Robot Learning from Demonstration - a brief literature review

*

Deepak Jowel
School of Computer Science
University of Lincoln
Lincoln, U.K

***Abstract*—This paper discusses the different methodologies for a robot learning from demonstrations(LfD). In the domain of robotics, learning from demonstrations is the traditional method by which the robot acquire new skills. It can also be considered as imitating the teacher. LfD is a technique that develops policies from example state to action mappings In this paper certain methods and existing state of art methodologies have been discussed.**

***Index Terms*—Robot Learning, LfD**

I. Introduction

In the domain of robotics learning, at times the robot has to perform complicated maneuvers. These are to be considered simple for humans as they have trained their brain and built muscle memory over the years to perform these task. But to teach a robot a simple motion of even lifting a bottle placed on a table can be a tricky task. As per (Ravichandar et al., 2020), the sequence of actions or movements that are expected to be performed by the robot were requiree to be explicitly specify. Moreover in there work the authors emphasises on utilizing methods of motion planning were the teacher can eliminate the need to specify the entire sequence of low level actions, however motion planning still require the user to specify higher-level actions, such as goal locations and sequences of via points. These practices are not robust to change in environment and thus arises the need to make amend with the new scenario or create algorithms that will function aptly to those changes in the environment. The authors (Ravichandar et al., 2020) highlight that the method of LfD can benefit various industries where there is a need for the robots to execute complex maneuvers like manufacturing healthcare. (Ravichandar et al., 2020) propose Imitation learning, programming by demonstration, and behavioral cloning as other popular phrases used to describe the process of learning from demonstrations.As highted in (Argall et al., 2009) the author states that these approaches depend heavily upon the accuracy of the world model.

In (Argall et al., 2009) the authors describe learning a mapping between world state and actions as the core of many robotics applications. This mapping is also called a policy which enable the robot to select a action based upon its current state. The policies are sequence of waypoints of state-action pair. In contrast to some machine learning approaches . In their work the (Argall et al., 2009) the authors hight on the fact that the policy derived under LfD are confined to the task application and the work space. (Argall et al., 2009) states that LfD in contarst to Reinforcement Learning is a tricky task as building the policy require gathering information by visiting states to receive rewards, which is non-trivial for a robot learner executing actual actions in the real world..

LfD can be considered as a supervised learning. (Ravichandar et al., 2020) show concerns as this “ limit the performance of LfD techniques to the abilities of the teacher; to tackle this problem. LfD has its own set of challenges and limitations, these challenges include the curse of dimensionality, learning from very large or very sparse datasets, incremental learning, and learning from noisy data.From (Ravichandar et al., 2020) it was understood the platform (robot + interface) used fro teach plays a vital role in the success a learning task from demonstrations.

A. *Kinesthetic Teaching*

in (Ravichandar et al., 2020) authors describes Kinesthetic teaching as a robot learning the desired motions through demonstrate performed by physically moving the robot. The sensors present onboard the robot record the training data for machine learning. ”(Ravichandar et al., 2020) highlight That The method is popular for manipulators, including lightweight industrial robots, due to its intuitive approach and minimal user training requirements. (Argall et al., 2009) speaks about the strategy for providing data to the learner. Two such methods are batch learning and interactive approaches. For the case of, batch learning (Ravichandar et al., 2020) the policy is learned only once all data has been gathered. And also, interactive approaches allow the policy to be updated incrementally as training data becomes available.

The authors in (Ravichandar et al., 2020) write that the quality of the demonstrations depends on the dexterity and smoothness of the human user. Highlighting this limitaion and suggesting that the the data obtained requires smoothing and post-processing. Further elaborating thatkinesthetic teaching is most effective for manipulators due to their relatively intuitive form factor its applicability is limited on other platforms, such as legged robots or robotic hands, where demonstrations are more challenging to perform.

B. *Teleoperation*

This is also a demonstration technique that “has been applied to trajectory learning , task learning , grasping , and high level tasks”(Ravichandar et al., 2020). A teacher operates the robot learner platform and the robot’s sensors record the execution.(Argall et al., 2009) In this case the robot is being provided by an external input by he use of joystick, graphical user interface or other means.“teleoperation does not require the user to be copresent with the robot, allowing LfD techniques to be applied in remote settings”(Ravichandar et al., 2020).“Teleoperation is also applied to a wide variety of simulated domains, ranging from static mazes to dynamic driving and soccer domains, and many other applications.”(Argall et al., 2009). In place of a human teacher, hand-written controllers are also used to teleoperate robots. Data recorded using real robots frequently does not represent the full observation state of the teacher. if the teacher observes parts of the world that are inaccessible from the robot’s cameras. “access to remote demonstrators opens the opportunity for crowdsourcing” more data for training with possibly high variation will be ideal to teach an agent. “Limitations of teleoperation include additional effort to develop the chosen input interface some cases a more lengthy user training process,and the availability of input hardware”(Ravichandar et al., 2020).result of these efforts,be applied to more complex systems,“teleoperation can be easily coupled with simulation to further facilitate data collection and experimentation at scale, as often required within reinforcement learning (RL) frameworks.”(Ravichandar et al., 2020).This method requires an extra algorithmic component which enables the robot to track and actively shadow

