the same. For example, in the trial from Figure 6, both $V^*_{\mu_k}(s)$ and $V^*_{\gamma}(s)$ ramp up in value towards the upper-right corner, from the three other corners. Most importantly, despite the difference in the value functions, the policies generated from both are identical; both MTM and ELM yield the optimal policy. The end result is that, while the option models learned under MTM are correct and optimal, those learned under ELM are near-optimal but acquired sooner, while still yielding the optimal policy.

We consider results on two variants of the Bridge Room domain, grids of size 9×9 and 11×11 (Figure 2b). The joints in the graphed curves reflect when option models solidify (the majority of transitions in R-MAX become "known") In the latter figure, as with Four Rooms, we remark that ELM begins

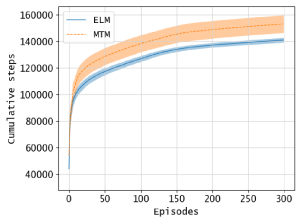


Figure 4: Taxi, three passengers.

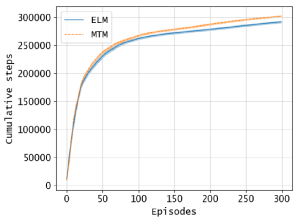


Figure 5: Playroom.

converging earlier consistently, reflecting its ability to generalize more quickly about the expected length, and thus value, of the available options. In the former, however, the results are not statistically significant, and we see here a trade-off of ELM over MTM. For this smaller domain, the bridge is short enough that ELM may randomly happen to cross it safely several times. If this event occurs, the agent learns to expect higher reward from the bridge option, negatively impacting ELM's overall performance until it eventually learns the impact of stochastically falling into a pit. Hence, the confidence interval of ELM on 9×9 in Figure 2b widens as ELM is less consistent across trials; we designed this domain precisely to exhibit this potential downside of ELM. Note that, while the ELM options here are not optimal and are subject to greater variance, the resultant policy converged to by the planning algorithm using these models *is* optimal.

For the Taxi domain, we consider the cumulative number of samples as task complexity increases from one to three passengers. For each, we discern that both learn models in relatively few episodes. In the case of one and two passengers (Figures 3a and 3b), the results are closely aligned, and the benefit of ELM over MTM is significant but minimal. For the largest task, three passengers (Figure 4), we observe similar results but draw attention to the lower variance among trials.

Figure 5 presents results, again measuring cumulative steps taken (so lower on the y-axis means faster learning) in the discrete Playroom domain. Here, the patterns manifested in the other examples recur, though the two trends diverge later

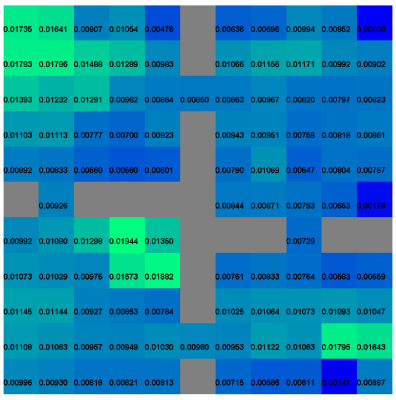


Figure 6: The difference in value between ELM and MTM for a Four Rooms task, with a goal in the upper right. Each cell reports the error, $|V^*_{\mu_k}(s) - V^*_{\gamma}(s)|$, visualized from low (blue) to high (green), where s is the state in which the agent occupies that cell.

than in the Taxi experiments. This behavior is due to the immense state–action space that must be learned for the effector-moving options, such that, even as they are being learned, we see ELM's effect—favoring expected length leads to the generation of overall shorter plans.

6 Conclusion

In this work, we propose a simpler option model, ELM. Our analysis and experiments illuminate its potential for retaining a reasonable approximation of MTM while removing the overhead in its construction. Our main theorem bounds the value difference of MTM and ELM in SSPs, and our experimental findings corroborate the claim that ELM can be a suitable replacement for MTM. Many open questions remain. First, we take the restriction to SSPs to serve as a reasonable initial constraint, but relaxing this assumption is a major direction for future work. We suspect that a nearby approximation of ELM can serve as a sufficient replacement for MTM in richer classes of MDPs. Second, we foresee a connection between ELM and the problem of *option discovery*–we speculate that finding options with simple models may serve as a useful objective for learning. For instance, inherent stochasticity leads to higher ELM error. Thus, finding options that minimize this source of error may enable quick learning of options *and* their models. Finally, further analysis may shed light on the bias-variance trade-off induced by the ELM.

Acknowledgments

The authors would like to thank Kavosh Asadi for his comments on a draft of the paper, along with Ron Parr and George Konidaris for helpful conversations, and the anonymous reviewers for their clarifying remarks and suggestions. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1426452, and by DARPA under grants W911NF-15-1-0503 and D15AP00102.

References

[Asadi *et al.*, 2018] Kavosh Asadi, Dipendra Misra, and Michael L Littman. Lipschitz continuity in model-based reinforcement learning. *ICML*, 2018.

