Active Learning of Parameterized Skills

Bruno Castro da Silva BSILVA@CS.UMASS.EDU
School of Computer Science, University of Massachusetts Amherst, MA 01003

George Konidaris GDK@CSAIL.MIT.EDU
Computer Science and Artificial Intelligence Lab, MIT, Cambridge, MA 02139

Andrew Barto BARTO@CS.UMASS.EDU
School of Computer Science, University of Massachusetts Amherst, MA 01003

Abstract

We introduce a method for actively learning parameterized skills. Parameterized skills are flexible behaviors that can solve any task drawn from a distribution of parameterized reinforcement learning problems. Approaches to learning such skills have been proposed, but limited attention has been given to identifying which training tasks allow for rapid skill acquisition. We construct a non-parametric Bayesian model of skill performance and derive analytical expressions for a novel acquisition criterion capable of identifying tasks that maximize expected improvement in skill performance. We also introduce a spatiotemporal kernel tailored for non-stationary skill performance models. The proposed method is agnostic to policy and skill representation and scales independently of task dimensionality. We evaluate it on a non-linear simulated catapult control problem over arbitrarily mountainous terrains.

1. Introduction

One approach to dealing with high-dimensional control problems is to specify or discover hierarchically structured policies. The most widely used hierarchical reinforcement learning formalism is the options framework (Sutton et al., 1999), where *options* (or *skills*) define temporally-extended policies that abstract away details of low-level control. One of the motivating principles underlying this framework is the idea that subproblems recur, so that options can be reused in a variety of related tasks or contexts. Options, however, are typically defined as single policies and cannot be used to solve ensembles of related reinforcement learning problems. To address this issue, *parameterized skills* have recently emerged as a promising framework for producing reusable behaviors that can solve a *distribution* of related control problems, given only a parameterized description of a task (Kober et al., 2012; da Silva et al., 2012; Neumann et al., 2013; Deisenroth et al., 2014). Once acquired, parameterized skills can produce—on-demand—policies for any tasks in the distribution, even those which the agent has never had direct experience with. As an example, consider a soccer-playing agent tasked with learning a kicking skill parameterized by desired force and direction. Learning a single policy for each possible variation of the task is infeasible. The agent might wish, instead, to learn policies for a few specific kicks and use them to synthesize a single general skill for kicking the ball—parameterized by force and direction—that it can execute on-demand.

In this paper we are concerned with the question of how an agent, tasked with learning a parametrized skill and given a distribution from which future tasks will be drawn, should practice. Such an agent should choose to practice tasks that lead it to maximize skill performance improvement over all tasks in the target distribution. Intuitively, the tasks from which experience is most beneficial are those that allow the skill to better generalize to a wider range of related problems. As we will show, identifying such tasks is not straightforward—sampling tasks according to the target distribution, for instance, is inefficient because it does not account for the varying difficulty of tasks. Furthermore, non-adaptive sampling strategies ignore how some tasks may require policies qualitatively different from those of neighboring tasks, thus demanding more extensive training. Continuing with the previous example, a soccer-playing agent may wish to learn a parameterized kicking skill by practicing different types of kicks, such as accurate ones

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

towards the goal or powerful but inaccurate kicks in the general direction of the opponent's half field. Policies for the former type of kick are harder to learn and generalize, since their parameters are very sensitive to the desired direction. Carefully choosing which kicks to practice allows the agent to identify which ones require less practice and which ones are more challenging, thus focusing on the aspects of the skill that can more readily be improved.

These observations are consistent with recent theories of how human experts acquire professional levels of achievement, which propose that skill improvement involves deliberate efforts to change particular aspects of performance (Ericsson, 2006). Indeed, thoughtful and deliberate practice is one of the defining characteristics of expert performance in sports, arts and science: world-class athletes, for instance, carefully construct training regimens to build on their strengths and shore up their weaknesses.

We construct a non-parametric Bayesian model of skill performance and derive analytical expressions for a novel acquisition criterion capable of identifying tasks that maximize skill performance improvement over a given target distribution. We also introduce a spatiotemporal kernel tailored for modeling non-stationary skill performance functions. The proposed method is agnostic to policy and skill representation and scales independently of task dimensionality. We evaluate it on a non-linear simulated catapult control problem in which different launch profiles are required depending on the target position and on the smoothness characteristics of the terrain.

2. Active Learning of Parameterized Skills

Assume an agent presented with a sequence of tasks drawn from some task distribution. Each task is modeled as a Markov Decision Process (MDP). Furthermore, assume that the MDPs have dynamics and reward functions similar enough so that they can be considered variations of a same task. The objective of a parameterized skill is to maximize the expected reward over the distribution of MDPs:

$$\int P(\tau) J\big(\pi_\theta, \tau\big) d\tau, \tag{1}$$

where π_θ is a policy with parameters $\theta \in \mathbb{R}^M$, τ is a task parameter vector drawn from a continuous space T, $J(\pi, \tau)$ is the expected return obtained when using policy π to solve task τ (given an initial state distribution), and $P(\tau)$ is a probability density function describing the probability of task τ occurring.

