Learning Constraints with Keyframes

Baris Akgun, Andrea Thomaz
Georgia Institute of Technology
Email: {bakgun3, athomaz}@cc.gatech.edu

I. Extended Abstract

One way to combine high-level task descriptions with low-level motion execution, is to learn them concurrently and figure out the relationship between them. We approach the problem from a Learning from Demonstration (LfD) perspective. We are interested in scenarios where the robot user will not have robotics background such as in a factory floor, hospital or at home, whom we call end-users. In our previous work on learning low-level motion execution with end-users [2, 3], we observed they were trying to communicate *what to do*,in a task more than *how to do* it, *i.e.* , they were more concerned on completing the task successfully rather then providing clean, consistent and noise free demonstrations. The *what to do* part can be interpreted as (sub-)goals and *how to do* part as the low-level motion to achieve them. We can relate (sub-)goals as constraints since, if certain constraints are satisfied, then the task is achieved. Based on these observations, we propose a buttom-up approach to learning constraints and motions that satisfy these constraints from human demonstrations.

In [2], we proposed keyframe demonstrations, which are temporally sparse set of points instead of full trajectories which satisfy the task when connected together. We argue that keyframes are easier for end-users to provide and result in *cleaner* data. Keyframes end up being explicit salient points of the task which would serve well as being point constraints. However, trajectories are needed where the low-level motion is non-linear (*e.g.* scooping) or has dynamic components (*e.g.* throwing). As such, it is not easy to extract continuous constraints (*e.g.* keeping a mug straight when transporting it) just from keyframes. In [2], we also proposed hybrid demonstrations, in which, the user is free to mix keyframes and trajectories in the context of a single demonstration. The trajectory portions of the task can be used to extract continuous constraints. In [1], we proposed a learning framework, called Keyframe Learning from Demonstration (KLfD), for keyframe, trajectory and hybrid demonstrations.

Constraints can be in the task space (*e.g.* keep end-effector close a certain point) as well as the perceptual space. For example in scooping coffee beans, the aim is to have some coffee beans inside a spoon after the execution. If a camera is used, the color change of the spoon (*e.g.* from the original color to the color of the coffee beans) can represent a constraint. Due to the stochastic nature of real environments, perception plays an important role, *e.g.* to know whether a constraint is satisfied or not and if it is not, help error recovery. In this work, we augment this framework to learn action and perception models of a task demonstrated with keyframes , evaluate it with end-users and discuss potential future directions.

In the original KLfD work, a single keyframe was the state of the robot's end effector. In our approach, we also collect a perceptual keyframe sequence. In this section we describe (1) gathering demonstrations and (2) learning action and perception models, and (3) using the learned model.

The user provides demonstration via *kinesthetic teaching*, whereby the teacher physically manipulates the robot.The system records a motor keyframe and a perceptual keyframe at each keyframe location. The motor keyframe is a typical state representation in LfD skill learning, end-effector pose with respect to the object of interest. The perceptual keyframe is a *perceptual snapshot* of the workspace, taken with an overhead camera. Background subtraction and known object hue ranges are used to segment the image and extract features

We use the motor keyframe sequence to learn an action model of the skill, and the perceptual keyframe sequence to learn a perceptual model of the skill. We use a Hidden Markov Model (HMM) approach since HMMs provide a probabilistic framework which we leverage. We use the same learning algorithm on both the motor keyframes and the perceptual keyframes to learn action models and perceptual models respectively. We model the emissions as multivariate Gaussian distributions on the corresponding space. We train our HMM with the Baum-Welch algorithm and use Bayesian Information Criterion to decide on the number of states.

We expect to learn *meaningful* state representations in the perceptual space. A state with a high prior but with a low incoming transition probability is an initial condition of a skill. A state with a high incoming probability and outgoing probability can be interpreted as a sub-goal of the skill. A state with low outgoing transition probability and high incoming transition probability acts like a terminal state and can be interpreted as a goal of the skill. These notions are depicted in Fig. 1. These individual states can be interpreted as constraints, thus these HMMs model both the pose and perceptual constraints, their ordering (via transition probabilities) and their strength (via the emission probabilities, if variance is low then the state is a "harder" constraint).

In the execution phase, a keyframe sequence is generated by using the motor skill model. This is done by selecting a state with a non-zero prior and transitioning to the next most likely state (by following the probabilities in the transition matrix) until a goal state is encountered. Then, the means of the corresponding emission probabilities define an end-effector pose sequence to be followed in the motor space. The robot

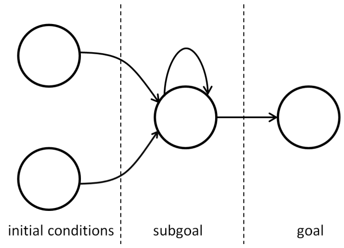

Fig. 1. A typical HMM resulting from the learning process described.

extracts a perceptual keyframe at these poses which results in a sequence of perceptual keyframes for the execution, which is then tested against the learned perceptual skill model.

