Behavioral Cloning from Image Observation

Nathan Schneider Gavenski
School of Technology
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, RS, Brazil

Abstract

Behavior cloning is an imitation learning technique that takes as inputs samples of the environment state as an expert agent behaves and tries to replicate the policy used by the human expert. Recent approaches to behavior cloning use unlabelled fully-observable snapshots of the structured internal representation of such environmental state to replicate the expert behavior at a high fidelity. The reliance on fully-observable snapshots of the internal representation is a serious limitation for its application to real-world problems, where the snapshots are either noisy (e.g., imagery) or continuous. In this paper, we overcome these challenges by using unlabelled multi-modal inputs and novel neural-network architectures to effectively learn from unstructured data. Using unstructured data allows policy training from sources such as images or video (e.g., YouTube videos), expanding the range of expert sample sources for such training.

Introduction

Humans often learn new skills by observing other humans exercising their skills and imitating this behavior.

Human beings can learn about different tasks in many ways. One remarkable manner of acquiring expertise is just by mimicking another person's actions, such as babies looking at someone else's behavior and trying to learn primitive actions. The ability of learning by imitation is a general study in the area of psychology (Bandura and Walters 1977). We explore such ability by applying techniques, commonly known as *learning from demonstration* (*LfD*), which focus on giving autonomous agents the ability to learn from expert demonstrations (Schaal 1997; Argall et al. 2009).

Even though the LfD area is motivated by how humans learn from observation, it is our understanding that two significant points were overlooked and should need mentioning. First, and one of the most crucial differences from classical LfD works when compared with humans, is that people do not have explicit knowledge of the action performed by the expert. Secondly, is that humans can learn from imitation without spending much time converting the observed information into the action itself. One example would be when watching someone else play an unknown video game, although we are not able to fully understand the action performed by whose in control, we are still able to map what we observed into another known game, and comprehend the action executed. By not taking those differences into account, we can see that the LfD works are not accurately following necessary assumptions in order to imitate humans.

Another LfD challenge is the amount of labeled data need to learn from the demonstration. The major part of LfD algorithms usually is trained in a supervised way, which has the necessity labeled data. Unfortunately, it is not easy to acquire labeled videos from expert performing some specific task. Although, if we could develop an approach that does not need much prior information about expert actions, we can go from various sources of data. For example, experts performing a task from a tutorial video on Youtube.

In this work, we address the problem by learning to imitate images with no prior information. This task motivates us since humans imitators do not have the information about the action that they are trying to imitate — differing from other LfD works, where the autonomous agent trains supervised with prior information, in our work we propose a novel that performs imitation learning through a visual perspective with no prior information called Behavior Cloning from Image Observation (BCIO).

Technical Approach

Behavior Cloning from Observation

This project will be based on Torabi et al. (Torabi, Warnell, and Stone 2018) work, where we create two different networks that are responsible for learning the action and how the expert performs. The first network, accountable for mapping action without domain expertise, is called Inverse Dynamics Model (IDM). IDM's train data is a tuple that constitutes from the initial state (s_i), an action (a), and the next state caused by that action (s_{i+1}). Since we do not have any expert labeled data, all data created for the IDM is random actions in any given environment. The second network, responsible for learning how the expert acts, is called Policy Model. As the name suggests, the Policy Network is responsible for learning which action should it perform given any state. With that in mind, training both models is done by randomly generating batches of (s_i, a_i, s_{i+1}) in an environment and passing it through the IDM. Afterward the trained model

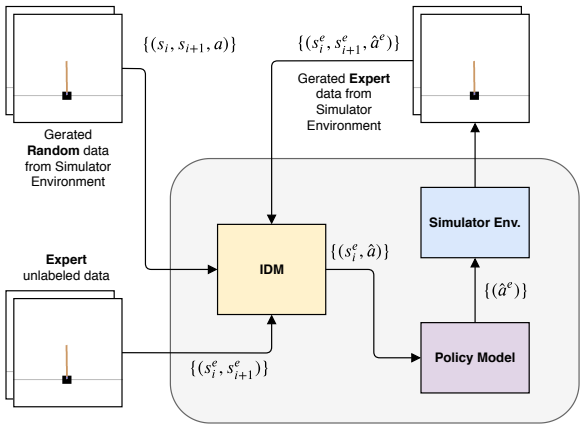

Figure 1: Pipeline of our proposed architecture.

will infer in the expert unlabeled data (s_i^e, s_{i+1}^e) and label all tuples with an action $\hat{a}$. This enable the Policy Model to train in a supervised manner, where all actions were created based upon the mapped function $\mathcal{M}_a^{s_i s_{i+1}} = P(a|s_i, s_{i+1})$. As an improvement to Torabi et al. (Torabi, Warnell, and Stone 2018) work, they propose BCO(α), where after the Policy Model end its training phase, it will interact with the environment creating a new batch of data (s_i^e, s_{i+1}^e, $\hat{a}^e$). The new batch is passed through the IDM so it can better map how the actions performed in the current domain. That allows both models to learn with each other cooperatively and enables the IDM to correct all possible mislabeled actions created in the previous episode since the new batch allows the model to get unseen labeled data. The training phase can be seen in Figure 1.