A *parameterized skill* is a function $\Theta : T \rightarrow \mathbb{R}^M$ mapping task parameters to policy parameters. When using a parameterized skill to solve a given task, the parameters of the policy to be used are specified by Θ. An efficient parameterized skill maximizes the expected performance of the policies it specifies over the entire distribution of possible tasks:

$$\int P(\tau) J\big(\pi_{\boldsymbol{\Theta}(\tau)}, \tau\big) d\tau. \tag{2}$$

A parameterized skill can be learned via a set of training tasks and their corresponding policies. The simplest strategy for constructing this set is to select tasks uniformly at random or to draw them from the task distribution P. These strategies, however, ignore how a carefully-chosen task can improve performance not only on that task, but over a wider range of related tasks.

We introduce a general framework for active task selection with arbitrary parameterized skill representations. The framework uses a Bayesian model of skill performance and a specially tailored acquisition function designed to select training tasks that maximize expected skill performance improvement. The process of actively selecting training tasks consists of the following steps: *1)* identify, by means of a model of expected skill performance, the most promising task τ to practice; *2)* practice the task and learn a corresponding policy π^*_θ; *3)* update the parameterized skill; *4)* evaluate the performance $J(\tau)$ of the updated skill at that task; and *5)* update the model of expected skill performance with the observed performance. These steps are repeated until the skill cannot be further improved or a maximum number of practicing episodes is reached. This training process is depicted in Figure 1.

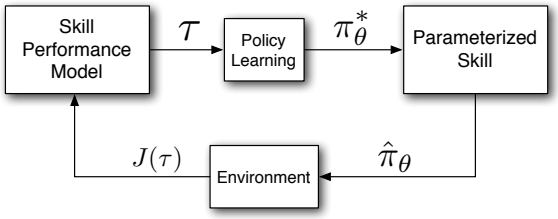

Figure 1. The process of actively learning a parameterized skill.

3. A Bayesian Model of Skill Performance

Let $J\big(\pi_{\boldsymbol{\Theta}(\tau)}, \tau\big)$ be the performance achieved by a parameterized skill $\boldsymbol{\Theta}$ on task τ. To simplify notation we suppress the dependence on $\boldsymbol{\Theta}$ and write simply $J(\tau)$. We propose creating a Bayesian model capable of predicting the expected skill performance over a given set of tasks. This model allows the agent to predict performance—even at tasks which were never directly experienced—without incurring the costs of executing the skill. As will be shown in Section 4, this is particularly useful when estimating how different training tasks may contribute to improving the performance of a skill over a distribution of tasks. Let $\mu(\tau)$ be the mean predicted performance and $\sigma^2(\tau)$ the cor-

responding variance. One way to represent this model is via a Gaussian Process (GP) prior (Rasmussen & Williams, 2006). A GP is a (possibly infinite) set of random variables, any finite set of which is jointly Gaussian distributed. GPs extend Gaussian distributions to the case of infinite dimensionality, and thus can be seen as distributions over functions—in this case, predictive distributions over skill performance functions. To fully specify a GP one must define a mean function m and a positive-definite kernel k:

$$m(\tau) = E[J(\tau)]$$

$$k(\tau_p, \tau_q) = E\big[(J(\tau_p) - m(\tau_p))(J(\tau_q) - m(\tau_q))^\top\big].$$

A kernel specifies properties of J such as smoothness and periodicity. In this section and the following we do not make any assumptions about the form of k; in Section 5 we introduce a kernel designed specifically for use with our framework. We assume that the performance function J is zero-mean, so that m can be dropped from the equations; the extension for the non-zero mean case is straightforward.

Given k and a set of observations $D = \{(\tau_1, J(\tau_1)), \ldots, (\tau_N, J(\tau_N))\}$, both the log likelihood of the data conditioned on the model and the Gaussian posterior over skill performance, $P(J(\tau)|\tau, D)$, can be computed easily for any tasks τ. Under a Gaussian Process, the predictive posterior over skill performance at a task τ is normally-distributed according to $J(\tau) \sim \mathcal{N}(\mu(\tau), \sigma^2(\tau))$, where

$$\mu(\tau) = \mathbf{k}^\top \big(\mathbf{C_D} + \sigma^2 I\big)^{-1} \mathbf{y_D} \quad (3)$$

$$\sigma^2(\tau) = k(\tau, \tau) + \sigma^2 - \mathbf{k}^\top \big(\mathbf{C_D} + \sigma^2 I\big)^{-1} \mathbf{k}, \quad (4)$$

with $\mathbf{k} = [k(\tau, \tau_1) \ldots k(\tau, \tau_N)]^\top$, $\mathbf{C_D}$ is an $N \times N$ matrix with entries $(\mathbf{C_D})_{ij} = k(\tau_i, \tau_j)$, $\mathbf{y_D} = [J(\tau_1) \ldots J(\tau_N)]^\top$ and σ^2 is the additive noise we assume affects measurements of skill performance.

