Fast Learning and Sequencing of Object-centric Manipulation Skills

Leonel Rozo, Andras Kupcsik, Meng Guo, Marco Todescato,
Philipp Schillinger, Nicolai Waniek, Markus Giftthaler, Mathias Bürger
Bosch Center for Artificial Intelligence
Robert-Bosch-Campus 1, Renningen, 71272 – Germany
Email: leonel.rozo@de.bosch.com

***Abstract*—Enabling robots to quickly learn manipulation skills is an important and yet challenging problem. In industrial settings, this is potentially even more challenging given the complexity of tasks involved. Ideally, such manipulation skills should be flexible in order to adapt either to the actual configurations of the objects and of the robot or to new tasks. Furthermore, to accomplish complex manipulation tasks, the robots, rather than learning very complex skills, should be able to sequence several simple skills. In this work, we propose to sequence generic robot manipulation skills encoded by learning from demonstration (LfD) to achieve complex manipulation tasks. Our sequencing approach smoothly blends manipulation skills together, thus humans only need to demonstrate simpler skill primitives. This reduces the cognitive load on the human demonstrator, as kineshtetic teaching of complex manipulation skills is prone to errors. Given a task goal and a high-level plan (possibly provided by a human operator or by a high level planner), our method generates smooth robot trajectories (in task space) to complete the manipulation objective.**

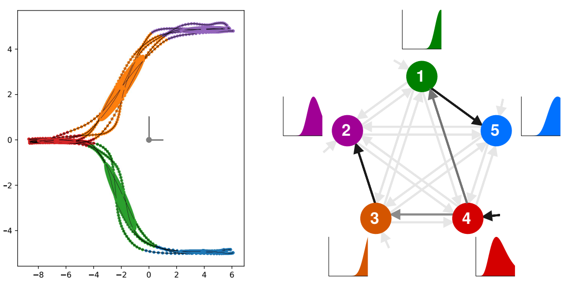

Fig. 1: **Left:** Learned 5-states HSMM for skill "grasp" in 2-D, where demonstration data are labeled in color by the associated states. **Right:** transition and duration functions of the HSMM. Arrows color intensity is proportional to the learned transition probability, where black and light gray respectively depict high and low probabilities.

I. Introduction

Deploying service robots in daily household environments or in highly flexible manufacturing sites is promising, but also highly challenging [1]. The challenges arise in many different sub-fields of robotics, e.g., perception [3], motion planning [7], mapping and navigation [8], human-robot interaction [5]. In this work, we address two of these challenges.

First, it is almost impossible for robot manufacturers to pre-program *all* robot capabilities (referred to as *skills*) that final users may potentially require from the robot. Hence, to avoid inquiring engineers whenever a new skill is needed, it is crucial to provide an easy and efficient method with which laymen can teach the robot new skills. Simply recording and replaying a demonstrated trajectory is often insufficient because changes in the environment, such as varying robot and/or object poses, would render any attempt unsuccessful. In other words, the robot needs to *recognize* and *encode* the intentions behind these demonstrations and, more importantly, to generalize over unforeseen situations.

Many learning-from-demonstration (LfD) frameworks have shown great improvements in this aspect. Compared to hard-coded alternatives, they embed extracted knowledge into probabilistic models. Examples are probabilistic motion primitives (ProMPs) [9], Stable Estimators of Dynamical Systems (SEDS) [6] and Task-Parameterized Gaussian Mixture Models (TP-GMMs) [2]. However, most of these approaches train models specifically for each skill instantiation. For instance, "grasp the work piece from the top" and "grasp the work piece from the side" are usually treated as two separate skills and, thus, different models are trained. Clearly, this not only greatly decreases the teaching efficiency, but also significantly limits the *reusability* of each skill.

In contrast, one would like to learn general and potentially reusable skills. However, the more general each skill is, the more challenging it is to sequence skills in an appropriate way. For example, a general grasping skill would be able to grasp an object from different sides, but different instantiations of this skill are needed depending on the following actions.

To tackle this problem, we propose a method to adapt skill parameters given a *sequence* of skills. This fusion significantly reduces human teaching and modeling efforts, while ensuring high degree of flexibility and re-usability of learned skills.
Hence, the main contributions of this paper are:

- (i) to enable robot learning of general skills by extending existing work on task-parameterized probabilistic models.
- (ii) to present a novel algorithm for the sequencing of several skills to fulfill, with maximum probability of successful task completion, a given high-level task plan.

