Recent Advances in Imitation Learning from Observation

Faraz Torabi1, Garrett Warnell2 and Peter Stone1
1The University of Texas at Austin
2Army Research Laboratory
{faraztrb, pstone}@cs.utexas.edu, garrett.a.warnell.civ@mail.mil

Abstract

Imitation learning is the process by which one agent tries to learn how to perform a certain task using information generated by another, often more-expert agent performing that same task. Conventionally, the imitator has access to both state and action information generated by an expert performing the task (e.g., the expert may provide a kinesthetic demonstration of object placement using a robotic arm). However, requiring the action information prevents imitation learning from a large number of existing valuable learning resources such as online videos of humans performing tasks. To overcome this issue, the specific problem of imitation learning from observation (IfO) has recently garnered a great deal of attention, in which the imitator only has access to the state information (e.g., video frames) generated by the expert. In this paper, we provide a literature review of methods developed for IfO, and then point out some open research problems and potential future work.

1 Introduction

Imitation learning [Schaal, 1997; Argall et al., 2009; Osa et al., 2018] is a problem in machine learning that autonomous agents face when attempting to learn tasks from another, more-expert agent. The expert provides demonstrations of task execution, from which the imitator attempts to mimic the expert’s behavior. Conventionally, methods developed in this framework require the demonstration information to include not only the expert’s states (e.g., robot joint angles), but also its actions (e.g., robot torque commands). For instance, a human expert might provide a demonstration of an object-manipulation task to a robot by manually moving the robot’s arms in the correct way, during which the robot can record both its joint angles and also the joint torques induced by the human demonstrator. Unfortunately, requiring demonstration action information prevents imitating agents from using a large number of existing valuable demonstration resources such as online videos of humans performing a wide variety of tasks. These resources provide state information (i.e., video frames) only—the actions executed by the demonstrator are not available.

In order to take advantage of these valuable resources, the more-specific problem of imitation learning from observation (IfO) must be considered. The IfO problem arises when an autonomous agent attempts to learn how to perform tasks by observing state-only demonstrations generated by an expert. Compared to the typical imitation learning paradigm described above, IfO is a more natural way to consider learning from an expert, and exhibits more similarity with the way many biological agents appear to approach imitation. For example, humans often learn how to do new tasks by observing other humans performing those tasks without ever having explicit knowledge of the exact low-level actions (e.g., muscle commands) that the demonstrator used.

Considering the problem of imitation learning using state-only demonstrations is not new [Ispeert et al., 2002; Bentivegna et al., 2002]. However, with recent advances in deep learning and visual recognition, researchers now have much better tools than before with which to approach the problem, especially with respect to using raw visual observations. These advances have resulted in a litany of new imitation from observation techniques in the literature, which can be categorized in several fundamentally-different ways. In this paper, we offer an organization of recent IfO research and then consider open research problems and potential future work.

2 Background

In this section, we first describe Markov decision processes (MDPs), which constitute the foundation of all the algorithms presented in this paper. We then provide background on conventional imitation learning, including the problem setting and a number of algorithms developed for that problem.

2.1 Markov Decision Processes

We consider artificial learning agents operating in the framework of Markov decision processes (MDPs). An MDP can be described as a 6-tuple $\mathcal{M} = \{S, \mathcal{A}, P, r, \gamma\}$, where $S$ and $\mathcal{A}$ are state and action spaces, $P(s_{t+1}|s_t, a_t)$ is a function which represents the probability of an agent transitioning from state $s_t$ at time $t$ to $s_{t+1}$ at time $t + 1$ by taking action $a_t$, $r: S \times \mathcal{A} \rightarrow \mathcal{R}$ is a function that represents the reward feedback that the agent receives after taking a specific action at a given state, and $\gamma$ is a discount factor. We denote by $o \in \mathcal{O}$ visual observations, i.e., an image at time $t$ is
denoted by $o_t$. Typically, these visual observations can only provide partial state information. $s$, on the other hand, constitutes the full proprioceptive state of the agent, and therefore is considered to provide complete state information. In the context of the notation established above, the goal of reinforcement learning (RL) [Sutton and Barto, 1998] is to learn policies $\pi : S \times A \rightarrow [0, 1]$—used to select actions at each state—that exhibit some notion of optimality with respect to the reward function.