We intend to use BCO with images so we can create states that better represent the environment. Some environments do have better representation throw a list of scalars during the simulation of the agent, but others might lack information. Since IDM is a task-independent model, we assume that the visual representation might bring better features for the domain-driven task, and the model will better learn how to interact and mimic the expert.

Evaluation Method

For this work, we will not test our approach with continuous environments, since those require better features than the only image (e.g., Optical Flow). We consider categorical environments those that when an agent performs an action, and the current state of the simulation is modified, no other change will occur until another action is taken (Hussein et al. 2017). However, Torabi et al. work, and Edwards et al. (Edwards et al. 2018) do not work with categorical problems. For this project, we intend to use their methods with all environments chosen by us and evaluate them using the same performance measurements.

- Performance: Torabi et al. (Torabi, Warnell, and Stone 2018) uses the same evaluation metric as GAIL (Ho and Ermon 2016), where the reward is scaled so that the expert achieves one and a random policy achieves zero.
- Average Episodic Reward: Edwards et al. (Edwards et al. 2018) conducted 50 runs over each environment and averaged the reward for each episode.

The difference between both metrics is that Edwards et al. (Edwards et al. 2018), although having an expert, interacts with the environment and learning with each experience. The Performance measurement allows us to understand better how the agent is capable of learning what the expert does, while the Average Episodic Reward enables us to see how our model learned the domain-specific task.

Environments

- Maze: a simple 2D maze environment where an agent (blue dot) finds its way from the top left corner (blue square) to the goal at the bottom right corner (red square). The objective is to find the shortest path from the start to the goal (Chan 2017).
- Sokoban: is a type of puzzle video game, in which the player pushes crates or boxes around in a warehouse, trying to get them to storage locations (Schrader 2018).

Project Management

We already have implemented all BCIO models, and the training pipeline, in Python with Pytorch and Tensorflow. We also have already chosen two different categorical environments that enable us to test our current models and hypothesis. Bellow, we propose a project weakly goal that will allow us to complete our project in due time.

- Week 1: we need to create expert demonstrations for each environment we intend to use and create one expert dataset using an algorithmic approach (e.g., Genetic).
- Week 2 and 3: we will train our models with the Maze and Sokoban environments and see how it performs with the same maps that the expert interacts with, and afterward, we will test it with randomly generated mazes and warehouses to see how well it generalized.
- Week 4: we will try to use the same IDM from the Maze environment to learn the Sokoban movement, and how well its knowledge of task-independent movement transfers from one game to another. Domain adaptation is not the objective of this project, but we will this as bonus.
- Week 5: we will analyze the results obtained and write the final essay for this class based on the findings.

Conclusion

In this project, we aim to create a new model based upon Torabis et al. (Torabi, Warnell, and Stone 2018) work and use Imitation learning as a way to create stationary agents capable of executing tasks given expert experiences. We believe that we will solve the Maze environment and will be able to generalized, but we do not think that the Sokoban will be easy to generalized. Moreover, we believe that the domain adapation bonus will not perform well given the visual differences from each environment, but the training time will be reduced. The project, if it can achieve better rewards on those environments, can be published afterward in an Artificial Intelligence Conference.

References

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Bandura, A., and Walters, R. H. 1977. *Social Learning Theory*. Prentice-hall Englewood Cliffs, NJ, 1 edition.

Chan, M. 2017. gym-maze. https://github.com/MattChanTK/gym-maze.

Edwards, A. D.; Sahni, H.; Schroecker, Y.; and Isbell, C. L. 2018. Imitating latent policies from observation. *ArXiv* abs/1805.07914.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*, 4565–4573.

Hussein, A.; Gaber, M. M.; Elyan, E.; and Jayne, C. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)* 50(2):21.

Schaal, S. 1997. Learning from demonstration. In *Advances in neural information processing systems*, 1040–1046.

Schrader, M.-P. B. 2018. gym-sokoban. https://github.com/mpSchrader/gym-sokoban.

Torabi, F.; Warnell, G.; and Stone, P. 2018. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, 4950–4957.