Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Robust and Efficient Transfer Learning with Hidden Parameter Markov Decision Processes

Taylor W. Killian
Harvard University
Cambridge, MA 02148
taylorkillian@g.harvard.edu

George Konidaris
Brown University
Providence, RI 02912
gdk@cs.brown.edu

Finale Doshi-Velez
Harvard University
Cambridge, MA 02148
finale@seas.harvard.edu

Abstract

An intriguing application of transfer learning emerges when tasks arise with similar, but not identical, dynamics. Hidden Parameter Markov Decision Processes (HiP-MDP) embed these tasks into a low-dimensional space; given the embedding parameters one can identify the MDP for a particular task. However, the original formulation of HiP-MDP had a critical flaw: the embedding uncertainty was modeled independently of the agent's state uncertainty, requiring an arduous training procedure. In this work, we apply a Gaussian Process latent variable model to jointly model the dynamics and the embedding, leading to a more elegant formulation, one that allows for better uncertainty quantification and thus more robust transfer.

Motivation

With the prevalence of systems with similar, but not identical, processes (e.g. healthcare, sensing networks, robotics) there is a compelling need to develop learning frameworks that account for system variations in an efficient and robust manner. These variations in both unobserved and observed representations of the system can contribute to inefficiencies or, in some dramatic cases, failure of an agent's ability to learn an optimal control policy. In order to develop optimal control policies, it is undesirable and ineffectual to start afresh each time a new instance is encountered. Ideally, an agent will leverage the similarities across separate, but related, instances. This paradigm of learning introduces an intriguing use case for transfer learning.

The Hidden Parameter Markov Decision Process (HiP-MDP) (Doshi-Velez and Konidaris 2013), was introduced as a formalization of these domains with two primary features. First, that a bounded number of latent parameters, w, for a single task instance can fully specify the system dynamics $\theta \in \Theta$, the set of all parameter variations with prior P_Θ, if learned. That is, the dynamics dictating the transition between states can be expressed as $T(s'|s, a, \theta_b)$ for instance b. Second, that the system dynamics will not change during a task and an agent would be capable of determining when a change occurs. The HiP-MDP is described by the tuple: $\{S, A, \Theta, T, R, \gamma, P_\Theta\}$, where S and A are the sets of states and actions respectively, $R(s, a)$ is the function mapping the utility of taking action a in state s. Thereby, the HiP-MDP describes a *class* of tasks; where particular instances of that class are obtained by independently sampling some θ_b at the initiation of each task instance b.

The original HiP-MDP had a transition model of the form:

$$(s'_d - s_d) \sim \sum_k^K z_{kad} w_{kb} f_{kad}(s) + \epsilon$$
$$\epsilon \sim \mathcal{N}(0, \sigma^2_{nad})$$

which sought to learn weights w_{kb} based on the k^{th} latent factor corresponding to task instance b, filter parameters $z_{kad} \in \{0, 1\}$ denoting whether the k^{th} latent parameter is relevant in predicting dimension d when taking action a as well as task specific basis functions f_{kad} drawn from a Gaussian Process (GP).

Doshi-Velez and Konidaris show that the HiP-MDP is able to rapidly identify the dynamics T of a new task instance and adapt to the variations therein. However, that formulation had a critical flaw: the embedding uncertainty of the latent parameter space was modeled independently from the agent's state uncertainty. This requires all tasks to have the ability to visit every part of the state space, which is not guaranteed to be feasible in most systems.

A HiP-MDP with Joint Uncertainty

We present an alternative formulation to the original HiP-MDP that embeds the latent parametrization in the observed data via a Gaussian Process latent variable model (GPLVM) (Lawrence 2004).This approach creates a unified GP model, with the augmented state $\tilde{s} =: [s^\intercal,\ a,\ w_b]^\intercal$ as input, for both inferring the transition dynamics (Wang, Fleet, and Hertzmann 2005) within a task instance but also in the transfer between task instances (Cao et al. 2010). The approximated transition model then takes the form of:

$$s'_d \sim f_d(\tilde{s}) + \epsilon$$
$$f_d \sim GP(\psi)$$
$$w_b \sim \mathcal{N}(\mu_b, \Sigma_b)$$
$$\epsilon \sim \mathcal{N}(0, \sigma_{bd})$$

This approach enables the HiP-MDP to flexibly infer the dynamics T of a new instance by virtue of the statistical similarities found in the learned covariance function between observed states of the new instance and those from prior instances. Another feature of formulating the HiP-MDP after

this fashion is that we are able to leverage the marginal log likelihood of the GP to optimize the weight distribution and thereby quantify the uncertainty (Candela et al. 2003) of the latent embedding of w_b for θ_b. These two features of reformulating the HiP-MDP as a GPLVM allows for more robust and efficient transfer.

