Towards Robust Skill Generalization: Unifying LfD and Motion Planning

Muhammad Asif Rana, Mustafa Mukadam, S. Reza Ahmadzadeh, Sonia Chernova and Byron Boots
Institute for Robotics & Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, USA
{asif.rana, mmukadam3, reza.ahmadzadeh, chernova, bboots}@gatech.edu

***Abstract*—We present a novel unifying approach to conventional learning from demonstration (LfD) and motion planning using probabilistic inference for skill reproduction. We also provide a new probabilistic skill model that requires minimal parameter tuning, and is more suited for encoding skill constraints and performing inference in an efficient manner. Preliminary experimental results using real-world data are presented.**

I. Introduction

Most existing LfD approaches model a skill by encoding the given set of trajectory demonstrations using either a probabilistic representation [2, 9, 7] or a dynamical system [8, 4]. However, widely used approaches including probabilistic movement primitives [7] and dynamic movement primitives [8], require significant parameter tuning to learn a good skill model. Moreover, probabilistic skill models like Gaussian mixture models [2] and LfD by averaging trajectories [10] usually do not encode the variations and correlations in higher-order state variables (velocities, accelerations and so on), imposed by the skill and the trajectory dynamics.

Skill reproduction methods do not perform well either, when subjected to changes in the initial or goal state of the robot or the environment. Probabilistic approaches [2][9] tend to reproduce an average form of the movement regardless of the the initial state. Dynamical system approaches [8][4] can generalize to initial and final pose but require significant parameter tuning to preserve the shape of the movement. Furthermore, most of the above-mentioned approaches are not designed to deal with obstacles [2, 9], or even if they do, the generated path is not guaranteed to follow any specific trajectory since they only preserve reaching an end goal [8, 4, 7].

To improve the ability of LfD techniques to appropriately respond to obstacles, recent work has considered the integration of LfD and motion planning [11] in a hierarchical fashion. In this approach, the LfD solution is re-adapted in a new environment using a motion planner as a post ad-hoc step. However, there is a certain level of redundancy induced by making a strong structural assumption that the two steps are independent. Because, at the heart of both LfD and planning lies a need to find a trajectory that follows some prior model (either known beforehand or learned from data) given any other costs, constraints and/or observations.

Our approach, 1) learns a skill model which extracts the variations in the demonstrations, considers inherent trajectory dynamics while requiring minimal parameter tuning, and then, 2) reproduces a continuous-time trajectory, $\boldsymbol{\theta}(t)$, that is *optimal* with respect to the learned skill model, while remaining *feasible* when subjected to different situations or environments.

II. Finding Trajectories with Probabilistic Inference

Within the context of finding trajectories that are optimal and feasible, the reproduction problem inherently has the same motivations as motion planning. However, motion planning in contrast defines optimality with hard coded metrics (that are known a priori instead of being learned), for example, smooth accelerations. We adopt the probabilistic inference perspective on motion planning [3], because it naturally allows us to incorporate optimality metrics that can then instead, be learned from demonstrations in the form of a prior i.e. a skill model.

Within this view, we seek to find the posterior distribution of the trajectory, $p(\boldsymbol{\theta}|\mathrm{e}) \propto p(\boldsymbol{\theta})p(\mathrm{e}|\boldsymbol{\theta})$, given some random binary events, $\boldsymbol{e}$, that can represent a combination of various events, for example, collision avoidance, starting from a given location, reaching a desired goal location or a via-point, etc. Here,

1) The Prior: $p(\boldsymbol{\theta}) \propto \exp\{-\frac{1}{2}||\boldsymbol{\theta} - \boldsymbol{\mu}||^2_{\boldsymbol{\mathcal{K}}}\}$, defines optimality and is learned from demonstrations. This can also be interpreted as learning the hyper-parameters (mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\mathcal{K}}$) of the trajectory distribution. We use structured Gaussian processes to model the prior [1] (see the next section for details).

2) The Likelihood: $p(\mathrm{e}|\boldsymbol{\theta}) \propto \exp\{-\frac{1}{2}||\boldsymbol{h}(\boldsymbol{\theta}, \mathrm{e})||^2_{\boldsymbol{\Sigma}}\}$, encodes feasibility and defines the probability of the events occurring given the trajectory. We model this as a distribution in the exponential family [3].

