Hazara, Murtaza; Li, Xiaopu; Kyrki, Ville

Active Incremental Learning of a Contextual Skill Model

Published in:
Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019

DOI:
10.1109/IROS40897.2019.8967837

Published: 01/01/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Hazara, M., Li, X., & Kyrki, V. (2019). Active Incremental Learning of a Contextual Skill Model. In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019* (pp. 1834-1839). Article 8967837 (Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems). IEEE. https://doi.org/10.1109/IROS40897.2019.8967837

Active Incremental Learning of a Contextual Skill Model

Murtaza Hazara*, Xiaopu Li*, and Ville Kyrki

***Abstract*— Contextual skill models are learned to provide skills over a range of task parameters, often using regression across optimal task-specific policies. However, the sequential nature of the learning process is usually neglected. In this paper, we propose to use active incremental learning by selecting a task which maximizes performance improvement over entire task set. The proposed framework exploits knowledge of individual tasks accumulated in a database and shares it among the tasks using a contextual skill model. The framework is agnostic to the type of policy representation, skill model, and policy search. We evaluated the skill improvement rate in two tasks, ball-in-a-cup and basketball. In both, active selection of tasks lead to a consistent improvement in skill performance over a baseline.**

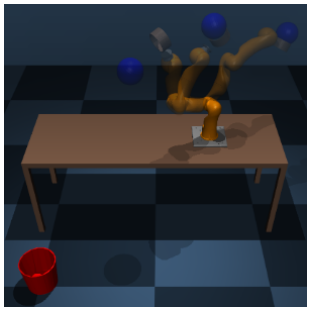

Fig. 1: Learning basketball skill using KUKA LBR 4+ in MuJoCo.

I. INTRODUCTION

Skill learning in animals is incremental [1]. For example, monkeys retain existing motor skills for future learning [2] such that skills learned for particular tasks are used to improve future learning. Similarly, schema theory states that humans also learn skills incrementally [3], [4], [5]. In fact, mainly because of sequential flow of information [6], limited memory and processing power, people cannot have access all the previously acquired information about previously learned tasks. Thus, they tend to retain and update generalizable aspects of a task for future use. For example, when learning to throw a basketball, a person can first learn to score from a fixed location. They will then move on to another location. Subsequently, generalizing to new situations (e.g. location) becomes easier as the individual learns incrementally the underlying regularities of the ball throwing skill (see Fig. 1).

In robotics, regression has been applied to learn the generalizable aspects of tasks using a contextual skill model (CSM) [7], [8], [9], [10], [11]. These methods have achieved zero-shot generalization where learning is not necessary for new situations. However, they assume the availability of optimal sample policies in advance of the generalization where each sample has been learned independently from a human demonstration (LfD). On the other hand, contextual policy search (CPS) learns optimal policies from scratch while maintaining a linear CSM [12], [13]. However, CPS requires learning of a task for a new situation. Furthermore, CPS neglects the sequential nature of decision making.

In contrast to isolated learning where a CSM has been built from independent optimal policies learned using LfD, incremental learning combines regression with policy search to construct a CSM incrementally from scratch [14]. In this framework, tasks are assumed to arrive sequentially and knowledge is shared among related tasks in a database (DB). Incremental learning has been shown to generalize better than the isolated learning both in terms of interpolation, extrapolation and the speed of learning [15]. Furthermore, it has been transferred incrementally from simulation to the real world [16].

However, the learning process in [15], [14] is passive where the agent does not have control over the order of tasks. Instead of choosing tasks randomly, they could be selected to maximize future learning performance [17]. However, they assume learning is continued with a task until it converges, that is, an optimal policy is achieved.

In this paper, we propose a novel active incremental learning framework. The main focus of this paper is to endow incremental learning with a task manager. The task manager selects a new task by maximizing future learning while considering the current task performance. In this way, continuous incremental learning is achieved with a minimum effort generalization to new situations.

The main benefit of the proposed framework is being agnostic to the policy representation, the contextual skill model and to the used policy search approach. We evaluated how efficiently the proposed framework can learn a skill model in two tasks in simulation. Results demonstrated that active learning achieved significant improvement over random task order consistently in both skills. Furthermore, in both tasks, the generalization performance consistently improved indicating continuous incremental learning.

II. RELATED WORK

In this section we will briefly review the generalizable skill models learned either using regression or CPS, active learning and incremental learning.