[Bacon *et al.*, 2017] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, pages 1726–1734, 2017.

[Brafman and Tennenholtz, 2002] Ronen I Brafman and Moshe Tennenholtz. R-MAX: A general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*, 3(Oct):213–231, 2002.

[Brunskill and Li, 2014] Emma Brunskill and Lihong Li. PAC-inspired option discovery in lifelong reinforcement learning. In *ICML*, pages 316–324, 2014.

[Dietterich, 2000] Thomas G Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *JAIR*, 13:227–303, 2000.

[Fruit and Lazaric, 2017] Ronan Fruit and Alessandro Lazaric. Exploration–exploitation in MDPs with options. In *AISTATS*, pages 576–584, 2017.

[Gopalan *et al.*, 2017] Nakul Gopalan, Marie desJardins, Michael L Littman, James MacGlashan, Shawn Squire, Stefanie Tellex, John Winder, and Lawson LS Wong. Planning with abstract Markov decision processes. In *ICAPS*, 2017.

[Harutyunyan *et al.*, 2018] Anna Harutyunyan, Peter Vrancx, Pierre-Luc Bacon, Doina Precup, and Ann Nowé. Learning with options that terminate off-policy. In *AAAI*, 2018.

[Jong and Stone, 2008] Nicholas K Jong and Peter Stone. Hierarchical model-based reinforcement learning: R-MAX+MAXQ. In *ICML*, pages 432–439. ACM, 2008.

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

[Konidaris and Barto, 2007] George Konidaris and Andrew G Barto. Building portable options: Skill transfer in reinforcement learning. In *IJCAI*, 2007.

[Konidaris and Barto, 2009] George Konidaris and Andrew G Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *NeurIPS*, pages 1015–1023, 2009.

[Konidaris *et al.*, 2018] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *JAIR*, 61:215–289, 2018.

[Littman and Szepesvári, 1996] Michael L Littman and Csaba Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *ICML*, volume 96, pages 310–318, 1996.

[Littman *et al.*, 1995] Michael L Littman, Thomas L Dean, and Leslie Pack Kaelbling. On the complexity of solving Markov decision problems. In *UAI*, pages 394–402, 1995.

[Machado *et al.*, 2017] Marlos C Machado, Marc G Bellemare, and Michael Bowling. A Laplacian framework for option discovery in reinforcement learning. *ICML*, 2017.

[Mankowitz *et al.*, 2014] Daniel J Mankowitz, Timothy A Mann, and Shie Mannor. Time-regularized interrupting options. In *ICML*, 2014.

[Mankowitz *et al.*, 2016] Daniel J Mankowitz, Timothy A Mann, and Shie Mannor. Adaptive skills adaptive partitions (ASAP). In *NeurIPS*, pages 1588–1596, 2016.

[Mann and Mannor, 2013] Timothy A Mann and Shie Mannor. The advantage of planning with options. *RLDM 2013*, page 9, 2013.

[Mann and Mannor, 2014] Timothy Mann and Shie Mannor. Scaling up approximate value iteration with options: Better policies with fewer iterations. In *ICML*, 2014.

[Parr, 1998] Ronald Edward Parr. *Hierarchical Control and Learning for Markov Decision Processes*. PhD thesis, University of California, Berkeley, 1998.

[Precup and Sutton, 1997] Doina Precup and Richard S Sutton. Multi-time models for reinforcement learning. In

ICML Workshop on Modelling in Reinforcement Learning, 1997.

[Precup and Sutton, 1998] Doina Precup and Richard S Sutton. Multi-time models for temporally abstract planning. In *NeurIPS*, pages 1050–1056, 1998.

[Puterman, 2014] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[Silver and Ciosek, 2012] David Silver and Kamil Ciosek. Compositional planning using optimal option models. In *ICML*, volume 2, pages 1063–1070, 2012.

[Şimşek and Barto, 2004] Özgür Şimşek and Andrew G Barto. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *ICML*, page 95. ACM, 2004.

[Singh *et al.*, 2005] Satinder P Singh, Andrew G Barto, and Nuttapong Chentanez. Intrinsically motivated reinforcement learning. In *NeurIPS*, pages 1281–1288, 2005.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[Sutton *et al.*, 1999] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

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

[Talvitie, 2017] Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *AAAI*, pages 2597–2603, 2017.

[Topin *et al.*, 2015] Nicholay Topin, Nicholas Haltmeyer, Shawn Squire, John Winder, James MacGlashan, and Marie desJardins. Portable option discovery for automated learning transfer in object-oriented Markov decision processes. In *IJCAI*, 2015.

[Tseng, 1990] Paul Tseng. Solving H-horizon, stationary Markov decision problems in time proportional to log(H). *Operations Research Letters*, 9(5):287–297, 1990.

[White, 2017] Martha White. Unifying task specification in reinforcement learning. In *ICML*, pages 3742–3750, 2017.