C. *Passive Observations*

In (Argall et al., 2009) authors describe passive observation as the robot platform mimicing the teacher’s demonstrated motions while recording from its own sensors .Embodiment issues do exist between the teacher and learner for imitation approaches.(Ravichandar et al., 2020) states this method as easy for the demonstrators it requires almost no training to perform. Suitable for application to high-DOF or non-anthropomorphic robots”. This method can be implemented in two different ways. As mentioned below:

- **Sensor on teacher** In this technique the sensor is placed on the executing body. The record mapping is direct. (Argall et al., 2009) headlights the strength of this technique that the teacher provides precise measurements of the example execution and the overhead attached to the specialized sensors, such as human-wearable sensor-suits, or customized surroundings, such as rooms outfitted with cameras, is non-trivial and limits the applicability settings of this technique.
- **External Observation** (Argall et al., 2009) states that in this method the sensors are external and there exist no direct record mapping.

This method is suitable for instances where kinesthetic teaching is difficult. In ”(Ravichandar et al., 2020) the author highlights the limitations such as Occlusions, rapid movement, and sensor noise in the observations of human actions are some of the challenges for this type of task. The authors (Ravichandar et al., 2020) state taht despite the challenges, learning from passive observation has been successfully applied to various tasks, such as collaborative furniture assembly, autonomous driving, table-top actions, and knot tying

II. Learning from Machine Learning Methods

In (Argall et al., 2009) the author states that the goal of this type of algorithm is to reproduce the underlying teacher policy, which is unknown, and to generalize over the set of available training examples such that valid solutions are also acquired for similar states that may not have been encountered during demonstration. Mapping approximation techniques fall into two categories depending on whether the prediction output of the algorithm is discrete or continuous. Classification techniques produce discrete outputs, and regression techniques produce continuous output.

- Classification approaches categorize their input into discrete classes, thereby grouping similar input values together. In the context of policy learning, the input to the classifier is robot states and the discrete output classes are robot actions. Low-level robot actions include basic commands such as moving forward or turning. Example applications that learn a mapping from states to low-level actions include controlling a car within a simulated driving using Gaussian Mixture Models(GMMs). flying a simulated airplane using decision trees and learning obstacle avoidance and navigation behaviors using Bayesian network and k-Nearest Neighbors(kNN) classifiers. When states are mapped to motion primitives, the primitives typically are then composed or sequenced together. Similar approaches have been used for the classification of highlevel behaviors. The behaviors themselves are generally developed (by hand or learned) prior to task learning.
- Regression approaches map demonstration states to continuous action spaces. Similar to classification, the input to the regressor are robot states, and the continuous output are robot actions.Since the continuous-valued output often results from combining multiple demonstration set actions, typically regression approaches apply to low-level motions and not high-level behaviors. A key distinction between methods is whether the mapping function approximation occurs at run time, or prior to run time.

In (Ravichandar et al., 2020) the authors appraised machine learning methods to have a significant impact on the type of skills that can be learned through LfD, therefore many of the challenges in LfD follow directly from challenges faced such techniques.Such challenges include the curse of dimensionality, learning from very large or very sparse datasets, incremental learning, and learning from noisy data inherits challenges from control theory such as predictability of the response of the system under external disturbances, ensuring stability LfD is not only sensitive to who teaches the robot, but it is also still quite dependent on the platform (robot + interface) used.)