This work was supported by Academy of Finland, decision #268580

* Two first authors contributed equally to the work. Authors are with Department of Electrical Engineering and Automation, Aalto University, Finland murtaza.hazara@aalto.fi, xiaopu.li@aalto.fi, ville.kyrki@aalto.fi

A. Generalization using Regression

Regression has been used to learn a CSM from previously learned optimal policies [7], [8], [9], [10], [18] where dynamic movement primitives (DMPs) has been their main policy representation. Calinon et al. [7] uses a Gaussian mixture model as the CSM and generalize it to new situations using expectation maximization. Although their model is capable of linear extrapolation, it is only applicable when the task parameters can be represented in the form of coordinate systems. Stulp et al. [8] used Gaussian kernels for the CSM and merged it with the forcing function of DMPs which is also represented using a weighted sum of Gaussian kernels. The merging by multiplication of the two Gaussian kernels resulted into a two dimensional kernel which is a function of task parameter where the kernel centers can be selected automatically and arbitrarily. Forte et al. [9] utilized Gaussian process regression (GPR) where the inverse of the kernel is calculated using only the training samples resulting in fast generalization suitable for on-line application. Ude et al. [10] has provided a generalizable LfD framework where they use GPR for mapping the task parameters to meta-parameters such as goal and duration of DMPs, and they apply LWR to encode their CSM. All the previous generalizable LfD frameworks have used a local model for their CSM which can achieve interpolation across training samples. On the other hand, parametric CSM which can achieve extrapolation using linear [19] and non-linear models [18] have also been proposed.

B. Generalization using contextual policy search

The generalizable LfD models using regression have assumed the availability of optimal policy parameters in a database of motor primitives (MPs). These optimal policy parameters have either been imitated from several human demonstrations, or learned using reinforcement learning (RL) in an isolated manner. On the other hand, contextual policy search (CPS) do not assume the availability of these optimal MPs in advance of generalization. In fact, CPS learn the policies in an online manner from the scratch. CPS has been applied to learn a CSM using a model-based approach [12] and and also a model-free approach [13]. In both of these approaches the CSM is linear and modeled using a Gaussian function whose hyper-parameters are updated iteratively. However, CPS neglects the sequential nature of decision making.

C. Active Learning

Unlike passive contextual skill modeling where an agent is provided with a manually selected list of task parameters to learn, active learning provides the agent with a tool to select a task parameter automatically. The objective of active learning is to select a task which maximizes future skill performance. Active learning has been considered in [20], [21], [17]. We will review very briefly [20] and [21] and elaborate on [17] since it is the most relevant to our proposed approach.

In [20], a heuristic reward function is used with a discounted multi-arm bandit to actively select the next task. In [21], a task is selected based on active contextual entropy search (ACES) which is an information theoretic approach minimizing uncertainty about optimal policy parameters for task parameters.

Da Silva et al. [17] provided a non-parametric Bayesian approach for active learning of a contextual skill model. They model reward $R(\tau)$ for a certain task τ using a Gaussian process (GP) with a spatio-temporal kernel which can accommodate the non-stationary behavior of a reward function. They learn a posterior $P(R_t(\tau)|\tau, D_t)$ with mean $\mu_t(\tau)$ and variance $\sigma_t^2(\tau)$ from current database $D_t = \{(\tau_1, r(\tau_1)), \ldots, (\tau_N, r(\tau_N))\}$ corresponding to the evaluated total reward $r(\tau_i)$ of optimal policies which have been practiced for previously selected tasks τ_i. They consider the skill performance $SP_t = \int P(\tau)\mu_t(\tau)d\tau$ across the task space where $p(\tau)$ denotes the probability of task τ occurring. They introduced an acquisition function which involves the expected improvement for a candidate task τ_c

$$EISP_t(\tau_c) = \int P(\tau')(\hat{\mu}_{t+1}(\tau') - \mu_t(\tau'))d\tau', \quad (1)$$

where $\hat{\mu}_{t+1}$ represents the mean of Gaussian posterior $\hat{R}_{t+1}$ which is computed by fitting a GP to the updated database $D_u = D_t \cup \{(\tau_c, \hat{j}(\tau_c))\}$. They used an optimistic upper bound $\hat{j}(\tau_c) = \mu_t(\tau) + 1.96\sqrt{\sigma_t^2(\tau)}$ estimated based on the current GP posterior reward model R_t. A task will be selected according to $\tau^* = \arg\max_\tau EISP_t(\tau)$.