GPs are often used in Bayesian optimization to find the maximum of unknown, expensive-to-sample functions. To do so, a surrogate *acquisition function* is maximized. Acquisition functions guide the search for the optimum of a function by identifying the most promising points to sample. Standard acquisition functions include Lower Confidence Bounds, Maximum Probability of Improvement and Expected Improvement (Snoek et al., 2012). Although these criteria may seem, at first, appropriate for selecting training tasks, that is not the case: if applied to a skill performance function, standard acquisition criteria would only identify the tasks that can be solved with highest performance. By contrast, our goal is to identify training tasks which result in the highest *expected improvement in skill performance*. This goal, formally defined in Equation 6, measures how much skill performance may improve *over a target distribution of tasks* if the skill is updated by the practice of a selected task. A motivating principle behind this objective is that since parameterized skills naturally generalize policies to related tasks, an effective acquisition criterion should focus not on improving the performance at a single task, but on the performance gains that practicing it may bring to the skill as a whole.

In the next section we introduce a novel acquisition criterion specially tailored for parameterized skills. We also derive closed-form expressions for this criterion and its gradient, thus obtaining analytical expressions for the two quantities required for optimizing the process of selecting training tasks.

4. Active Selection of Training Tasks

An acquisition function to identify tasks that provide *maximum expected improvement in skill performance* should: *1)* take into account that tasks may occur with different probabilities and prioritize training accordingly; and *2)* model how the practice of a single task may, due to the generalization properties inherent to a parameterized skill, improve performance not only at that task but also over related tasks.

Let us assume we have practiced N tasks, $\tau_1, \ldots, \tau_N$, and used the corresponding (optimal or near-optimal) learned policies to construct a parameterized skill. Assume that we annotate each task τ_i with the time t_i it was practiced, which we denote by $\tau_i^{t_i}$. Here, time refers to the *order* in which tasks are practiced, so that the i-th task to have been practiced is annotated with time $t = i$. Assume that we have evaluated the efficiency of the skill on tasks $\tau_1, \ldots, \tau_N$ and observed performances $J(\tau_1), \ldots, J(\tau_N)$. Let D be a training set of tuples $\{\tau_i^{t_i}, J(\tau_i)\}$, for $i \in \{1 \ldots N\}$. Given this training set, Equations 3 and 4 can be used to compute a posterior distribution over skill performance, $P(J(\tau)|\tau, D)$, for any tasks τ. Let J_t be the posterior distribution obtained when conditioning the Gaussian Process on all tasks practiced up to time t, and let $\mu_t(\tau)$ and $\sigma_t^2(\tau)$ be its mean and variance, respectively. We define *Skill Performance* (SP) as the expected performance of the skill with respect to an arbitrary distribution P of tasks:

$$SP_t = \int P(\tau)\mu_t(\tau)d\tau. \quad (5)$$

Furthermore, let the *Expected Improvement in Skill Performance* (EISP), given task τ, be the expected increase in skill performance resulting from an agent practicing τ and observing subsequent skill performance $j(\tau)$. Here, $j(\tau)$ is an optimistic upper bound on the skill performance at τ, computed with respect to the current posterior distribution J_t. To compute the EISP of a task τ we consider the

Gaussian posterior, $\hat{J}_{t+1}$, that would result if J_t were to be updated with a new observation $\{(\tau, j(\tau))\}$. Let $\hat{\mu}_{t+1}(\tau)$ and $\hat{\sigma}^2_{t+1}(\tau)$ be the mean and variance of $\hat{J}_{t+1}$. The EISP of a task τ is defined as

$$\text{EISP}_t(\tau) = \int P(\tau')(\hat{\mu}_{t+1}(\tau') - \mu_t(\tau'))d\tau'. \quad (6)$$

EISP can be understood intuitively as a quantitative way of comparing tasks based on their likely contributions to improving the overall quality of the skill. Tasks whose practice may improve skill performance on a wide range of related tasks have higher EISP; conversely, tasks whose solutions are already well-modeled by the skill have lower EISP. Computing the EISP is similar to executing a mental evaluation of possible training outcomes: the agent uses its model of expected skill performance to estimate—without ever executing the skill—the effects that different training tasks may have on its competence across a distribution of problems.

The maximum of Equation 6 identifies the task τ^* which, if used to update the parameterized skill, results in the highest expected improvement in overall performance. This corresponds to an acquisition function which selects training tasks according to:

$$\tau^* = \arg\max_{\tau} \text{EISP}_t(\tau). \quad (7)$$

One way of evaluating Equation 7 is to use a gradient-based method. This requires an analytic expression for (or good approximation of) the gradient of $\text{EISP}_t(\tau)$ with respect to arbitrary tasks. To make notation less cluttered, we focus on the case of 1-dimensional task parameters; the extension to higher-dimensions is straightforward. Assume, without loss of generality, that the parameter describing a task is drawn from a bounded, continuous interval $[A, B]$. To derive the expression for the gradient of EISP, we first observe that $\mu_t(\tau')$ in Equation 6 does not depend on τ and can be removed from the maximization. It is possible to show that the function to be maximized in Equation 7 is equivalent to:

$$\text{EISP}_t(\tau) = G_t(\tau)^\top \left(\left(\mathbf{C_t}(\tau) + \sigma^2 I \right)^{-1} \mathbf{y}(\tau) \right), \quad (8)$$

