Hidden Parameter Markov Decision Processes: An Emerging Paradigm for Modeling Families of Related Tasks

George Konidaris[*]
Duke University
Durham, NC 27708
gdk@cs.duke.edu

Finale Doshi-Velez[*]
Harvard Medical School
Boston, MA 02115
finale@alum.mit.edu

Introduction

The goal of transfer is to use knowledge obtained by solving one task to improve a robot's (or software agent's) performance in future tasks. In general, we do not expect this to work; for transfer to be feasible, there must be something in common between the source task(s) and goal task(s). The question at the core of the transfer learning enterprise is therefore: *what makes two tasks related?*, or more generally, *how do you define a family of related tasks?* Given a precise definition of how a particular family of tasks is related, we can formulate clear optimization methods for selecting source tasks and determining what knowledge should be imported from the source task(s), and how it should be used in the target task(s).

This paper describes one model that has appeared in several different research scenarios where an agent is faced with a family of tasks that have similar, but not identical, dynamics (or reward functions). For example, a human learning to play baseball may, over the course of their career, be exposed to several different bats, each with slightly different weights and lengths. A human who has learned to play baseball well with one bat would be expected to be able to pick up any similar bat and use it. Similarly, when learning to drive a car, one may learn in more than one car, and then be expected to be able to drive any make and model of car (within reasonable variations) with little or no relearning. These examples are instances of exactly the kind of flexible, reliable, and sample-efficient behavior that we should be aiming to achieve in robotics applications.

One way to model such a family of tasks is to posit that they are generated by a small set of latent parameters (e.g., the length and weight of the bat, or parameters describing the various physical properties of the car's steering system and clutch) that are fixed for each problem instance (e.g., for each bat, or car), but are not directly observable by the agent. Defining a distribution over these latent parameters results in a family of related tasks, and transfer is feasible to the extent that the number of latent variables is small, the task dynamics (or reward function) vary smoothly with them, and to the extent to which they can either be ignored or identified using transition data from the task. This model has appeared under several different names in the literature; we refer to it as a *hidden-parameter Markov decision process* (or HIP-MDP).

HIP-MDPs

A single Markov decision process is defined by a tuple $\langle S, A, R, T, \gamma \rangle$, where S is a state space, A is a set of available actions, $R(s, a, s')$ defines the reward obtained when executing action a in state s and transitioning to state s', $T(s, a, s')$ defines the probability of arriving in state s' given that action a has been executed at state s, and $\gamma \in (0, 1]$ is a discount factor expressing the extent to which immediate rewards are preferred over future ones.

HIP-MDPs model a distribution of tasks where the variation in the dynamics (or the reward function, though we do not discuss that case here) across the family of tasks can be captured by a set of hidden parameters, θ, encountered with probability P_θ. Consequently, the transition function T is conditioned on θ: $T(s, a, s'|\theta)$. Thus, a HIP-MDP describes a *class* of tasks, and any setting $\theta_b \sim P_\theta$ results in a single *task instance* MDP. The values of θ_b are not observed.[1] We also assume that θ_b is drawn *once*, at the beginning of the task instance, that it does not change until the beginning of the next instance, and that the agent is made aware that a change has occurred.

Properties

Some early publications (Fern and Tadepalli 2010; Bai, Hsu, and Lee 2013a) using a HIP-MDP model emphasized its similarity to the partially-observable Markov decision process, or POMDP, model (Kaelbling, Littman, and Cassandra 1998). Here, the hidden θ_b values are likened to the hidden state of the POMDP, and the state observed during an individual task is similar to an observation in a POMDP. In particular, Bai, Hsu, and Lee (2013a) showed that we can use a POMDP planner to produce a policy for solving tasks drawn from a known HIP-MDP. Such a planner will naturally and automatically trade off information-gathering actions that actively determine the value of θ_b with reward seeking actions.

[*] Both authors are primary authors on this occasion.

[1] If the values of θ_b were observed, we have a parametrized skill learning problem (da Silva, Konidaris, and Barto 2012).

However, HIP-MDPs have some useful properties which suggest that they merit study in their own right. One is that **any specific task instance is an MDP, and can be solved independently as one**. More specifically, given some fixed (but unknown) task parameters θ_b, the state space S is Markov. When solving one task instance, we need not necessarily worry about the overall family of tasks. This allows us to model scenarios where the source tasks for transfer learning can be solved directly by reinforcement learning.

Moreover, **the hidden parameters θ are constant for each task** (i.e., the unobserved variables have no dynamics). We also assume that new task instances are drawn from P_θ independently from previous task instances. These properties allow us to gather large batches of transition data that correspond to fixed settings of θ, easing model learning.