D. Incremental Learning

The current dominant machine learning paradigm is isolated learning, where a model is learned for a task in an isolated fashion (see Figure 2.a). Once a new task is encountered, the learning process must be repeated while the previously learned models are ignored. Hence, learning several tasks require substantial amount of data as each task is learned separately and from the scratch. In fact, this isolated framework does not share information across the tasks. In contrast, schema theory suggests that human learning involves sharing information among related tasks, using a knowledge base and adapting it to accommodate information acquired from learning a new task. This feature has been addressed in the context of lifelong learning framework.

Lifelong (incremental) learning is a framework which provides continuous learning of tasks arriving sequentially [22], [23], [24]. The essential component of this framework (see Fig. 2.b) is a database (DB) which maintains the knowledge acquired from previously learned tasks $\tau_1, \tau_2, \cdots, \tau_{N-1}$. Incremental learning starts from the task manager assigning a new task τ_N to a learning agent. In this case, the agent exploits the knowledge in the DB as a prior data for enhancing the generalization performance of its model on the new task. After the new task τ_N is learned, DB is updated with the knowledge obtained from learning τ_N. In fact, the incremental learning framework provides an agent with three capabilities:

1) continuous learning
2) knowledge accumulation
3) re-using previous knowledge to enhance future learning

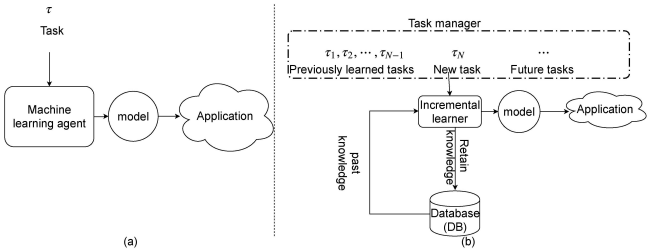

Fig. 2: Two machine learning paradigms: (a) isolated learning versus (b) incremental learning.

III. METHOD

A. Problem Definition

We assume that tasks arrive sequentially and we will have a database $\mathcal{D}_t = \{(\boldsymbol{\tau}_i, \boldsymbol{\theta}_{\boldsymbol{\tau}_i}) | i = 1 \dots N\}$ at time t consisting of N sample set of task parameters $\boldsymbol{\tau}_i$ and their associated policy parameters $\boldsymbol{\theta}_{\boldsymbol{\tau}_i}$. A skill model S_t extracts the knowledge accumulated in $\mathcal{D}_t$ by fitting a regression model mapping a task parameter $\boldsymbol{\tau}$ to policy parameters $\boldsymbol{\theta} = S(\boldsymbol{\tau})$. Using $S_t(\boldsymbol{\tau})$, we can generalize the policy parameters for any situation characterized by a measurable task parameter $\boldsymbol{\tau}$.

We also assume that executing the policy with parameters $\boldsymbol{\theta}_{\boldsymbol{\tau}}$ generated by $S(\boldsymbol{\tau})$ for a specific task $\boldsymbol{\tau}$ will result into a deterministic performance behavior evaluated by $r(\boldsymbol{\tau}; S)$. Next, we define the skill performance

$$SP(S_t) = \int P(\boldsymbol{\tau}) r(\boldsymbol{\tau}; S_t) d\boldsymbol{\tau}, \tag{2}$$

where $P(\boldsymbol{\tau})$ denotes the probability which the task $\boldsymbol{\tau}$ occurs. We assume that the tasks occur with the same probability. Thus, we can rewrite (2) into

$$SP(S_t) = \frac{1}{\boldsymbol{\tau}_{max} - \boldsymbol{\tau}_{min}} \int_{\boldsymbol{\tau}_{min}}^{\boldsymbol{\tau}_{max}} r(\boldsymbol{\tau}; S_t) d\boldsymbol{\tau}. \tag{3}$$

Using the skill performance, we can define the expected skill performance

$$\begin{aligned} EISP(\boldsymbol{\tau}_c) &= SP(S_{t+1}) - SP(S_t) \\ &= \frac{1}{\boldsymbol{\tau}_{max} - \boldsymbol{\tau}_{min}} \int_{\boldsymbol{\tau}_{min}}^{\boldsymbol{\tau}_{max}} r(\boldsymbol{\tau}; S_{t+1}) - r(\boldsymbol{\tau}; S_t) d\boldsymbol{\tau}, \end{aligned} \tag{4}$$

where S_{t+1} represents the skill model which is fit to the updated database $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\boldsymbol{\tau}_c, \boldsymbol{\theta}_{\tau_c})\}$. It is worth mentioning that the proposed $EISP$ in (4) corresponds to the expected skill performance definition (1) considered in [17].