where

$$G_t(\tau) = [g_t(\tau_1^{t_1}) \quad \dots \quad g(\tau_N^{t_N}) \quad g(\tau^{t+1})]^\top, \quad (9)$$

$$g_t(\tau_i^{t_i}) = \int_A^B P(r) k(r^{t+1}, \tau_i^{t_i}) dr, \quad (10)$$

$$\mathbf{y}(\tau) = [J(\tau_1) \dots J(\tau_N) \quad j(\tau)]^\top, \quad (11)$$

and where $\mathbf{C_t}(\tau)$ is the covariance matrix of the extended training set $D \cup \{(\tau^{t+1}, j(\tau))\}$:

$$\mathbf{C_t}(\tau) = \begin{pmatrix} k(\tau_1^{t_1}, \tau_1^{t_1}) & \dots & k(\tau_1^{t_1}, \tau^{t+1}) \\ \vdots & \ddots & \vdots \\ k(\tau_N^{t_N}, \tau_1^{t_1}) & \dots & k(\tau_N^{t_N}, \tau^{t+1}) \\ k(\tau^{t+1}, \tau_1^{t_1}) & \dots & k(\tau^{t+1}, \tau^{t+1}) \end{pmatrix}. \quad (12)$$

Furthermore, the gradient of $\text{EISP}_t(\tau)$ with respect to any given task τ is

$$\begin{aligned} \nabla_\tau \text{EISP}_t(\tau) &= \nabla_\tau G_t(\tau)^\top W_t(\tau) y(\tau) \\ &- G_t(\tau)^\top W_t(\tau) \nabla_\tau C_t(\tau) W_t(\tau) y(\tau) \\ &+ G_t(\tau)^\top W_t(\tau) \nabla_\tau y(\tau), \end{aligned} \quad (13)$$

where

$$\nabla_\tau G_t(\tau) = \left[\underbrace{0 \quad \dots \quad 0}_{\text{N times}} \quad \nabla_\tau g_t(\tau) \right]^\top, \quad (14)$$

$$\nabla_\tau y(\tau) = \left[\underbrace{0 \quad \dots \quad 0}_{\text{N times}} \quad \nabla_\tau j(\tau) \right]^\top, \quad (15)$$

and $W(\tau) = \left(C(\tau) + \sigma^2 I\right)^{-1}$. Let $j(\tau)$ be the upper endpoint of the 95% confidence interval around the mean predicted performance at τ: $j(\tau) = \mu_t(\tau) + 1.96\sqrt{\sigma_t^2(\tau)}$. Then, $\nabla_\tau j(\tau) = \nabla_\tau \left(\mu_t(\tau) + 1.96\sqrt{\sigma_t^2(\tau)}\right)$. If we assume that the variance σ_t^2 of the process is approximately constant within infinitesimal neighborhoods of a given task, then $\nabla_\tau j(\tau) = \nabla_\tau \mu_t(\tau)$, which can be rewritten as

$$\begin{aligned} \nabla_\tau j(\tau) = &\left(\nabla_\tau [k(\tau^t, \tau_1^{t_1}) \quad \dots \quad k(\tau^t, \tau_N^{t_N})]^\top \right) \times \\ &\left(\mathbf{C_D} + \sigma^2 I\right)^{-1} \mathbf{y_D}. \end{aligned} \quad (16)$$

It can be shown that $\nabla_\tau \text{EISP}_t(\tau)$ can be expressed as a linear form $\phi_1 j(\tau) + \phi_2 \mathbf{y_D}$, where both ϕ_1 and ϕ_2 depend only on the kernel function k and on the task parameters sampled so far. This reveals an interesting property: given an arbitrary fixed set of training tasks, the gradient of EISP can be linearly decomposed into one component that depends solely on the performances y_D that are achievable by the skill, and another component that depends solely on the optimistic assumptions made when defining $j(\cdot)$. This implies that the direction of maximum improvement of EISP is *independently* influenced by *1)* the generalization capabilities of the skill—specifically, the actual performances it achieves on various tasks; and *2)* the optimistic assumptions regarding how further practice of a particular task may improve its performance.

Note that some of the equations in this section depend directly or indirectly on the choice of kernel k. In Section 5 we introduce a novel spatiotemporal kernel specially designed to better model skill performance functions, and in Appendix A we derive analytical expressions for the quantities involving it; namely, $\nabla_\tau k(\tau_i^{t_i}, \cdot)$, $g_t(\tau_i^{t_i})$ and $\nabla_\tau g_t(\tau_i^{t_i})$.

5. Modeling Non-Stationary Skill Performance Functions

Kernels encode assumptions about the function being modeled by a GP, such as its smoothness and periodicity. An implicit assumption made by standard kernels is that the underlying function is stationary—that is, it does not change with time. Kernel functions also specify a measure of similarity or correlation between input points, usually defined in terms of the coordinates of the inputs. If dealing with non-stationary functions, however, defining similarities is harder: when a point is resampled, for instance, we generally expect the similarity between its new and previous values to decrease with time. Note that the model of expected performance introduced in Section 3 is intrinsically non-stationary, since skill performance naturally improves with practice. If a standard kernel were to be used to model this function, outdated performance observations would contribute to the predicted mean, thus keeping the GP from properly tracking the changing performance.