Parameter Learning and Updates We deploy the HiP-MDP when the agent is provided batch observational data from several task instances and asked to quickly learn optimal control policies for new instances in an online fashion. With this observational data, the GP transition functions f_d are learned and the individual weighting distributions for w_b are periodically optimized. To streamline the approximation of T we choose a set of support points s^* from S that sparsely approximate the full GP, which are also periodically updated as the latent weighting distribution is updated. Procedures exist to select these support points accurately (Snelson and Ghahramani 2005), we however heuristically select these points to minimize the maximum reconstruction error within each batch using simulated annealing. The outcome of this online learning method proves to be robust to the choice of initial parametrization.

Control Policy A control policy is learned for each task instance b following the procedure outlined in (Deisenroth and Rasmussen 2011) where a set of tuples $(s,\ a,\ s',\ r)$ are observed and the policy is periodically updated (as is the latent embedding w_b) in an online fashion, leveraging the approximate dynamics of T via the f_d^* to create synthetic observations from the current instance. The policy is updated via a Double Deep Q Network using prioritized experience replay (van Hasselt, Guez, and Silver 2016), (Schaul et al. 2015).

Multiple episodes are run from each instance b to further optimize the policy over the hidden parameter setting θ_b. After doing so, the hyperparameters of the GP defining the f_d are updated before encountering another randomly manifest task instance.

Demonstration We demonstrate an example (see Figure 1) of a toy domain where an agent is able to learn separate policies according to a hidden latent parameter. Instances inhabiting a "blue" latent parametrization can only pass through to the goal region over the blue boundary while those with a "red" parametrization can only cross the red boundary. After a few training instances, the HiP-MDP is able to separate the two latent classes and develops individualized policies for each. Due to the flexibility enabled by embedding the latent parametrization into the system's state, the GPLVM identifies which class the current instance belongs to within the first couple of training episodes. In total, this example took approximately 30 minutes to develop optimal policies for 20 task instances. We place an unclassified survey point in the top left quadrant to gather information about the policy uncertainty given the two latent classes: we see that there is larger uncertainty associated when the survey point is associated with the "red" parametrization in comparison with the "blue" parametrization. This indicates the ability of the HiP-MDP to provide coherent inference across multiple latent classes present in the observed task, affirming the motivation for jointly modeling the dynamics and latent embedding for the transfer between related task instances.

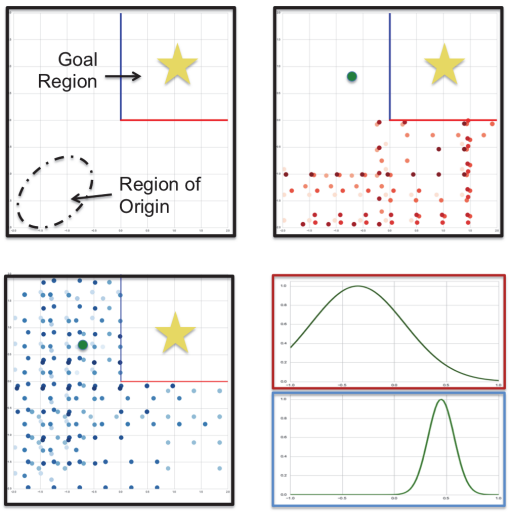

Figure 1: Toy Problem: (a) Schematic outlining the domain, (b) learned policy for "red" parametrization, (c) learned policy for "blue" parametrization, (d) uncertainty measure for input point according to separate latent classes.

References

Candela, J. Q.; Girard, A.; Murray-Smith, R.; and Rasmussen, C. 2003. Propagation of uncertainty in bayesian kernel models - application to multiple-step ahead time series forecasting. In *Proceedings of the ICASSP*, 701–704.

Cao, B.; Pan, S.; Zhang, Y.; Yeung, D.; and Yang, Q. 2010. Adaptive transfer learning. In *AAAI*, volume 2, 7.

Deisenroth, M., and Rasmussen, C. 2011. Pilco: A model-based and data-efficient approach to policy search. In *In Proceedings of the International Conference on Machine Learning*.

Doshi-Velez, F., and Konidaris, G. 2013. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. *CoRR* abs/1308.3513.

Lawrence, N. 2004. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems*, volume 16, 329–336.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Snelson, E., and Ghahramani, Z. 2005. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 17, 1257–1264.

van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Wang, J.; Fleet, D.; and Hertzmann, A. 2005. Gaussian Process Dynamical Models. In *Advances in Neural Information Processing Systems*, volume 17, 1441–1448.