We can find the *maximum a posteriori* (MAP) trajectory i.e. the mode of the posterior distribution through inference

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{p(\boldsymbol{\theta})p(\mathrm{e}|\boldsymbol{\theta})\right\} \tag{1}$$

and get the desired trajectory that is optimal and feasible. Our key insight is that skill reproduction in any new scenarios is in fact equivalent to performing planning as inference.

Following [3], we use factor graphs [5] to represent distributions and use the duality between inference and optimization to arrive at a fast and efficient approach that solves (1).

III. Trajectory Prior as Skill Model

We model the prior using Gaussian processes (GPs) such that for any collection of times $\boldsymbol{t} = \{t_0, \ldots, t_N\}$, $\boldsymbol{\theta}$ has a joint Gaussian distribution, $\boldsymbol{\theta} \doteq [\boldsymbol{\theta}_0 \quad \dots \quad \boldsymbol{\theta}_N]^\top \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\mathcal{K}})$. GPs have been used in LfD before [10], however the choice of structured GPs here, that can be generated via stochastic differential equations (SDE), is more suited for modeling continuous-time trajectories for the skills. This structured GP not only takes care of the higher-order system dynamics (velocities, accelerations etc.), but also allows us to exploit the sparsity of the underlying problem such that both learning and inference are computationally efficient.

A. Structured Heteroscedastic GP From LTV-SDE

To learn the prior from demonstrations, we consider a SDE with a white noise process, $\mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_C(t)\delta(t-t'))$. Unlike [1], the covariance $\mathbf{Q}_C(t)$ is considered time-varying, to generate a heteroscedastic GP which is more expressive to learn skills. Taking the first and second order moment of the solution to the linear time-varying SDE (LTV-SDE), $\dot{\boldsymbol{\theta}}(t) = \mathbf{A}(t)\boldsymbol{\theta}(t) + \mathbf{u}(t) + \mathbf{F}(t)\mathbf{w}(t)$, provides the desired GP. Here, $\mathbf{A}(t)$ and $\mathbf{F}(t)$ are time-varying system matrices, $\mathbf{u}(t)$ is a bias term, and the state can comprise of the configuration and any higher order time derivatives. The inverse covariance matrix of this GP has a sparse block diagonal structure that is exploited to make learning and inference efficient.

In case of manipulators where demonstrations of end-effector trajectories are generally suitable to represent a particular skill, we can model the LTV-SDE on the end effector's state. We estimate the hyper-parameters of the prior distribution by considering the provided demonstrations as solutions to this LTV-SDE.

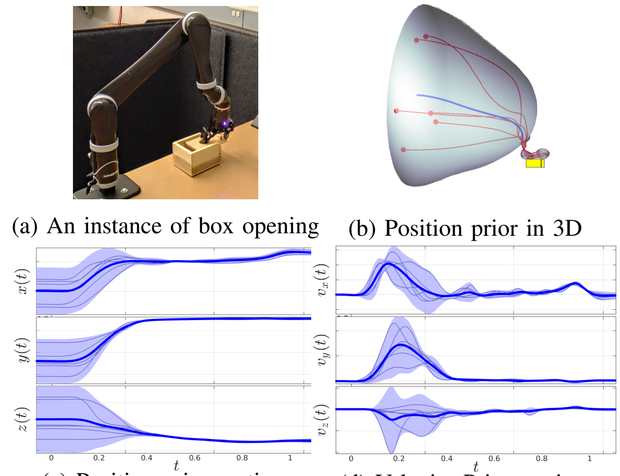
B. Learned Prior

We use end-effector trajectory demonstrations to learn a prior, $p(\mathbf{x}|\boldsymbol{\theta}) \propto \exp\{-\frac{1}{2}||\mathbf{C}(\boldsymbol{\theta}) - \boldsymbol{\mu}_\mathbf{x}||^2_{\boldsymbol{\mathcal{K}}_\mathbf{x}}\}$, where $\mathbf{x}$ is the end effector trajectory that can be transformed from the trajectory in configuration space, $\boldsymbol{\theta}$, with a mapping $\mathbf{C}$. For the skills we consider and ease of implementation, a discrete version of the LTV-SDE above, proved sufficient to learn $\boldsymbol{\mu}_\mathbf{x}$ and $\boldsymbol{\mathcal{K}}_\mathbf{x}$. We learn these parameters by performing maximum likelihood estimation with ridge regression. Our current implementation supports an end effector state composed of 3D positions and linear velocities i.e. we ignore orientation and angular velocities. Note that the learned prior requires no parameter tuning and is directly available for inference.