To address this issue we introduce a new spatiotemporal kernel designed to better model non-stationary skill performance functions. Let us assume an arbitrary kernel $k_S(\tau_1, \tau_2)$ capable of measuring the similarity between tasks based solely on their parameters τ_1 and τ_2. We expect k_S to be higher if comparing related tasks, and close to zero otherwise. Note that k_S does not account for the expected decrease in the similarity between observations of a task at very different times. To address this issue we construct a composite spatiotemporal kernel k_C, based on k_S, capable of evaluating the similarity between tasks based on their parameters and on the times they were sampled. Let $k_C(\tau_1^{t_1}, \tau_2^{t_2})$ be such a kernel, where τ_i^t denotes a task τ_i sampled at time t. For k_C to be suitable for modeling non-stationary functions, it should ensure the following properties: *1)* related tasks have higher similarity if sampled at similar times; that is, $k_C(\tau_1^{t_1}, \tau_2^{t_2}) > k_C(\tau_1^{t_1+\Delta t}, \tau_2^{t_2})$, for $\Delta t > 0$; *2)* if related tasks are sampled at significantly different times, no temporal correlation can be inferred and similarity is defined solely on their task parameters; that is, $k_C(\tau_1^{t_1}, \tau_2^{t_2}) \to k_S(\tau_1, \tau_2)$ as $|t_1 - t_2| \to \infty$; and *3)* the more unrelated tasks are, the smaller the correlation between them, independently of when they were sampled; that is, $k_C \to 0$ as $k_S \to 0$. The first property implies that if tasks are related, closer sampling times suggest higher correlation; the second property implies that nothing besides similarity in task space can be inferred if tasks are sampled at very different times; and the third property implies that sampling times, on their own, carry no correlation information if the tasks being compared are significantly different. To define k_C we introduce an isotropic exponential kernel $k_T(t_1, t_2)$ for measuring the similarity between sampling times:

$$k_T(t_1, t_2) = 1 + (C - 1) \exp\Big(- \rho^{-1}(t_1 - t_2)^2 \Big), \quad (17)$$

for some $C > 0$. k_T is such that $k_T \to C$ as $|t_1 - t_2| \to 0$, and $k_T \to 1$ as $|t_1 - t_2| \to \infty$. The parameter ρ is similar to the length-scale parameter in squared exponential kernels and regulates our prior assumption regarding how non-stationary the function is. We can now define k_C as

$$k_C(\tau_1^{t_1}, \tau_2^{t_2}) = k_S(\tau_1, \tau_2) \times \left(1 + (C - 1)e^{-\frac{(t_1 - t_2)^2}{\rho}}\right). \quad (18)$$

Intuitively, k_T boosts the correlation between tasks if they were sampled at similar times and ensures that only spatial correlation is taken into account as the difference between sampling times increases. Furthermore, note that when $C = 1$ all temporal information is ignored and k_C degenerates to the purely-spatial kernel k_S. Several methods, such as evidence maximization, are available to automatically identify suitable parameters for k_C and k_S (Rasmussen & Williams, 2006).

Figure 2 depicts the predicted posterior mean and variance of a process defined over a synthetic non-stationary function. Two curves are shown: one for the predicted mean if using the purely-spatial kernel k_S, which does not take sampling time into account, and one for the improved predicted mean obtained if using the spatiotemporal kernel k_C. Note how the latter kernel allows the predicted mean to correctly track the non-stationary function.

6. The Catapult Domain

We evaluate our task selection method on a simulated catapult control problem where the agent is tasked with learning a parameterized skill for hitting targets on mountainous terrains (Figure 3)[1]. Targets can be placed anywhere on a 2-dimensional terrain with various elevations and slopes—both of which are unknown to the agent. The task space T consists of a single parameter describing the horizontal distance from the catapult to the target; note that this task parameterization does not convey any information about the

[1]Code will be made available at http://bitbucket.org/bsilvapoa/active_paramskill.

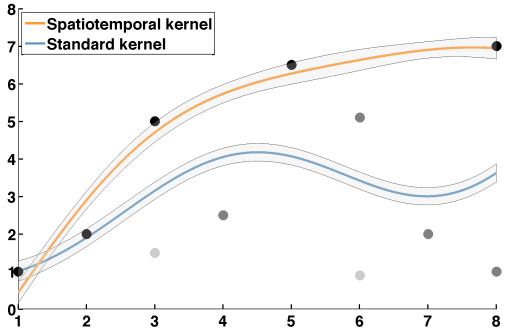

Figure 2. GP posteriors obtained when using a standard kernel and a spatiotemporal kernel. Lighter-colored points indicate older samples, while darker ones indicate more recent ones.

elevation of the target or the geometry of the terrain. Learning such a skill is difficult because it requires generalizing over an ensemble of continuous-state, continuous-action control problems. We learn the parameterized skill via Gaussian Process regression. The skill maps target positions to continuous launch parameters—namely, angle and velocity. Finally, we define performance of a policy as the distance between where the projectile hits and the intended target.