Finally, **the latent parameters are a sufficient statistic for specifying an individual task** (given that you are solving a problem drawn from this family of tasks). Consequently, if we have a model of the family of tasks, learning reduces to determining the values of θ for the current task (Bai, Hsu, and Lee 2013a). Once these have been identified, we can simply compute a policy for the resulting MDP and use it. Thus, if we have a model of the HIP-MDP, we can expend great computational effort offline to solve various task instances, and then synthesize a parameterized policy (da Silva, Konidaris, and Barto 2012) to be deployed given the agent's belief over θ_b at runtime. Online execution will then correspond to filtering over θ_b and swapping in the appropriate policy. This type of approach will be most useful when the number of task parameters $|\theta|$ is very much smaller than the size of the state space S, and when observed transitions allow us to determine θ_b quickly.

Existing Research Using HIP-MDPs

Whiteson et al. (2011) proposed the notion of a *generalized environment*—where a distribution of environments is generated using a set of latent parameters and a distribution over those parameters—as a means of constructing *generalized methodologies* for avoiding environment overfitting, which is often problematic in reinforcement learning research. They report experimental results using generalized versions of the common Mountain Car, Acrobot, and Puddle World benchmarks, and point out that several of the reinforcement learning competitions (Whiteson, Tanner, and White 2010) have used such generalized environments, including generalized helicopter control (Koppejan and Whiteson 2009).

Bai, Hsu, and Lee (2013a) used a POMDP planner to precompute a policy that naturally and automatically trades off exploration and exploitation when planning for an uncertain model. One of their domains is a HIP-MDP: the Acrobot problem where one of the robot arms has an unknown mass, drawn from a fixed distribution. The goal of planning in this case is to produce a single policy capable of solving all members of the class of tasks. This is accomplished by treating the HIP-MDP as a POMDP and planning in belief space—in this case the joint space of observed state and sufficient statistics of the distribution over latent variables—so that the plan naturally trades off information-gathering and reward-maximizing actions. A similar approach is taken in Bai, Hsu, and Lee (2013b).

Doshi-Velez and Konidaris (2013) learn a HIP-MDP model given experience obtained while learning to solve several task instances. The effect of the latent parameters are modeled using semi-parametric regression, inferring both the number of latent parameters and their effect on the transition function. They learn models for the Acrobot with two unknown parameters (the masses of each arm) and the cart-pole problem with a pole of unknown length and mass.

Wilson, Fern, and Tadepalli (2012) introduced hierarchical model-based transfer, where a family of tasks is generated through a task class prior which in turn generates a distribution of models. This can be considered a hierarchical HIP-MDP, where θ_b is used to generate a focused distribution over MDPs (rather than a single MDP) from which an observed MDP is drawn. Their method was used to successfully transfer policies across a class of colored mazes and a real-time strategy game.

Several researchers have explored problems that can be modeled as HIP-MDPs with a single, discrete hidden variable, often called a *type*. Each task instance is then one of a fixed number of MDPs, but the MDP identity is not observed. Mahmud et al. (2014) consider the problem of personalizing user-interfaces to users with different skill levels, and use Bayesian model selection to find a regret-minimizing solution. Here, the user's behavior is the MDP, and the hidden parameters correspond to user skill type; the learning agent changes the interface presented to the user to better match their type. Rosman (2014) presents a Bayesian policy reuse algorithm, where several domains (a golf club selection problem, a telephone interface personalization system, a PacMan agent, and a surveillance domain) are modeled as HIP-MDPs with hidden types. Fern and Tadepalli; Fern et al. (2010; 2014) use a hidden type variable in the context of an interactive assistant which must infer a user's hidden goals. Azar, Lazaric, and Brunskill (2013) and Brunskill and Li (2013) consider the transfer problem across a finite number of multi-armed bandits.

Most recently, Ammar et al. (2014) describe a method for transferring learned policy parameters across families of related tasks using policy gradient methods. Rather than attempting to infer the hidden parameters, they synthesize a new policy as a linear combination of policies obtained by solving previous task instances. This paper uses four different domains, each of which is a HIP-MDP: a spring-mass damper system with three unknown variables; cart-pole with unknown mass, cart length, and damping parameter; three-link inverted pendulum with unknown cart mass and mass, length, inertia, and damping parameters for each link; and a quad rotor system with unknown arm length and inertia around the x, y, and z axes. Each domain had a fixed range from which each parameter was drawn uniformly at random, generating a distribution of related tasks across which transfer is affected.

HIP-MDPs for Model Uncertainty in Robotics

The majority of the papers we have cited in the preceding section use a particular type of HIP-MDP: a physical model has some parameters (e.g., the mass of a link, or the inertia around an axis), and these parameters are unknown to the agent in advance. This kind of physics-based prior (Scholz et al. 2014), where physical quantities are modeled as latent variables, is of particular relevance to robotics.