B. Active Incremental Learning

Policy search optimizes a parametric policy by updating its parameters iteratively. To be able to predict reward improvement over a single iteration, we need to model the learning rate of policy search, that is, the evolution of total rewards over time. We assume the learning rate can be modeled with function $J(t)$ that approaches the optimal rewards R^* as $t \to \infty$. Furthermore, we assume that $J(t)$ does not depend on task parameters. In other words, the convergence profile is independent of the task parameters, even if the current rewards for different tasks may vary, indicating that the policy has at that point converged more for some tasks than others. On the other hand, to model the consistency of the skill across tasks, we assume that rewards achieved by the skill model are similar for similar task parameters. This consistency is then modeled with current reward model $R(\boldsymbol{\tau})$. Using these models, we can evaluate $EISP$ for any task.

We assume a policy expressed in the form

$$u = \boldsymbol{\theta}^T \boldsymbol{g}(\boldsymbol{x}), \tag{5}$$

where u denotes an action, $\boldsymbol{\theta}$ represents a vector of policy parameters, and $\boldsymbol{g}$ is the vector of basis functions (kernels). Several policy encoding follows the parametric representation in (5) such as dynamic movement primitives [25], radial basis functions [26], or a linear policy. For learning the corresponding optimal policy parameter $\boldsymbol{\theta}_{\boldsymbol{\tau}_0}$, we can apply a model-based RL approach such as PILCO [27] or Black-DROPS [28]. We can also apply model-free RL such as PoWER [29], REPS [30], and PI^2 [31], [32].

Incremental learning of a contextual skill model begins with initializing the database $\mathcal{D}$, skill model $S(\boldsymbol{\tau})$, and learning rate model $J(t; \beta_J)$ in lines 1-5 (see Algorithm 1). We utilize an exponential family to represent the learning rate

$$J(t; \beta_J) = \exp(a(t - b)) + c_J \tag{6}$$

where $\beta_J = \{a, b, c_J\}$ denotes the hyper-parameters of the learning rate model. The hyper-parameters are estimated using [33] with data gathered while optimizing policy parameters $\boldsymbol{\theta}_{\boldsymbol{\tau}_0}$ for an initial task parameter $\boldsymbol{\tau}_0$. After that, we estimate the skill model $S(\boldsymbol{\tau})$ using the database $\mathcal{D}$ containing the initial sample $(\boldsymbol{\tau}_0, \boldsymbol{\theta}^*_{\boldsymbol{\tau}_0})$.

We then update the skill model $S(\boldsymbol{\tau})$ in an incremental manner (lines 7-16). This is achieved by running an iterative process where the main steps are predicting reward improvement in line 10, evaluating the expected improvement of skill performance $EISP$ for all alternatives in a discrete evaluation set of tasks $\boldsymbol{\tau}_{eval}$ in lines 9-12, selecting the most promising task $\boldsymbol{\tau}^*$ which maximizes $EISP$ in line 13 and updating the corresponding policy parameters $\boldsymbol{\theta}_{\boldsymbol{\tau}^*}$ by running one (or Δ) update steps of policy search in line 14. Note that, the policy parameter $\boldsymbol{\theta}_{\boldsymbol{\tau}^*}$ is not necessarily (sub-)optimal for $\boldsymbol{\tau}^*$ since we did not run the policy search until convergence.

In order to model the reward $R(\boldsymbol{\tau})$ across tasks, we evaluate the reward achievable by the current estimate of the skill model $S_t(\boldsymbol{\tau})$ for every task $\boldsymbol{\tau}_j \in \boldsymbol{\tau}_{eval}$ in the evaluation set $\boldsymbol{\tau}_{eval}$ in line 7. This is achieved by calculating the corresponding policy parameter vector $\boldsymbol{\theta}_{\boldsymbol{\tau}_j}$ using the skill model $S_t(\boldsymbol{\tau}_j)$; executing the policy with $\boldsymbol{\theta}_{\boldsymbol{\tau}_j}$ will lead to reward $r(\boldsymbol{\tau}_j)$. Using the evaluated rewards, we can build a reward model

$$R(\boldsymbol{\tau}; \beta_R) \sim GP(\boldsymbol{\tau}, \beta_R), \tag{7}$$

using GP with hyper-parameter β_R which can be optimized by maximizing evidence function [34].