III. Stability and Convergence

In (Ravichandar et al., 2020) manipulators which are expected to provide assistance and healthcare, are also usually taught by kinesthetic demonstrations similar to the manufacturing applications. In addition, assistive robots are expected to operate in closer-proximity with humans compared to manufacturing cases. This gives rise to the need of stability and convergence guarantees of the learned policy.Further Highlighting that for instance, numerous dynamical systems-based trajectory learning methods provide strong convergence guarantees. Thus,(Ravichandar et al., 2020) emphasises rises the need for creating a learning method which does not require much data., and it can converge with a small amount of demonstrations.In (Ravichandar et al., 2020) states an approach to evaluate the convergence by a data-error plot where the number of demonstrations are illustrated w.r.t the error of the objective function. Further defining the convergence point as the amount of data that do not cause significant changes to the objective function which can be determined by the elbow method. It is very challenging to make comparison across multiple methods taking into consideration with the multiple evaluation criteria.(Ravichandar et al., 2020). In (Ravichandar et al., 2020) evaluating the performance of a LfD method is a challenging task due to the multiple factors that have to be taken into consideration.The learning method should be data and computationally efficient the outcome of the method has to be smooth without sharp changes in order to minimize the risk of damage.The method should be able to generalize efficiently and solve the desired task with high repeatability.(Ravichandar et al., 2020) states that in-order to Benchmark LfD methods "the design of a standard is needed which should include evaluation criteria, metrics, and tasks". .

IV. Dynamical System Motion Primitives

Dynamical System Motion Primitives (DMPs) are a mathematical framework for representing and generating complex motor actions. They are formalized as stable nonlinear attractor systems, which are highly flexible in creating complex rhythmic and discrete behaviors that can be adapted to changing environments. DMPs are composed of a set of basis functions, which are used to approximate the desired trajectory. The basis functions are typically Gaussian or radial basis functions, and the weights of these functions are learned from a demonstration or through reinforcement learning.

A. ***DMPs***

The motion primitives are mostly used for teaching robot motion via demonstrations. They are easy to program complex and constrained motion.Movement Primitives try to maintain the demonstration's profile regardless of the starting/goal state. Movements can be reproduced with various speeds

$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y})$

where y is our system state, g is the goal, and α and β are gain terms. It's a PD control signal, this is going to do is draw the system to the target.To the above a force term is introduced which will give let the user to a modify this trajectory.

$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f$

The crux of the DMP framework is an additional nonlinear system used to define the forcing function f over time, giving the problem a well defined structure that can be solved in a straight-forward way and easily generalizes. The introduced system is called the canonical dynamical system, is denoted x, and has very simple dynamics:

$\dot{x} = -\alpha_x x$ The forcing function f is defined as a function of the canonical system:

$f(x, g) = \frac{\Sigma_{i=1}^{N} \psi_i w_i}{\Sigma_{i=1}^{N} \psi_i} x(g - y_0)$ where y_0 is the initial position of the system,

$\psi_i = \text{exp}\left(-h_i\left(x - c_i\right)^2\right)$

and w_i is a weighting for a given basis function ψ_i. You may recognize that the ψ_i equation above defines a Gaussian centered at c_i, where h_i is the variance. So our forcing function is a set of Gaussians that are 'activated' as the canonical system x converges to its target. Their weighted summation is normalized, and then multiplied by the$x(g - y_0)$ term, which is both a 'diminishing' and spatial scaling term.

B. *SEDs*

As stated by the authors of (Khansari-Zadeh and Billard, 2011) Stable estimator of dynamic systems (SEDS) learns the parameters of the dynamical system to ensure that all motions follow closely the demonstrations while ultimately reaching and stopping at the target. More precisely, SEDS is a constrained optimization algorithm that formulates any arbitrary motion as a Mixture of Gaussian Functions. The objective function of SEDS could be mean square error or likelihood. The constraints in SEDS guarantees the global asymptotic stability of a non-linear time-independent DS. This method is particularly useful for learning discrete robot motions from a set of demonstrations. The SEDS approach models a motion as a nonlinear autonomous (i.e., time-invariant) dynamical system and defines sufficient conditions to ensure global asymptotic stability at the target.

Consider a state variable $\boldsymbol{\xi} \varepsilon Rd$ that can be used to unambiguously define a discrete motion of a robotic system (e.g. $\boldsymbol{\xi}$ could be a robot's joint angles, the position of an arm's end-effector in the Cartesian space, etc). Let the set of N given demonstrations $\boldsymbol{\xi} t, n, \boldsymbol{\xi} t, nTn, Nt = 0, n = 1$ be instances of a global motion model governed by a first order autonomous Ordinary Differential Equation (ODE):

$\boldsymbol{\xi} = f(\boldsymbol{\xi}) + \varepsilon$

V. Result

In the case of DMPs the radial base function for the model was manually changed and optimal value was selected over the trials. Similarly , in the case of SEDs the number of Gaussian

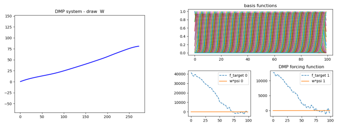

Fig. 1. Trajectory derived from DMP-Line

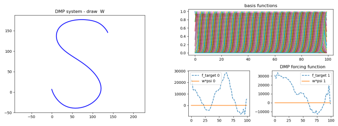

Fig. 2. Trajectory derived from DMP-Sshape

function over the trial was observed and an ideal number was chosen

For generating the DMPs trajectories the number of radial base function was set at 200. It was nortes that any base function below 70 - 150 was not given good results for the provided shapes.