### 2.2 Imitation Learning

In imitation learning (IL), agents do not receive task reward feedback $r$. Instead, they have access to expert demonstrations of the task and, from these demonstrations, attempt to learn policies $\pi$ that produce behaviors similar to that present in the demonstration. Conventionally, these demonstrations are composed of the state and action sequences experienced by the demonstrator, i.e., expert demonstration trajectories are of the form $\tau_e = \{s_{t}, a_{t}\}$. Broadly speaking, research in imitation learning area can be split into two main categories: (1) behavioral cloning (BC), and (2) inverse reinforcement learning (IRL).

#### Behavioral Cloning

Behavioral cloning [Bain and Sommut, 1999; Ross et al., 2011; Daftry et al., 2016] is a class of imitation learning algorithms where, given $\tau_e$, supervised learning is used to learn an imitating policy. BC has been used in a variety of the applications. For instance, it has recently been used in the context of autonomous driving [Bojarski et al., 2016] and in the context of autonomous control of aerial vehicles [Giusti et al., 2016]. BC is powerful in the sense that it requires only demonstration data to directly learn an imitation policy and does not require any further interaction between the agent and the environment. However, BC approaches can be rather brittle due to the well-known covariate shift problem [Ross and Bagnell, 2010].

#### Inverse Reinforcement Learning

The other major category of IL approaches is composed of techniques based on inverse reinforcement learning [Russell, 1998; Ng et al., 2000]. Instead of directly learning a mapping from states to actions using the demonstration data, IRL-based techniques iteratively alternate between using the demonstration to infer a hidden reward function and using RL with the inferred reward function to learn an imitating policy. IRL-based techniques have been used for a variety of tasks such as maneuvering a helicopter [Abbeel and Ng, 2004] and object manipulation [Finn et al., 2016]. Using RL to optimize the policy given the inferred reward function requires the agent to interact with its environment, which can be costly from a time and safety perspective. Moreover, the IRL step typically requires the agent to solve an MDP in the inner loop of iterative reward optimization [Abbeel and Ng, 2004; Ziebart et al., 2008], which can be extremely costly from a computational perspective. However, recently, a number of methods have been developed which do not make this requirement [Finn et al., 2016; Ho and Ermon, 2016; Fu et al., 2018]. One of these approaches is called generative adversarial imitation from observation (GAIL) [Ho and Ermon, 2016], which uses an architecture similar to generative adversarial networks (GANs) [Goodfellow et al., 2014], and the associated algorithm can be thought of as trying to induce an imitator state-action occupancy measure that is similar to that of the demonstrator.

### 3 Imitation Learning from Observation

We now turn to the problem that is the focus of this survey, i.e., that of imitation learning from observation (IFO), in which the agent has access to state-only demonstrations (visual observations) of an expert performing a task, i.e., $\tau_e = \{o_t\}$. As in IL, the goal of the IFO problem is to learn an imitation policy $\pi$ that results in the imitator exhibiting similar behavior to the expert. Broadly speaking, there are two major components of the IFO problem: (1) perception, and (2) control.

#### 3.1 Perception

Because IFO depends on observations of a more expert agent, processing these observations perceptually is extremely important. In the existing IFO literature, multiple approaches have been used for this part of the problem. One approach to the perception problem is to record the expert’s movements using sensors placed directly on the expert agent [Ijspeert et al., 2001]. Using this style of perception, previous work has studied techniques that can allow humanoid or anthropomorphic robots to mimic human motions, e.g., arm-reaching movements [Ijspeert et al., 2002; Bentivegna et al., 2002], biped locomotion [Nakanishi et al., 2004], and human gestures [Calinon and Billard, 2007]. A more recent approach is that of motion capture [Field et al., 2009], which typically uses visual markers on the demonstrator to infer movement. IFO techniques built upon this approach have been used for a variety of tasks, including locomotion, acrobatics, and martial arts [Peng et al., 2018a; Merel et al., 2017; Setapen et al., 2010]. The methods discussed above often require costly instrumentation and pre-processing [Holden et al., 2016]. Moreover, one of the goals of IFO is to enable task imitation from available, passive resources such as YouTube videos, for which these methods are not helpful.

Recently, however, convolutional neural networks and advances in visual recognition have provided promising tools to work towards visual imitation where the expert demonstration consists of raw video information (e.g., pixel color values). Even with such tools, the imitating agent is still faced with a number of challenges: (1) embodiment mismatch, and (2) viewpoint difference.