In order to be able to calculate the $EISP$ for every candidate task $\boldsymbol{\tau}_c$, we need to predict the reward improvement

if we continue optimizing the corresponding policy parameters $\boldsymbol{\theta}_{\boldsymbol{\tau}_c}$ for Δ update steps of policy search. The expected improvement for a specific candidate task $\boldsymbol{\tau}_c$ is calculated using the learning rate model $J(t;\beta_J)$. First, the time index t_c corresponding the candidate task $\boldsymbol{\tau}_c$ is computed by reading from the inverse of the learning rate model

$$t_c = J^{-1}(R(\boldsymbol{\tau}_c;\beta_J)). \tag{8}$$

Then, the expected reward improvement for the candidate task $\Delta R(\boldsymbol{\tau}_c)$ is computed by

$$\Delta R(\boldsymbol{\tau}_c) = J(t_c + \Delta;\beta_J) - R(\boldsymbol{\tau}_c;\beta_R). \tag{9}$$

Next, we compute the expected reward $r(\boldsymbol{\tau}, S_{t+1})$ where S_{t+1} represents the skill model built using $\mathcal{D} = \mathcal{D} \cup (\boldsymbol{\tau}_c, \boldsymbol{\theta}_{\boldsymbol{\tau}_c})$. The policy parameters $\boldsymbol{\theta}_{\boldsymbol{\tau}_c}$ for the candidate task $\boldsymbol{\tau}_c$ have been computed using the current estimate of the skill model $S_t(\boldsymbol{\tau}_c)$. Instead of evaluating the improvement, we predict it using

$$r(\boldsymbol{\tau}; S_{t+1}) = R(\boldsymbol{\tau};\beta_R) + \Delta R(\boldsymbol{\tau}) \times \exp(c_d \left\|\boldsymbol{\tau} - \boldsymbol{\tau}_{\mathbf{c}}\right\|^2), \tag{10}$$

which is based on our second assumption that the reward across task parameters changes smoothly. The constant c_d controls the similarity across tasks. We used $c_d = -0.1$ in our experiments. Now that we have predicted the reward improvement, we evaluate the $EISP$ (4) in discrete form as

$$EISP = \frac{1}{\boldsymbol{\tau}_{max} - \boldsymbol{\tau}_{min}} \sum_{\boldsymbol{\tau}=\boldsymbol{\tau}_{min}}^{\boldsymbol{\tau}_{max}} r(\boldsymbol{\tau}; S_{t+1}) - r(\boldsymbol{\tau}; S_t) \tag{11}$$

.

Algorithm 1 Active Incremental Learning of a CSM $S(\boldsymbol{\tau})$

Input: $\boldsymbol{\tau} = \{\boldsymbol{\tau}_i \mid 1 \leq i \leq n\}$, $\boldsymbol{\tau}_{eval} = \{\boldsymbol{\tau}_j \mid 1 \leq j \leq k\}$
Output: Skill model $S(\tau)$.

Initialization :
Choose initial task parameter $\boldsymbol{\tau}_0$.
Optimize policy for $\boldsymbol{\tau}_0$ using RL to determine $\boldsymbol{\theta}^*_{\boldsymbol{\tau}_0}$.
Estimate parameters β_J for learning rate model $J(t;\beta_J)$.
Initialize database of policies $\mathcal{D} = \{(\boldsymbol{\tau}_0, \boldsymbol{\theta}^*_{\boldsymbol{\tau}_0})\}$.
Estimate skill model $S(\boldsymbol{\tau})$ with $\mathcal{D}$.
repeat
  Evaluate $r(\boldsymbol{\tau})$ for $\boldsymbol{\tau} \in \boldsymbol{\tau}_{eval}$.
  Estimate parameters β_R for reward model $R(\boldsymbol{\tau};\beta_R)$ using $r(\boldsymbol{\tau})$.
  for each $\boldsymbol{\tau}_c \in \boldsymbol{\tau}_{eval}$ **do**
    Predict reward improvement $\Delta R(\boldsymbol{\tau}_c)$ using (9).
    Evaluate $EISP(\boldsymbol{\tau}_c)$ using (4) and (10).
  end for
  Choose next task $\tau^* = \arg\max_\tau EISP(\tau)$.
  Optimize policy for one step for τ^* to determine θ_{τ^*}.
  Update $D = D \cup \{(\tau^*, \theta_{\tau^*})\}$.
  Re-estimate $S(\tau)$ with D.
until S provides success for all $\boldsymbol{\tau} \in \boldsymbol{\tau}_{eval}$.
return Skill model $S(\tau)$.