Manipulators in general have high degrees of freedom. So during inference, to prevent our formulation from being under constrained, we also incorporate a smoothness prior in configuration space, $p(\boldsymbol{\theta}) \propto \exp\{-\frac{1}{2}||\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}}||^2_{\boldsymbol{\mathcal{K}}_{\boldsymbol{\theta}}}\}$, analogous to the homoscedastic GP prior used in [6, 3]. Specifically, we use the constant velocity prior that encourages smooth acceleration. The state in configuration space comprises of position and velocities for all the robot joints. When reproducing skills, a joint prior $p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})$ is utilized.

IV. Experiments

We validate the proposed method on a *box-opening* skill using a Kinova JACO^2 6-DOF arm. We provided 6 demonstrations through kinesthetic teaching with varying initial end-effector states (varying initial position, zero initial velocity) and aligned them using dynamic time warping (DTW). The temporal data and end-effector positions were recorded and the corresponding end-effector linear velocities were estimated by fitting a cubic spline and differentiating with respect to time. Fig. 1 shows the learned prior distribution i.e. the skill model.



(a) An instance of box opening (b) Position prior in 3D

(c) Position prior vs time (d) Velocity Prior vs time

Fig. 1: Prior for *box-opening* skill. The mean is in blue with 95% confidence interval.

(a) without obstacle (b) with obstacle

Fig. 2: Reproduced trajectories in red from different initial states. The obstacle is in yellow and the prior position mean is in blue.

Each demonstration is composed of two primitive actions, reaching and sliding the lid of the box. The sliding part of the skill is more restrictive compared to the reaching part. Hence, as shown in Fig. 1 (c)-(d), the variance in the state variables (i.e. positions and velocities) becomes much smaller during the sliding part. It should be noted here that the prior also encodes the coupling between the state variables.

Fig. 2 shows the reproduced MAP trajectories. For reproduction with different initial states, trajectories were found by performing inference (1) where the likelihood contained the observation for the given initial state with a very small $\mathbf{\Sigma}$ since we are certain about the initial state. For reproduction in a new environment with obstacles, the likelihood also contained the collision-free likelihood from [3], where the obstacle cost is evaluated using a precomputed signed distance field. Here we set the parameter $\mathbf{\Sigma}$ manually such that it enables desired clearance of the robot from the obstacle. In general, $\mathbf{\Sigma}$ depends on the size of the robot, desired clearance and the environment itself. As noticeable from Fig. 2, the robot is still able to reproduce the learned skill from various initial states and in the presence of unknown obstacles in the environment.

REFERENCES

[1] Sean Anderson, Timothy D. Barfoot, Chi Hay Tong, and Simo Särkkä. Batch nonlinear continuous-time trajectory estimation as exactly sparse gaussian process regression. *Autonomous Robots*, 39(3):221–238, 2015.

[2] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007.

[3] Jing Dong, Mustafa Mukadam, Frank Dellaert, and Byron Boots. Motion planning as probabilistic inference using Gaussian processes and factor graphs. In *Proceedings of Robotics: Science and Systems (RSS-2016)*, 2016.

[4] S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[5] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

[6] Mustafa Mukadam, Xinyan Yan, and Byron Boots. Gaussian process motion planning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9–15, May 2016.

[7] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. Probabilistic movement primitives. In *Advances in neural information processing systems*, pages 2616–2624, 2013.

[8] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE, 2009.

[9] Benjamin Reiner, Wolfgang Ertel, Heiko Posenauer, and Markus Schneider. Lat: A simple learning from demonstration method. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4436–4441. IEEE, 2014.

[10] Markus Schneider and Wolfgang Ertel. Robot learning by demonstration with local gaussian process regression. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 255–260. IEEE, 2010.

[11] Gu Ye and Ron Alterovitz. Demonstration-guided motion planning. In *International symposium on robotics research (ISRR)*, volume 5, 2011.