#### Embodiment Mismatch

One challenge that might arise is if the demonstrating agent has a different embodiment from that of the imitator. For example, the video could be of a human performing a task, but the goal may be to train a robot to do the same. Since humans and robots do not look exactly alike (and may look quite different), the challenge is in how to interpret the visual information such that IFO can be successful. One IFO method developed to address this problem learns a correspondence between the embodiments using autoencoders in a supervised
fashion [Gupta et al., 2018]. The autoencoder is trained in such a way that the encoded representations are invariant with respect to the embodiment features. Another method learns the correspondence in an unsupervised fashion with a small amount of human supervision [Sermanet et al., 2018].

**Viewpoint Difference**

Another perceptual challenge that might arise in IFO applications comes when demonstrations are not recorded in a controlled environment. For instance, video background may be cluttered, or there may be mismatch in the point of view present in the demonstration video and that with which the agent sees itself. One IFO approach that attempts to address this issue learns a context translation model to translate an observation by predicting it in the target context [Liu et al., 2018]. The translation is learned using data that consists of images of the target context and the source context, and the task is to translate the frame from the source context to the that of the target. Another approach uses a classifier to distinguish between the data that comes from different viewpoints and attempts to maximize the domain confusion in an adversarial setting during the training [Stadie et al., 2017]. Consequently, the extracted features can be invariant with respect to the viewpoint.

**3.2 Control**

Another main component of IFO is control, i.e., the approach used to learn the imitation policy, typically under the assumption that the agent has access to clean state demonstration data \( \{s_t\} \). Since the action labels are not available, this is a very challenging problem, and many approaches have been discussed in the literature. We organize IFO control algorithms in the literature into two general groups: (1) model-based algorithms, and (2) model-free algorithms. In the following, we discuss the features of each group and present relevant example algorithms from the literature.

**Model-based**

Model-based approaches to IFO are characterized by the fact that they learn some type of dynamics model during the imitation process. The learned models themselves can be either (1) inverse dynamics models, or (2) forward dynamics model.

**Inverse Dynamics Models**

An inverse dynamics model is a mapping from state-transitions \( \{(s_t, s_{t+1})\} \) to actions \( \{a_t\} \) [Hanna and Stone, 2017]. One algorithm that learns and uses this kind of model for IFO is that of Nair et al. [2017]. Given a single video demonstration, the goal of the proposed algorithm is to allow the imitator to reproduce the observed behavior directly. To do so, the algorithm first allows the agent to interact with the environment using an exploratory policy to collect data \( \{(s_t, a_t, s_{t+1})\} \). Then, the collected data is used to learn a pixel-level inverse dynamics model which is a mapping from observation transition, \( \{(a_t, a_{t+1})\} \), to actions, \( \{a_t\} \). Finally, the algorithm computes the actions for the imitator to take by applying the inverse dynamics model to the video demonstration. Another algorithm of this type, reinforced inverse dynamics modeling [Pavse et al., 2019], after learning the inverse dynamics model, uses a sparse reward function to further optimize the model. Then it executes the actions in the environment. It is shown that in most of the experiments the resulting behavior outperforms the expert. Critically, these methods make the assumption that each observation transition is reachable through the application of a single action. Pathak et al. [2018] attempt to remove this assumption by allowing the agent to execute multiple actions until it gets close enough to the next demonstrated frame. Then this process is repeated for the next frame, and so on. All of the algorithms mentioned above attempt to exactly reproduce single demonstrations. The authors [Torabi et al., 2018], on the other hand, have proposed an algorithm, behavioral cloning from observation (BCO), that is instead concerned with learning generalized imitation policies using multiple demonstrations. The approach also learns an inverse dynamics model using an exploratory policy, and then uses that model to infer the actions from the demonstrations. Then, however, since the states and actions of the demonstrator are available, a regular imitation learning algorithm (behavioral cloning) is used to learn the task. In another work, Guo et al. [2019] proposed a hybrid algorithm that assumes that the agent also has access to both visual demonstrations and reward information as in the RL problem. A method similar to BCO is formulated for imitating the demonstrations, and a gradient-based RL approach is used to take advantage of the additional reward signal. The final imitation policy is learned by minimizing a linear combination of the behavioral cloning loss and the RL loss.

**Forward Dynamics Models**

A forward dynamics model is a mapping from state-action pairs, \( \{(s_t, a_t)\} \), to the next states, \( \{s_{t+1}\} \). One IFO approach that learns and uses this type of dynamics model is called imitating latent policies from observation (ILPO) [Edwards et al., 2019]. ILPO creates an initial hypothesis for the imitation policy by learning a latent policy \( \pi(z|s_t) \) that estimates the probability of latent (unreal) action \( z \) given the current state \( s_t \). Since actual actions are not needed, this process can be done offline without any interaction with the environment. In order to learn the latent policy, they use a latent forward dynamics model which predicts \( s_{t+1} \) and a prior over \( z \) given \( s_t \). Then they use a limited number of environment interactions to learn an action-remapping network that associates the latent actions with their corresponding correct actions. Since most of the process happens offline, the algorithm is efficient with regards to the number of interactions needed.