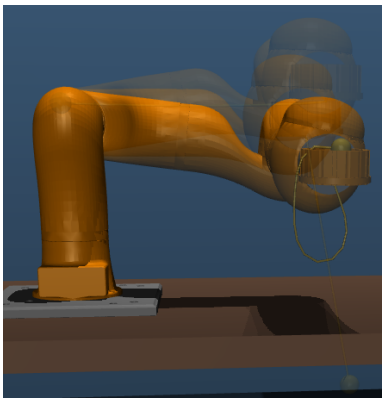

Fig. 3: Learning ball-in-a-cup skill using KUKA LBR 4+ in MuJoCo.

IV. EXPERIMENT

We studied experimentally the benefit of the proposed active incremental learning framework on improving the expected skill performance using ball-in-a-cup and basketball tasks on KUKA LBR 4+ in an environment simulated with MuJoCo. We utilized DMPs as the policy encoding since it provides us with a low-dimensional policy representation which is a less data-demanding model than high-dimensional policy representations such as deep RL. In this case, the action u in (5) corresponds to the forcing function of DMPs $u_d = \alpha_x(\beta_x(g - x) - \dot{x}) + u$ where $u = \boldsymbol{\theta}^T \boldsymbol{g}$ [15]. In this section, we explain the tasks, contextual skill model, and then analyze the result of active incremental learning.

A. Ball-in-a-Cup Task

The ball-in-a-cup game consists of a cup, a string, and a ball; the ball is attached to the cup by the string (see Fig. 3). The objective of the game is to get the ball in the cup by moving the cup. We chose the ball-in-a-cup game because variation in the environment can be generated by changing the string length. The string length is observable and easy to evaluate, thus providing a suitable task parameter, which was varied within $\tau \in \{29 \text{ cm}, 30 \text{ cm}, \ldots, 43 \text{ cm}\}$. Nevertheless, changing the string length results into a significant change in the dynamics of the task which requires a complex change in the motion to succeed in the game. Hence, the generalization capability of a CSM can be easily assessed using this game. Similar to our previous set-up in [15], the trajectories along y and z axes were encoded using separate DMPs. Utilizing 20 kernels per DMP, in total $N = 40$ parameters are needed to describe the motion model for a single task parameter value.

B. Basketball Task

The basketball game consists of a ball holder, a basket, and a ball; the holder is attached to the end-effector of KUKA LBR 4+ (see Fig. 1) and the basket is set at a certain distance from the robot. The objective of the game is to throw the ball at the basket. In this case, the task parameter is the distance of the basket from the base of the robot, which was varied within $\tau \in \{120 \text{ cm}, 130 \text{ cm}, \ldots, 240 \text{ cm}\}$. KUKA LBR 4+

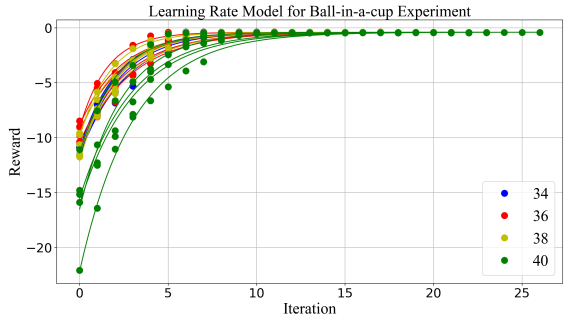

Fig. 4: Learning rate of model-free policy search observed for ball-in-a-cup with different task parameters.

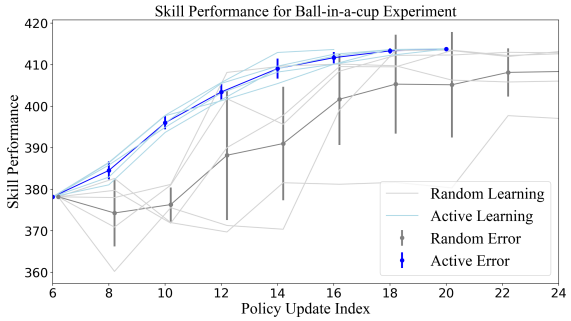

Fig. 5: Skill performance on ball-in-a-cup skill: active (in blue) versus random task selection (in grey).

has seven DOF, but only joints 2, 3, and 6 were used; the rest of the joints were kept fixed. Using 20 kernels per DMP, total of $N = 60$ parameters need to be determined for a task parameter value.

C. Contextual Skill Model

To map the task parameters to policy parameters, we used $GPDMP$ which is a parametric CSM with non-linear basis functions. We selected $GPDMP$ because of its generalization capabilities which has been shown to perform better than the linear CSM [18] or local models using model selection [15]. Besides that, it has been used in simulation to real world transfer [16].