II. Skill Learning

In this work we combine two probabilistic modeling approaches to learn flexible skills: Task Parametrized Gaussian Mixture Models (TP-GMMs) and Hidden semi-Markov Models (HSMM). The former are a variation of standard GMMs where a set of N demonstrations of lenght T, $\{\{\boldsymbol{\xi}_{t,i}^{(p)}\}_{t=1}^{T}\}_{i=1}^{N}$, are recorded from P different coordinate systems of interest and used to jointly learn a model $\{\pi_k, \{\boldsymbol{\xi}_k^{(p)}, \boldsymbol{\Sigma}^{(p)}\}_{p=1}^{P}\}_{k=1}^{K}$ with K the number of components, π_k the mixing coefficients and $\{\boldsymbol{\xi}_k^{(p)}, \boldsymbol{\Sigma}^{(p)}\}_{p=1}^{P}$ the parameters the k-th component within frame p. The latter, previosuly used for speech synthesis [12], extend standard HMMs by embedding termporal information of the underlying stochastic process.

A. Task-Parametrized Hidden Semi-Markov Models

Recently, TP-HSMMs have been successfully used for robot skill encoding to learn spatio-temporal features of the demonstrations [10]. More specifically, a TP-HSMM consists of

$$\mathcal{M}_{\boldsymbol{\theta}} = \Big\{ \{a_{kh}\}_{h=1}^{K}, (\mu_k^D, \sigma_k^D), \pi_k, \{\boldsymbol{\mu}_k^{(p)}, \boldsymbol{\Sigma}_k^{(p)}\}_{p=1}^{P} \Big\}_{k=1}^{K} \quad (1)$$

where a_{kh} is the transition probability from state k to h; (μ_k^D, σ_k^D) describe the Gaussian distributions for the duration of state k, i.e., the probability of staying in k for a certain number of steps; $\{\pi_k, \{\boldsymbol{\mu}_k^{(p)}, \boldsymbol{\Sigma}_k^{(p)}\}_{p=1}^{P}\}_{k=1}^{K}$ are the parameters of the TP-GMM for each state k and each task-relevant coordinate system p (e.g., object poses, world coordinate, etc.). The GMM components describe the emission probability encoding N demonstrated trajectories $\{\{\boldsymbol{\xi}_{t,i}\}_{t=1}^{T}\}_{i=1}^{N}$ consisting of robot end-effector poses and robot hand states.

As shown in [10], the probability of data point $\boldsymbol{\xi}_t$ belonging to state k (i.e., $s_t = k$) is given by the *forward* variable $\alpha_t(k) = p(s_t = k, \{\boldsymbol{\xi}_\ell\}_{\ell=1}^{t})$:

$$\alpha_t(k) = \sum_{\tau=1}^{t-1} \sum_{h=1}^{K} \alpha_{t-\tau}(h) a_{hk} \, \mathcal{N}(\tau | \mu_k^D, \sigma_k^D) \, o_\tau^t \,, \qquad (2)$$

where $o_\tau^t = \prod_{\ell=t-\tau+1}^{t} \mathcal{N}(\boldsymbol{\xi}_\ell | \hat{\boldsymbol{\mu}}_{\ell,k}, \hat{\boldsymbol{\Sigma}}_{\ell,k})$ is the emission probability. Furthermore, the same forward variable can also be used during reproduction to predict future steps until T. In this case however, since future observations are not available, only transition and duration information are used by setting $\mathcal{N}(\boldsymbol{\xi}_\ell | \hat{\boldsymbol{\mu}}_{\ell,k}, \hat{\boldsymbol{\Sigma}}_{\ell,k}) = 1$ for all k and $\ell > t$ in (2) [11]. At last, the sequence of the most-likely states $\boldsymbol{s}_T^\star = s_1^\star s_2^\star \cdots s_T^\star$ is determined by choosing $s_t^\star = \arg\max_k \alpha_t(k)$, $\forall 1 \leqslant t \leqslant T$. The future state sequence can be used in an optimal control problem to retrieve the smooth robot trajectory, which can be executed on the real robot.

III. Sequencing TP-HSMMs

TP-HSMMs are able to encode multi-modal demonstrations, and thus, more general skills. For example, “grasp the work piece from the top” and “grasp the work piece from the side” are two modes of the “grasp the work piece” skill (Fig. 1).

Given the model of a single skill, the forward variable provides the most likely state sequence until its termination. However, when dealing with a sequence of skills (from a human operator or from a task planner) we have to (i) cascade all TP-HSMMs into a “super-HSMM” of the complete manipluation skill and (ii) find the most likely state sequence of the super-HSMM, from initial state $\boldsymbol{\xi}_0$ to final state $\boldsymbol{\xi}_T^d$ at time T. Note, that executing skills independent of each other (by rolling out the forward variable) does not guarantee that the final robot state is $\boldsymbol{\xi}_T^d$.