**Model-free**

The other broad category of IFO control approaches is that of model-free algorithms. Model-free techniques attempt to learn the imitation policy without any sort of model-learning step. Within this category, there are two fundamentally-different types of algorithms: (1) adversarial methods, and (2) reward-engineering methods.

**Adversarial Methods**

Adversarial approaches to IFO are inspired by the generative adversarial imitation learning (GAIL) algorithm described in Section 2.2. Motivated by this work, Merel et al. [2017] proposed an IFO algorithm that assumes access to proprioceptive state-only demonstrations \( \{s_t\} \) and uses a GAN-like architecture in which the
imitation policy is interpreted as the generator. The imitation policy is executed in the environment to collect data, \( \{(s_i^t, o_i^t)\} \), and single states are fed into the discriminator, which is trained to differentiate between the data that comes from the imitator and data that comes from the demonstrator. The output value of the discriminator is then used as a reward to update the imitation policy using RL. Another algorithm, called OptionGAN [Henderson et al., 2018], uses the same algorithm combined with option learning to enable the agent to decompose policies for cases in which the demonstrations contain trajectories result from a diverse set of underlying reward functions rather than a single one. In both of the algorithms discussed above, the underlying goal is to achieve imitation policies that generate a state distribution similar to the expert. However, two agents having similar state distributions does not necessarily mean that they will exhibit similar behaviors. For instance, in a ring-like environment, two agents that move with the same speed but different directions (i.e., one clockwise and one counterclockwise) would result in each exhibiting the same state distribution even though their behaviors are opposite one another. To resolve this issue, the authors [Torabi et al., 2019c; Torabi et al., 2019b] proposed an algorithm similar to those above, but with the difference that the discriminator considers state transitions, \( \{(s_t, s_{t+1})\} \), as the input instead of single states. This paper also tested the proposed algorithm on the cases that the imitator has only access to visual demonstrations \( \{o_t\} \), and showed that using multiple video frames instead of single frames resulted in good imitation policies for the demonstrated tasks. In this paper, the authors consider policies to be a mapping from observations \( \{o_t\} \) to actions \( \{a_t\} \). In follow up work [Torabi et al., 2019d], motivated by the fact that agents often have access to their own internal states (i.e., proprioception), proposed a modified version of this algorithm for the case of visual imitation that leverages this information in the policy learning process by considering a multi-layer perceptron (instead of convolutional neural networks) as the policy which maps internal states \( s \) to actions \( a \). Then it uses the observation \( o \) as the input of the discriminator. By changing the architecture of the policy and leveraging the proprioceptive features, the authors showed that the performance improves significantly and the algorithm is much more sample efficient. In another follow up work [Torabi et al., 2019a], the authors modified the algorithm to make it more sample efficient in order to be able to execute it directly on physical robots. To do so, the algorithm was adapted in a way that linear quadratic regulators (LQR) [Tassa et al., 2012] could be used for the policy training step. The algorithm is tested on a robotic arm which has resulted in reasonable performance. Zolna et al. [2018] has built on this work, and proposed an approach to adapt the algorithm to cases in which the imitator and the expert have different action spaces. Instead of using consecutive states as the input of the discriminator, they use pairs of states with random time gaps, and show that this change helps improve imitation performance. Another adversarial IfO approach developed by Stadie et al. [2017] considers cases in which the imitator and demonstrator have different viewpoints. To overcome this challenge, a new classifier is introduced that uses the output of early layers in the discriminator as input, and attempts to distinguish between the data coming from different viewpoints. Then they train early layers of the discriminator and the classifier in such a way as to maximize the viewpoint confusion. The intuition is to ensure that the early layers of the discriminator are invariant to viewpoint. Finally, Sun et al. [2019] have also developed an adversarial IfO approach in which, from a given start state, a policy for each time-step of the horizon is learned by solving a minimax game. The minimax game learns a policy that matches the state distribution of the next state given the policies of the previous time steps.