D. Learning Rate Model

In order to verify our assumptions on the learning rate, we performed an experiment where we learned ball-in-a-cup game for different task parameters using model-free policy search. We started the learning process from the same initial policy parameters for all task parameters. The learning rate curves are shown in Fig. 4, with different colors indicating different task parameters. It can be observed, firstly, that the exponential model fits observed learning rates well. Secondly, the alignment of the curves indicates that the learning rate does not depend on the task parameter—even though the initial rewards may differ, the convergence rates are similar across task parameters. Similar observations were made for the basketball task, figure omitted here for brevity.

E. Active Incremental Learning

To study the performance benefit of the active task choice in incremental learning, we applied the proposed algorithm (see Algorithm 1) for learning ball-in-a-cup and basketball skills. As a baseline we used random order for tasks. We performed both active and random task selection 5 times.

The initial task parameters were $\tau_0 = 35$ cm for the string length in ball-in-a-cup, $\tau_0 = 180$ cm for the distance in basketball. Using PoWER [29] to train the initial task, policy converged after 6 policy updates for the ball-in-a-cup, 5 updates for the basketball. During incremental learning, $\Delta = 2$ updates were made in each policy search iteration using PoWER.

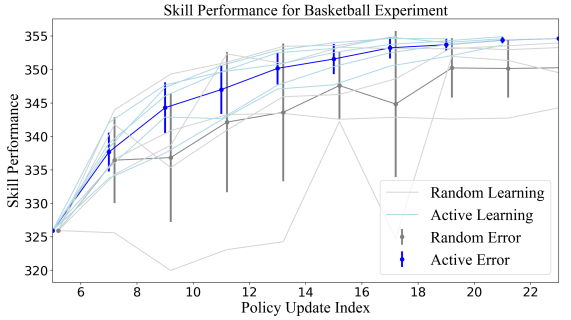

Fig. 6: Skill performance on basketball skill: active (in blue) versus random task selection (in grey).

Skill performance SP over time is shown in Figs. 5 (ball-in-a-cup) and 6 (basketball) where blue curve denotes the proposed active method and grey curve is the baseline, error bars denoting 1 standard deviation. As expected, the skill performance improves over time for both methods. However, the active method improves the skill performance more consistently, both in terms of learning faster on average as well as having smaller variance. With the active learning, the entire range of task parameters was successful after 20/23 policy updates (ball-in-a-cup/basketball correspondingly), while the success rate for the baseline after the same number of policy updates was 75%/80% (ball-in-a-cup/basketball correspondingly).

V. CONCLUSION

We proposed an active incremental learning framework for learning a contextual skill model. The framework allows learning of several related tasks in parallel such that information across policies is combined into the skill model while the order of tasks is optimized to maximize performance over all tasks. Experiments indicated that the active selection of task order during learning improves learning performance significantly.

The proposed approach models the learning rate deterministically which is obviously not true in general in reinforcement learning, even if the results show that the mean behavior is sufficient to provide consistent behavior. In environments where there is high stochasticity or with

reinforcement learning methods that exhibit large variance, including uncertainty in the learning rate model might be useful. This could be done, for example, by introducing uncertainties for the parameters of the model or adding a noise term to it. Moreover, the current learning rate was assumed to be independent of task parameters, which was found to be a valid approximation in our experiments. However, the model could be parametrized with respect to task parameters, even though estimation of this higher-order model would require more data, decreasing its usefulness. Altogether, the task-independent learning rate model was found to be a good trade-off between model complexity and usefulness.

During learning, the individual intermediate policies (samples in D) have not yet converged to optima. Thus, fitting the skill model using some of these samples, especially ones that have not been updated for a long time, may be counterproductive in improving the CSM. This problem has also been reported in the context of supervised learning [35], [36]. We avoided this problem by dividing the task space into several regions and choosing the most recent sample from each region. However, the issue how to use intermediate samples warrants further research.

In this work, the proposed framework was demonstrated with generalized linear policy and skill models and model-free policy search. However, the framework is agnostic to the type of policy representation, contextual skill model, and policy search. A main assumption behind the framework, exponential-type convergence of rewards over time, is typical for reinforcement learning. Therefore, it appears appealing to study the application of the framework in other contexts to address in part the challenge of how to guide exploration in reinforcement learning.

REFERENCES

[1] S. ZOLA-MORGAN and L. R. Squire, "The neuropsychology of memory: Parallel findings in humans and nonhuman primates a," *Annals of the New York Academy of Sciences*, vol. 608, no. 1, pp. 434–456, 1990.

