Learning from Demonstration

UPDATED BY: Dhawal Gupta at November 19, 2019

1 Introduction

Learning from Demonstration is a particular approach to policy learning, where the policy is learned from examples or demonstration provided by the instructor contrary to RL , in which the agent explores. This policy is only defined on those states and actions that it encounters during demonstration. The Lfd is broken into 2 phases: **1.** Gathering Examples
2. Deriving a policy from the examples
Using conventional RL, requires the robot to actually visit different states, which is often restricting for robots trying to learn, LfD on the other hand gives there benefits.
1. Doesnt require expert knowledge of domain dynamics
2. Open LfD to non robotics experts making them more commonplace
3. Demonstration acts as an intuitive medium for communication.
4. Practical benefit of focusing the dataset to state-space actually encountered during the task.
Problem Formulation : Demonstrator executed state and actions are recorded, a demonstration is presented as $d_j \in D$ for say k_j observations. $d_j \in \{(z_j^i, a_j^i)\}, z_j^i \in Z, a_j^i \in A$, for $i = 0, \ldots, k_j$

2 Design Choices

1. Demonstration Approach : 2 choices are to be made, a.) choice of *demonstrator*, b.) choice of *demonstration techniques*. a.) Choice of demonstrator can also be broken down to a.a) who control the demonstration (scripted agent, human teleoperator), a.b) who executes the demonstration (robot body). b.) demonstration techniques deals with the techniques to collect, is it an interactive procedure or a batch procedure.

2. Problem space continuity : Continuity can be decided by various factors like desired learning behaviour, set if available actions etc. The actions can be broked down to 3 control level though **a.** Low level motion control, **b.** Basic high level (action primitives), **c.** Complex behaviour actions.

3. Policy Derivation : There are multiple ways to actually go about deriving the policy and it depends on alot on the action continuity. There are 3 main policy derivation techniques i.e. a.) *Mapping Technique* , where we try to learn a function mapping from a state to action i.e. $f() : Z \to A$. b.) *System Model* : Data is used to learn the world dynamics i.e. $T(s'|s, a)$ and reward function $R(s)$ to be possibly used by an RL algo to learn the policy. c.) *Plans* : This corresponds to more to planning, where it tries to figure out the pre and post conditions of action and then tries to come up with a sequence of actions that reaches to the goal state.

4. Dataset limitation : The dataset collected is often limited by the teachers policy and capability to do the task, which can be suboptimal when compared to the ability of the learning algorithm. Hence our Learning algorithm should be able to learn beyond the demonstrators capacity.

3 Gathering Dataset

For a dataset to be successful the learner should be able to use the dataset. In the most best case, the states and actions of the teacher match the state and actions of the student. The challenges which arise between translation of the states in demonstrations to the states usable by the agent, is called as the *Correspondence Issues*. Correspondence deals with the identification of mapping between the teacher and the learner, they define it for 2 mappings 1. *record* and 2. *embodiment* mapping. Record (teacher execution → recorded execution) mapping refers to whether the exact states and actions observed by the teacher are recorded, Embodiment (recorded execution → learner) , when states and actions recorded in the dataset exactly those that the learner would observe. The papers splits the LfD data acquisition approaches to 2 categories based on embodiment mapping 1. Demonstration : No embodiment mapping i.e. $g_E(z, a) = I(z, a)$, i.e. demonstration is done on the robotic platform itself. 2. Imitation : There exists a mapping as demonstration is performed on a platform different from the robot platform.

3.1 Demonstration

Is further broken down into two cases i. e. Tele operation and Mimic.
1. *Teleoperation*: Deals with the case when the teacher demonstrates directly on the robotic platform via say teleoperation e.g. joystick , i.e. the robot should be manageable.
2. *Shadowing* :In this case the robots mimics the teachers motion while recording data from its own sensors. In contrast to teleoperation shadowing requires an extra component enabling the robot to track and actively shadow.

3.2 Imitation

: Within this setting the approaches are further divided as follows : 1. Sensors on teacher and 2. External Observation .
1. *Sensors on Teacher* : Technique in which sensors located on the teachers body are used to record the teacher execution, i.e. the records mappings are direct $g_R(z, a) = I(z, a)$.
2. *External Observation* : LfD implementation which acquire the teacher states indirectly, like in the sense of using visual cameras to record the teacher demonstration, one e.g. is first extract the teachers state/actions from demonstration, and then extract the learner state/actions from the record data.

4 Deriving the Policy

As described above there are 3 approaches to policy learning and we want to adopt the the one which involves minimal parameter tuning and fast learning times.

4.1 Mapping Approach

Mapping approaches correspond to the supervised learning setting where given a state we need to learn the corresponding action that need to be taken in a state, this can be generalized to some extent with the use of function approximator, where similar states can be mapped to having similar actions. The type of model depends on the type of state and action space i.e. continuous action space → regression models and discrete action space $classification$.

4.2 System Models

This approach uses a state transition model $T(s'|s, a)$ and from that derives a policy $\pi : Z \to A$. Reinforcement Learning is often used to solve and derive policies in this setting. There are 2 classes of approaches under this, 1. Hand Engineering Reward Function 2. Learned Reward

Function.
Hand Engineering Reward : In this settings reward functions (often sparse) are provided by the user.
Learned Reward Function : This class of methods tries to extract the reward function from the demonstration data itself and is often called as Inverse Reinforcement Learning.

4.3 Planning

In this approach we represent policy as a sequence of actions that lead from the initial state to the final goal state. These techniques not only rely on state action , but also require extra information like annotation and intentions from the teacher.

5 Limitation on Demonstration Dataset

The paper identifies 2 distinct causes for poor learner performance within LfD. 1. Dataset Sparsity and 2. Poor Quality dataset examples because of teachers inability to perform task optimally.

5.1 Undemonstrated States

This can be solved with 2 approaches
1. Generalizing using existing demonstrations : which is often done with the generalization capacity of function approximators like kNN etc.
2. Acquisition of new demonstrations: Another approach is to arrange the teacher to again provide demonstrations for unvisited states.

5.2 Poor Quality Data

Again this can be broken into two sub-problems as listed below.
1. Suboptimal and Ambiguous Demonstrations : Removing the suboptimal actions or actions that do not contribute to the optimal behaviour of the agent in the task is one way but may require domain knowledge to do the same.
2. Learning from experience : Something similar to the RL framework where executing of a task or policy is given a feedback by someone which guides the agent towards optimal behaviour.

6 Future Directions

- Learned state features/ representations
- Temporal Data
- Execution Failures
- New techniques for learning from experience
- Multi Robot Demonstration Learning
- Evaluation Metrics