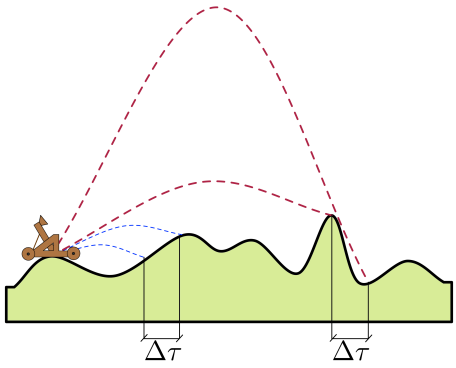

Figure 3. The Catapult Domain.

Determining which tasks to practice is challenging because irregular, non-smooth terrains may require significantly different launch profiles for hitting neighboring targets. Figure 3 depicts how a change $\Delta\tau$ in task parameters may require significantly different launch parameters depending on the region of the terrain. Figure 4 depicts the policy manifold associated with launch parameters required for hitting various targets on a randomly-generated terrain. The 1-dimensional task space is represented by the red line, and gray lines mapping points in task space to policy space indicate policy predictions made by the skill. Discontinuities in this mapping indicate irregular regions of the policy manifold in which generalization is difficult. Finally, note that identifying the target with maximum EISP corresponds to optimizing a one-step look-ahead sampling strategy to quickly uncover the structure of this manifold.

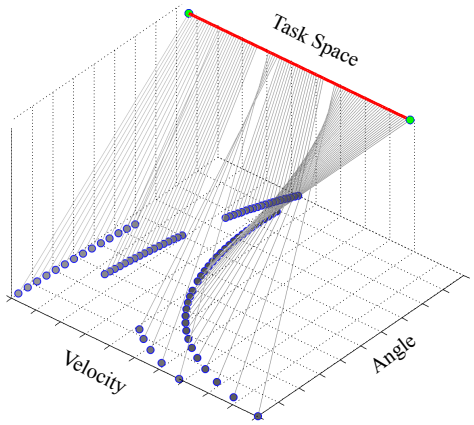

Figure 4. Policy manifold of the catapult domain. The red line represents the task space. Gray lines connecting task to policy space indicate predictions made by the skill. Discontinuities in the mapping indicate task regions where generalization is difficult.

We compared the performance of our method with four alternative approaches. Two of them are baseline, non-adaptive sampling strategies: selecting tasks uniformly at random, and probabilistically according to the task distribution P. We also compare with two active acquisition criteria commonly used in Bayesian optimization: Expected Improvement (EI) and Lower Confidence Bound (LCB). Figures 5 and 6 show skill performance as a function of the number of tasks practiced, for different task-selection methods. Skill performance was measured by evaluating the skill on a set of novel tasks; the observed performances were weighted by the task distribution P to reflect whether the agent was competent at tasks of higher interest. To report an absolute measure of skill quality we computed the mean squared difference between performance of the learned skill and the maximum performance that can be achieved by the skill model. The lower the difference, the better the skill is at solving tasks from the distribution of interest. All curves are averages over 50 randomly generated terrains.

Figure 5 shows how skill performance changes as the agent practices more tasks, assuming a uniform target distribution P of tasks. Note that in this case both non-adaptive sampling strategies—i.e., selecting tasks at random or drawing them from P—are equivalent. Similarly, Figure 6 shows skill performance as a function of tasks practiced but for the case of a non-uniform target distribution P—that is, an agent with stronger preference for becoming competent at targets in particular regions of the terrain. Here, P was defined as a Gaussian centered at the midpoint of the terrain. Under both types of task distribution, EI performed worse than all other methods. EI selects

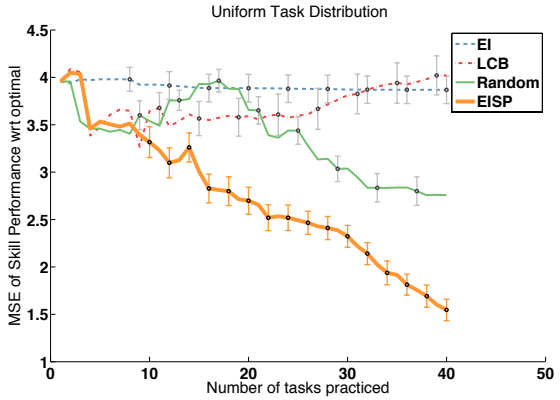

Figure 5. Average skill performance as a function of the number of sampled training tasks (uniform task distribution).

tasks whose individual performances are expected to improve the most by practice. This criterion leads the agent to repeatedly practice tasks that are already well-modeled by the skill but which may be marginally improved. This causes the agent to ignore regions of the task space in which it is not yet competent. LCB suffers from a similar shortcoming; it selects tasks in which the skill has *lowest* expected performance, thus focusing on improving the agent's weaknesses. This often leads the agent to obsessively practice tasks that it may be unable to solve well. Finally, both random selection of tasks and selection according to the target distribution P fail to account for the varying difficulty of tasks. These criteria choose to practice problems independently of the skill's current capability of solving them; furthermore, they often practice tasks that are irrelevant according to the target task distribution. EISP, on the other hand, correctly identifies which tasks can improve skill performance the most, and takes into account both their relative difficulties and how well their solutions generalize to related tasks.