[2] S. Zola-Morgan and L. R. Squire, "Preserved learning in monkeys with medial temporal lesions: Sparing of motor and cognitive skills," *Journal of Neuroscience*, vol. 4, no. 4, pp. 1072–1085, 1984.

[3] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[4] F. C. Bartlett and F. C. Bartlett, *Remembering: A study in experimental and social psychology*, vol. 14. Cambridge University Press, 1995.

[5] J. H. Flavell, "Piaget's legacy," *Psychological Science*, vol. 7, no. 4, pp. 200–203, 1996.

[6] R. S. Michalski, "Incremental learning of concept descriptions: A method and experimental results," 1988.

[7] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 610–616, IEEE, 2013.

[8] F. Stulp, G. Raiola, A. Hoarau, S. Ivaldi, and O. Sigaud, "Learning compact parameterized skills with a single regression," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 417–422, IEEE, 2013.

[9] D. Forte, A. Gams, J. Morimoto, and A. Ude, "On-line motion synthesis and adaptation using a trajectory database," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1327–1339, 2012.

[10] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[11] B. Nemec, R. Vuga, and A. Ude, "Efficient sensorimotor learning from multiple demonstrations," *Advanced Robotics*, vol. 27, no. 13, pp. 1023–1031, 2013.

[12] A. G. Kupcsik, M. P. Deisenroth, J. Peters, and G. Neumann, "Data-efficient generalization of robot skills with contextual policy search," in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[13] G. Neumann *et al.*, "Variational inference for policy search in changing situations," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 817–824, 2011.

[14] M. Hazara and V. Kyrki, "Speeding up incremental learning using data efficient guided exploration," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, IEEE, 2018.

[15] M. Hazara and V. Kyrki, "Model selection for incremental learning of generalizable movement primitives," in *18th IEEE International Conference on Advanced Robotics (ICAR 2017), Hong Kong*, 2017.

[16] M. Hazara and V. Kyrki, "Transferring generalizable motor primitives from simulation to real world," *IEEE Robotics and Automation Letters*, pp. 1–1, 2019.

[17] B. Da Silva, G. Konidaris, and A. Barto, "Active learning of parameterized skills," in *International Conference on Machine Learning*, pp. 1737–1745, 2014.

[18] J. Lundell, M. Hazara, and V. Kyrki, "Generalizing movement primitives to new situations," in *Conference Towards Autonomous Robotic Systems*, pp. 16–31, Springer, 2017.

[19] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural Networks*, vol. 24, no. 5, pp. 493–500, 2011.

[20] A. Fabisch and J. H. Metzen, "Active contextual policy search," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3371–3399, 2014.

[21] J. H. Metzen, "Active contextual entropy search," *arXiv preprint arXiv:1511.04211*, 2015.

[22] S. Thrun, "Is learning the n-th thing any easier than learning the first?," in *Advances in neural information processing systems*, pp. 640–646, 1996.

[23] Z. Chen and B. Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 10, no. 3, pp. 1–145, 2016.

[24] G. Fei, S. Wang, and B. Liu, "Learning cumulatively to become more knowledgeable," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1565–1574, ACM, 2016.

[25] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in neural information processing systems*, pp. 1547–1554, 2003.

[26] J. Platt, *A resource-allocating network for function interpolation*. PhD thesis, MIT Press, 1991.

[27] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

[28] K. Chatzilygeroudis, R. Rama, R. Kaushik, D. Goepp, V. Vassiliades, and J.-B. Mouret, "Black-box data-efficient policy search for robotics," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 51–58, IEEE, 2017.

[29] J. Kober and J. R. Peters, "Policy search for motor primitives in robotics," in *Advances in neural information processing systems*, pp. 849–856, 2009.

[30] J. Peters, K. Mülling, and Y. Altun, "Relative entropy policy search.," in *AAAI*, pp. 1607–1612, Atlanta, 2010.

[31] E. Theodorou, J. Buchli, and S. Schaal, "Learning policy improvements with path integrals," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 828–835, 2010.

[32] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," *arXiv preprint arXiv:1206.4621*, 2012.

[33] L. P. Zhao and R. L. Prentice, "Correlated binary regression using a quadratic exponential model," *Biometrika*, vol. 77, no. 3, pp. 642–648, 1990.

[34] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63–71, Springer, 2003.

[35] J. Baldridge and M. Osborne, "Active learning and the total cost of annotation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[36] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.