In general, it is reasonable for us to assume that we have a good model of our robot, since we either built it ourselves or can demand such a model from the company we bought it from. However, when our robots must operate in unstructured or semi-structure environments, it is almost never reasonable to expect a perfect model of the rest of the world. Nevertheless, it is also unreasonable to assume *no* knowledge of how the world works. A robot encountering a novel ball cannot expect to have a good model of how that particular ball works, but it should be able to use general knowledge of how balls move to drastically simplify the model estimation problem that it faces.

Approaches based on HIP-MDPs can thereby serve as a middle-ground between the reinforcement learning (Sutton and Barto 1998) setting, where we assume little or no knowledge of the world dynamics, and the motion planning (LaValle 2006) setting, where the environmental dynamics are always precisely specified. In such a setting, transfer learning has a very direct and important application: how can we generalize knowledge about the world, in terms of physical models and their latent variables, to be able to very rapidly (or even virtually instantly) achieve general competence when interacting with novel objects and environments?

In robotics (and many other applications), the assumption that the agent has repeated interaction with *exactly the same problem*, down to the smallest detail, over and over again, is unrealistic. Similarly, the assumption that the robot is gifted an accurate model of the environment is also unrealistic. Modeling the natural variations in task dynamics and reward functions via HIP-MDPs offers a natural approach to precisely defining the goal of transfer in these scenarios, and thereby provides a route to principled and practical transfer algorithms that could substantially improve robustness and autonomy in real robot applications.

References

Ammar, H.; Eaton, E.; Ruvolo, P.; and Taylor, M. 2014. Online multi-task learning for policy gradient methods. In *Proceedings of the 31st International Conference on Machine Learning*, 1206–1214.

Azar, M.; Lazaric, A.; and Brunskill, E. 2013. Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems 26*, 2220–2228.

Bai, H. Y.; Hsu, D.; and Lee, W. S. 2013a. Planning how to learn. In *International Conference on Robotics and Automation*, 2853–2859.

Bai, H.; Hsu, D.; and Lee, W. 2013b. Integrated perception and planning in the continuous space: A POMDP approach. In *Proceedings of Robotics: Science and Systems*, 18–25.

Brunskill, E., and Li, L. 2013. Sample complexity of multi-task reinforcement learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.

da Silva, B.; Konidaris, G.; and Barto, A. 2012. Learning parameterized skills. In *Proceedings of the Twenty Ninth International Conference on Machine Learning*, 1679–1686.

Doshi-Velez, F., and Konidaris, G. 2013. Hidden parameter Markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. ArXiV preprint arXiv:1308.3513.

Fern, A., and Tadepalli, P. 2010. A computational decision theory for interactive assistants. In *Advances in Neural Information Processing Systems 23*, 577–585.

Fern, A.; Natarajan, S.; Judah, K.; and Tadepalli, P. 2014. A decision-theoretic model of assistance. *Journal of Artificial Intelligence Research* 50:71–104.

Kaelbling, L.; Littman, M.; and Cassandra, A. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Koppejan, R., and Whiteson, S. 2009. Neuroevolutionary reinforcement learning for generalized helicopter control. In *GECCO 2009: Proceedings of the Genetic and Evolutionary Computation Conference*, 145–152.

LaValle, S. 2006. *Planning Algorithms*. Cambridge University Press.

Mahmud, M.; Rosman, B.; Ramamoorthy, S.; and Kohli, P. 2014. Adapting interaction environments to diverse users through online action set selection. In *Proceedings of the AAAI 2014 Workshop on Machine Learning for Interactive Systems*.

Rosman, B. 2014. *Learning Domain Abstractions for Long Lived Robots*. Ph.D. Dissertation, University of Edinburgh. Chapter 4.

Scholz, J.; Levihn, M.; Isbell, C.; and Wingate, D. 2014. A physics-based model prior for object-oriented MDPs. In *Proceedings of the 31st International Conference on Machine Learning*, 1089–1097.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Whiteson, S.; Tanner, B.; Taylor, M.; and Stone, P. 2011. Protecting against evaluation overfitting in empirical reinforcement learning. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*.

Whiteson, S.; Tanner, B.; and White, A. 2010. The reinforcement learning competitions. *AI Magazine* 31(2):81–94.

Wilson, A.; Fern, A.; and Tadepalli, P. 2012. Transfer learning in sequential decision problems: A hierarchical Bayesian approach. In *Proceedings of the ICML 2011 Workshop on Unsupervised and Transfer Learning*, 217–227.