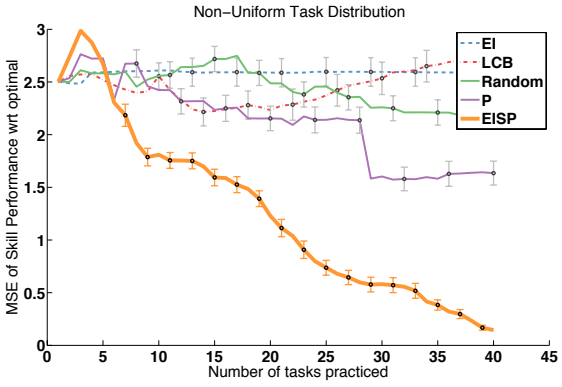

Figure 6. Average skill performance as a function of the number of sampled training tasks (non-uniform task distribution).

Figure 7 depicts the regions of task space that each method chooses to explore on a randomly-generated terrain. Random sampling and sampling according to the task distribution P do not adapt their task-selection strategies according to the problem. EI identifies two regions of the task space in which the skill is effective and focuses on trying to further improve those. LCB, on the contrary, samples more densely regions that contain difficult tasks. However, because it does not model whether skill performance is expected to improve, it often focuses on tasks that are too difficult. EISP prioritizes practice according to the target distribution P *and* selects problems according to how much they are expected to contribute to improving skill performance. In particular, note how it chooses to practice less on tasks at the beginning of the task range, even though those tasks have a high probability of occurring according to P. This happens because EISP quickly realizes that solutions to those tasks can be easily generalized and that no further samples are required. Finally, the peak of samples collect by EISP at the end of the task range corresponds to a particularly difficult part of the terrain which requires prolonged practice. Note how EISP devotes less attention to that region than EI since it is capable of predicting when no further skill improvement is expected.

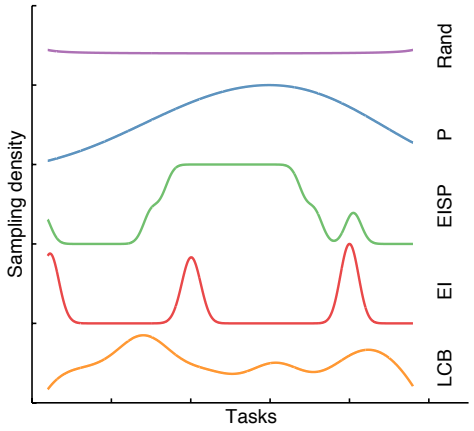

Figure 7. Density of samples collected by different training strategies.

We can draw a few important conclusions from our results: *1)* non-adaptive strategies implicitly assume that all tasks can be equally well generalized by the skill—or, equivalently, that the manifold has approximately uniform curvature; this causes them to unnecessarily practice tasks that may already be well-modeled by the skill; *2)* the LCB criterion always selects tasks with lowest predicted performance, often repeatedly practicing poorly-performing tasks as long as they can be infinitesimally improved; *3)* EI focuses on further improving single tasks at which the agent may already be competent, thus refraining from practicing more difficult ones, or ones that are more important according to the task distribution; and finally *4)* because EISP

uses a model of expected performance to infer the generalization capabilities of a skill, it correctly identifies the task regions in which practice leads to better generalization across a wider range of tasks. Furthermore, the use of an expected skill performance model allows for the identification of tasks that are either too difficult to solve or whose performance cannot yet be further improved, thus leading the agent to focus on problems that are compatible with its current level of competence.

7. Related Work

In Section 5 we introduced a spatiotemporal composite kernel to model non-stationary skill performance functions. Other methods have been proposed to allow GP regression of non-stationary functions. Rottmann and Burgard (2010) manually assigned higher noise levels to older samples, causing them to contribute less to the predicted mean. This approach, however, relies on domain-dependent cost functions, which are difficult to design. Spatiotemporal covariance functions, defined similarly to our composite kernel k_C, have been proposed to solve recursive least-squares problems on non-stationary domains. These functions, constructed by the product of a standard kernel and an a Ornstein-Uhlenbeck temporal kernel, often require estimating a forgetting factor (Van Vaerenbergh et al., 2012).

Previous research has also addressed the problem of selecting training tasks to efficiently acquire a skill. Hausknecht and Stone (2011) constructed a skill by solving a large number of tasks uniformly drawn from the task space. They exhaustively varied policy parameters and identified which tasks were solved, thus implicitly acquiring the skill by sampling *all* possible tasks. Kober, Wilhelm, Oztop, and Peters (2012) proposed a Cost-regularized Kernel Regression method for learning a skill but did not address how to select training tasks; in their experiments, tasks were sampled uniformly at random from the task space. Similarly, da Silva, Konidaris, and Barto (2012) proposed to acquire a skill by analyzing the structure of the underlying policy manifold, but assumed that tasks were selected uniformly at random. Finally, Baranes and Oudeyer (2013) proposed an active learning framework for acquiring parameterized skills based on competence progress. Their approach is similar to ours in that promising tasks are identified via an adaptive mechanism based on expected performance. However, their approach, unlike ours, requires a discrete number of tasks and can only optimize the task-selection problem over finite and discrete subsets of the task space.