Since the transition from one skill to the next is never demonstrated, we propose to compute such transition probabilities from the $KL(\cdot||\cdot)$ divergence of emission probabilities between the sets of final and starting states. Particularly, consider two consecutive skills $\mathsf{a}_d^\star$ and $\mathsf{a}_{d+1}^\star$ in $\mathbf{a}_D^\star$. The transition probability from one final state f of the first skill to one starting state i of the consecutive skill is given by

$$a_{fd} \propto \exp\Big(-\alpha \cdot \sum_{p \in P} KL\big(\mathcal{N}(\boldsymbol{\mu}_f^{(p)}, \boldsymbol{\Sigma}_f^{(p)}) || \mathcal{N}(\boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)})\big)\Big),$$

with blending constant $\alpha \in \mathbb{R}^+$. This process is repeated for all pairs of starting and final states between consecutive skills in $\mathbf{a}_D^\star$.

Note that the forward variable introduced in the previous section allows us to compute the sequence of *marginally* most probable states, while we are looking for the *jointly* most probable sequence of states given the last observation $\boldsymbol{\xi}_T^d$. As a result, when using (2) there is no guarantee that the returned sequence $\boldsymbol{s}_T^\star$ will match both the spatio-temporal patterns of the demonstrations and the final observation. To overcome this issue, we rely on a modification of the Viterbi algorithm presented in [4] which (i) works on HSMM instead of HMM; and more importantly (ii) most observations, except first and last, are missing. Specifically, in the absence of observations the Viterbi algorithm becomes

$$\begin{aligned} \delta_t(j) &= \max_{d \in \mathcal{D}} \max_{i \neq j} v_{t-d}(i) a_{ij} p_j(d) \prod_{t'=t-d+1}^{t} \tilde{b}_j(\boldsymbol{\xi}_{t'}) \,, \\ \delta_1(j) &= b_j(\boldsymbol{o}_1) \pi_j p_j(1), \end{aligned}$$

where

$$\tilde{b}_j(\boldsymbol{\xi}_{t'}) = \begin{cases} \mathcal{N}(\boldsymbol{\xi}_{t'} | \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j) & , t = 1 \lor t = T \\ 1 & , 1 < t < T \,. \end{cases}$$

At each time t and for each state j, the arguments maximizing $\delta_t(j)$ are recorded and a simple backtracking procedure can be used to find the most probable state sequence $\boldsymbol{s}_T^\star$.

IV. Summary

In this work we have shown how to combine probabilistic modeling and, in particular, TP-HSMM with a modified Viterbi algorithm for flexible skill learning and sequencing. To conclude, it is worth highlighting the importance of skills duration encoding naturally provided by the HSMM when skills sequencing is required. Indeed, the duration probabilities of the HSMM encode the temporal patterns observed across demonstrations of a single skill, for all its different instantiations. This temporal information is nicely exploited by the Viterbi algorithm to provide a sequence of states, consistent with the demonstrations. This would not be possible if explicit duration model are not used, as for standard TP-GMMs or TP-HMMs. In this latter case, the most probable states sequence would force the robot to visit each HMM state one single time step regardless the demonstrated temporal patterns, as the algorithm is only given the initial and final observations.

REFERENCES

[1] Sandra Bedaf, Patrizia Marti, Farshid Amirabdollahian, and Luc de Witte. A multi-perspective evaluation of a service robot for seniors: the voice of different stakeholders. *Disability and Rehabilitation: Assistive Technology*, pages 1–8, 2017.

[2] Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, 2016.

[3] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[4] G David Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[5] Michael A Goodrich, Alan C Schultz, et al. Human–robot interaction: a survey. *Foundations and Trends® in Human–Computer Interaction*, 1(3):203–275, 2008.

[6] 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.

[7] Steven M LaValle. *Planning Algorithms*. Cambridge university press, 2006.

[8] Don Murray and James J Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, 2000.

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

[10] Ajay Kumar Tanwani and Sylvain Calinon. Learning Robot Manipulation Tasks with Task-Parameterized Hidden Semi-Markov Model. *IEEE Robotics and Automation Letters*, pages 1–8, 2016.

[11] Shun-Zhen Yu and Hisashi Kobayashi. A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking. *Signal Processing*, 83(2):235–250, 2003.

[12] Heiga Zen, Keiichi Tokuda, Takashi Masuko, Takao Kobayasih, and Tadashi Kitamura. A Hidden Semi-Markov Model-Based Speech Synthesis System. *IEICE Transactions on information and systems*, 90(5):825–834, 2007.