**Reward Engineering** Another class of model-free approaches developed for IfO control are those that utilize reward engineering. Here, reward engineering means that, based on the expert demonstrations, a manually-designed reward function is used to find imitation policies via RL. Importantly, the designed reward functions are not necessarily the ones that the demonstrator used to produce the demonstrations—rather, they are simply estimates inferred from the demonstration data. One such method, developed by Kimura et al. [2018], first trains a predictor that predicts the demonstrator’s next state given the current state. The manually-designed reward function is then defined as the Euclidean distance of the actual next state and the one that the approximator returns. An imitation policy is learned via RL using the designed reward function. Another reward-engineering approach is that of time-contrastive networks (TCN) [Sermanet et al., 2018]. TCN considers settings in which demonstrations are generated by human experts performing tasks and the agent is a robot with arms. A triplet loss is used to train a neural network that is used to generate a task-specific state encoding at each time step. This loss function brings states that occur in a small time-window closer together in the embedding space and pushes others farther away.
apart. The engineered reward function is then defined as the Euclidean distance between the embedded demonstration and the embedded agent’s state at each time step, and an imitation policy is learned using RL techniques. Dwibedi et al. [2018] claims that, since TCN uses single frames to learn the embedding function, it is difficult for TCN to encode motion cues or the velocities of objects. Therefore, they extend TCN to the multi-frame setting by learning an embedding function that uses multiple frames as the input, and they show that it results in better imitation. Another approach of this type is developed by Goo and Niekum [2019] in which the algorithm uses a formulation similar to shuffle-and-learn [Misra et al., 2016] to train a classifier that learns the order of frames in the demonstration. The manually-specified reward function is then defined as the progress toward the task goal based on the learned classifier. Aytar et al. [2018] also take a similar approach, learning an embedding function for the video frames based on the demonstration. They use the closeness between the imitator’s embedded states and some checkpoint embedded features as the reward function. In another work, Gupta et al. [2018] consider settings in which the demonstrator and the imitator have different state spaces. First, they train an autoencoder that maps states into an invariant feature space where corresponding states have the same features. Then, they define the reward as the Euclidean distance of the expert and imitator state features in the invariant space at each time step. Finally, they learn the imitation policy using this reward function with an RL algorithm. Liu et al. [2018] also uses the same reward function to solve the task however in a setting where the expert demonstrations and the imitator’s viewpoints are different.

4 Conclusion and Future Directions

In this paper, we reviewed recent advances in imitation learning from observation (IfO) and, for the first time, provided an organization of the research that is being conducted in this field. In this section, we provide some directions for future work.

4.1 Perception

Adversarial training techniques have led to several recent and exciting advances in the computer vision community. One such advance is in the area of pose estimation [Cao et al., 2017; Wang et al., 2019], which enables detection of the position and orientation of the objects in a cluttered video through keypoint detection—such keypoint information may also prove useful in IfO. While there has been a small amount of effort to incorporate these advances in IfO [Peng et al., 2018b], there is still much to investigate.

Another recent advancement in computer vision is in the area of visual domain adaptation [Wang and Deng, 2018], which is concerned with transferring learned knowledge to different visual contexts. For instance, the recent success of CycleGAN [Zhu et al., 2017] suggests that modified adversarial techniques may be applicable to IfO problems that require solutions to embodiment mismatch, though it remains to be seen if such approaches will truly lead to advances in IfO.

4.2 Application on Physical Robots

Very few of the IfO algorithms discussed have actually been successfully tested on physical robots, such as [Sermanet et al., 2018; Liu et al., 2018]. That is, most discuss results only in simulated domains. For instance, while adversarial methods currently provide state-of-the-art performance for a number of baseline experimental IfO problems, these methods exhibit high sample complexity and have therefore only been applied to relatively simple simulation tasks. Thus, an open problem in IfO is that of finding ways to adapt these techniques such that they can be used in scenarios for which high sample complexity is prohibitive, i.e., tasks in robotics.

4.3 Integration

The papers reviewed in this survey are exclusively concerned with the IfO problem, i.e., finding imitation policies from state-only demonstrations. However, to achieve the overall goal of developing fully-intelligent agents, algorithms that have been developed for other learning paradigms (e.g., RL) should be integrated with these techniques. While there is some previous work that considers a combination of imitation learning and RL [Zhu et al., 2018] or IfO and RL [Guo et al., 2019], there is still much to investigate.

Acknowledgments

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (IIS-1637736, IIS-1651089, IIS-1724157), the Office of Naval Research (N00014-18-2243), Future of Life Institute (RFP2-000), Army Research Lab, DARPA, Intel, Raytheon, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References