The number of Gaussian introduced to the formulation was the vital factor in achieving the given figure from the SEDs methods.For the figures Line and Cshape the number of Gaussian functions was set to 1 though they also gave good results for higher gaussian values but to refrain the model from capturing noises lower value for the Gausssian for these shapes were selected(Ravichandar et al., 2020). The choice of Gaussian for higher values was chosen by the fact that shape were complex thus, more number of gaussian shall provide better result.

VI.

According to (Ravichandar et al., 2020) the choice of LfD over other robot learning methods is compelling when ideal behavior can neither be easily scripted, as done in traditional robot programming, nor be easily defined as an optimization problem, but can be demonstrated.Also, writing that state-action representations are usually modelled as a Gaussian mixture model (GMM) the number of the Gaussian components affects the complexity of the estimated functions similarly to the aforementioned cases. In (Ravichandar et al., 2020) stating hyper-parameters, such as the type of function, have to be set alongside the number of desired features. Highly nonlinear motions require more RBFs to be modelled.

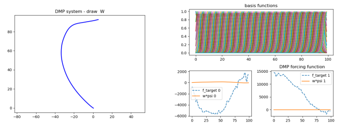

Fig. 3. Trajectory derived from DMP-Cshape

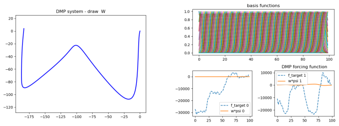

Fig. 4. Trajectory derived from DMP-WShape

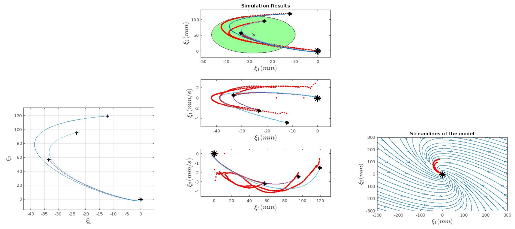

Fig. 5. Trajectory derived from Seds-CShape

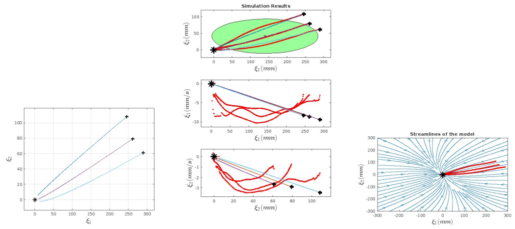

Fig. 6. Trajectory derived from Seds-Line

References

[1] Ravichandar, H., Polydoros, A.S., Chernova, S., Billard, A., 2020. Recent Advances in Robot Learning from Demonstration. Annu. Rev. Control Robot. Auton. Syst. 3, 297–330. https://doi.org/10.1146/annurev-control-100819-063206

[2] Argall, B.D., Chernova, S., Veloso, M., Browning, B., 2009. A survey of robot learning from demonstration. Robotics and Autonomous Systems 57, 469–483. https://doi.org/10.1016/j.robot.2008.10.024

[3] Sonia Chernova and Andrea L Thomaz. Robot learning from human teachers, volume 8. Morgan and Claypool Publishers, 2014

[4] Khansari-Zadeh, S.M., Billard, A., 2011. Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models. IEEE Trans. Robot. 27, 943–957. https://doi.org/10.1109/TRO.2011.2159412

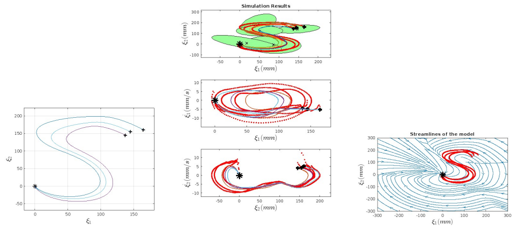

Fig. 7. Trajectory derived from Seds-Sshape

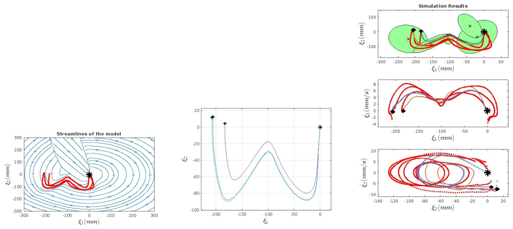

Fig. 8. Trajectory derived from Seds-WShape

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.