8. Conclusions and Future Work

We have presented a general framework for actively learning parameterized skills. Our method uses a novel acquisition criterion capable of identifying tasks that maximize expected skill performance improvement. We have derived the analytical expressions necessary for optimizing it and proposed a new spatiotemporal kernel especially tailored for non-stationary performance models. Our method is agnostic to policy and skill representation and can be coupled with any of the recently-proposed parameterized skill learning algorithms (Kober et al., 2012; da Silva et al., 2012; Neumann et al., 2013; Deisenroth et al., 2014).

This work can be extended in several important directions. The composite kernel k_C can be used to compute a posterior over expected *future* skill performance, which suggests an extension of EISP to the case of multistep decisions. This can be done by evaluating the predicted posterior mean (Equation 3) over a set of test points $\{\tau_i, T+\Delta\}$, where T is the current time and Δ is a positive time increment. This is useful in domains where a one-step look-ahead strategy, like the one optimized in Equation 7, is too myopic. Secondly, our model uses a homoscedastic GP prior, which assumes constant observation noise throughout the input domain. This may be limiting if the agent has sensors with variable accuracy depending on the task—for instance, it may be unable to accurately identify the position of distant targets. Heteroscedastic GP models (Kuindersma et al., 2012) may be used to address this limitation. Finally, taking advantage of human demonstrations may help biasing EISP towards tasks which an expert deems relevant, which suggests an integration with active learning from demonstration techniques (Silver et al., 2012).

A. Appendix

Analytical solutions to Equations 7 and 8 depend on the choice of kernel. If we assume that P is a uniform distribution over tasks and define k_S as the squared exponential kernel $k_S(\tau_1, \tau_2) = \sigma_f^2 \exp(-L^{-1}(\tau_1 - \tau_2)^2)$, then:

$$
\begin{aligned}
\nabla_\tau k_C(\tau^t, \tau_i^{t_i}) &= -\frac{2}{L}(\tau - \tau_i) k_C(\tau^t, \tau_i^{t_i}) \\
g_t(\tau_i^{t_i}) &= \frac{1}{2}\left(\sigma_f^2 \sqrt{\pi L}\right) k_T(t+1, t_i) \times \\
&\quad \left[\operatorname{erf}\left(\frac{\tau_i - A}{\sqrt{L}}\right) - \operatorname{erf}\left(\frac{\tau_i - B}{\sqrt{L}}\right)\right] \\
\nabla_\tau g_t(\tau_i^{t_i}) &= \sigma_f^2 k_T(t+1, t_i) \times \\
&\quad \left[\exp\left(\frac{-(A-\tau_i)^2}{L}\right) - \right. \\
&\quad \left. \exp\left(\frac{-(B-\tau_i)^2}{L}\right)\right].
\end{aligned}
$$

where erf(z) is the Gauss error function.

References

Baranes, A. and Oudeyer, P. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):69–73, 2013.

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

Deisenroth, M.P., Englert, P., Peters, J., and Fox, D. Multi-task policy search for robotics. In *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, 2014.

Ericsson, K. *The influence of experience and deliberate practice on the development of superior expert performance*, chapter 13, pp. 685–708. Cambridge University Press, 2006.

Hausknecht, M. and Stone, P. Learning powerful kicks on the Aibo ERS-7: The quest for a striker. In *RoboCup-2010: Robot Soccer World Cup XIV*, volume 6556 of *Lecture Notes in Artificial Intelligence*, pp. 254–65. Springer Verlag, 2011.

Kober, J., Wilhelm, A., Oztop, .E, and Peters, J. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*, pp. 1–19, 4 2012.

Kroemer, O., Detry, R., Piater, J., and Peters, J. Combining active learning and reactive control for robot grasping. *Robotics and Autonomous Systems*, 58(9):1105–1116, 2010.

Kuindersma, S., Grupen, R., and Barto, A. Variational Bayesian optimization for runtime risk-sensitive control. In *Robotics: Science and Systems VIII (RSS)*, Sydney, Australia, July 2012.

Neumann, G., Daniel, C., Kupcsik, A, Deisenroth, M., and Peters, J. Information-theoretic motor skill learning. In *Proceedings of the AAAI Workshop on Intelligent Robotic Systems*, 2013.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Rottmann, A. and Burgard, W. Learning non-stationary system dynamics online using gaussian processes. In *Proceedings of the Thirty-second Symposium of the German Association for Pattern Recognition*, pp. 192–201, 2010.

Silver, D., Bagnell, J., and Stentz, A. Active learning from demonstration for robust autonomous navigation. In *Proceedings of 2012 IEEE International Conference on Robotics and Automation*, May 2012.

Snoek, J., Larochelle, H., and Adams, R. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pp. 2951–2959, 2012.

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

Van Vaerenbergh, S., Santamaría, I., and Lázaro-Gredilla, M. Estimation of the forgetting factor in kernel recursive least squares. In *Proceedings of the IEEE International Workshop On Machine Learning For Signal Processing*, September 2012.