A sequence of a short length can have a high likelihood score but it might not be enough to complete the task, *e.g.* observing an incomplete portion of a task as a full execution would yield a high likelihood. On the other hand, it is not enough to just check whether the end-state is a goal state for all the skills. Sub-goals of the skill might be important to achieve but not be visible at the end state. We use a two step procedure to decide whether an observed sequence is a failed execution or not. First, we check if this sequence is likely to be from the HMM of the skill. If so, we check whether the last keyframe of this sequence is likely to be a goal state or not. We utilize thresholds on log-likelihood to decide on whether a sequence is from an HMM and whether a state is a goal state.

We collected data from 8 users, none of whom had prior experience interacting with a humanoid robot. We use two skills our evaluation are: *Close the box*: This skill requires to teach the robot to close the box. *Place*: This skill requires the placement the red semi-circular block and the green block in a certain configuration in front of the robot. There are 4 different initial conditions (*e.g.* 4 different object locations) per skill.

In our qualitative analysis, we visualize the resulting HMMs. When we look at the motor models, they do not match across the users (*e.g.* , closing the box back handed vs. a forward facing hand). However, their perceptual models are very similar. Moreover, when we pool their data, we can learn a consistent perceptual model, but the motor model does not make sense, due to the widely varying motor strategies.

We performed hold-one-out cross validation, training a perceptual skill model with the 28 demonstrations from 7 users, holding out the demonstrations of the 8th user for testing, and averaging the performance. The held out demonstrations are used to test the model in the case of success. Since all the demonstrations were positive examples of skill completion, all four demonstrations can be used as cases where the HMM should classify the observation as successful.

To test the model's ability to detect the case when the skill fails, we manipulate the demonstrated sequences to create failure cases. Recall that the success/failure decision has two components; checking if the observed sequence is likely to be from the HMM and checking if the observation terminated at a goal state. Thus we manipulate each of the test data to create observation sequences that fail in one or the other of these two cases; unfinished sequence and unlikely sequence.

The close the box skill, has typically shorter sequences, which results in higher likelihoods per sequence. As the threshold is relaxed, more failed data with correct ends are classified as coming from the learned HMM. The best threshold yields up to 95% correct classification on the data for both of the failure cases and 100% for the cross-validation.

The place skill has longer sequences, thus only ending in the correct state can be classified as failure easily, thus we have 100% accuracy in the unlikely sequence failure case. However there is also relatively high variance on the end-state. Thus, our failure detection with unfinished sequence results are around 80%. Cross validation results are around 90%.

Finally we analyze the performance of the perceptual models by using them in conjunction with the motor skill models. For this analysis we use the *close the box* skill model for two different users. The uncertainties in the series elastic actuators, object location estimation *etc.* make the skill execution results non-deterministic. We run each model 10 times and present the results in Table I. Both perceptual models performed well at the execution monitoring task. Particularly given that only 4 demonstrations were used to train the perceptual models. Subject 2 has a less successful skill model, that only succeeded in 2 out of 10 execution trials. This user's perceptual model was able to capture both the successes and the failures. Subject 1 on the other hand, had a more successful skill model, succeeding in 7 out of 10 execution trials. This user's perceptual model failed to classify 1 of the failures, which was hard for the perceptual features to distinguish.

These results are promising for future work to learn and monitor and combine motor and perceptual skill models and constraints for LfD. By grouping frequent constraints and their relations, a high-level representation can be created from low-level demonstrations which could lead to learning and using both low-level and high-level representations in parallel.

TABLE I

CLASSIFICATION RESULTS FOR ROBOT SELF-MONITORING.

		Success	Fail
Subject 1	Classified Success	7	1
	Classified Fail	0	2
Subject 2	Classified Success	2	0
	Classified Fail	0	8

REFERENCES

[1] Baris Akgun, Maya Cakmak, Karl Jiang, and L. Andrea Thomaz. Keyframe-based learning from demonstration: Method and evaluation. *International Journal of Social Robotics*, 2012.

[2] Baris Akgun, Maya Cakmak, Jae Wook Yoo, and L. Andrea Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *HRI*, pages 391–398, 2012.

[3] Baris Akgun, Kaushik Subramanian, and Andrea Thomaz. Novel interaction strategies for learning from teleoperation. In *AAAI Fall Symposia*, 2012.