An Investigation into Procedure Cloning

Thomas Walker

December 2022

1 Introduction

In this article, we explore the idea of training an artificial agent through imitation. We discuss the technique of imitation learning through procedural cloning (PC) which is proposed in [1]. The main idea behind this technique is to expose an agent to the action of an expert given an observation of the state, along with the computation made by the expert to arrive at their chosen action. We will first motivate this idea with an example, then explore how this technique extends previous techniques of imitation learning, and why procedural cloning seems to generalize the agent's understanding better than other techniques. Finally, we will illustrate some basic examples of where this technique has been empirically shown to work and use these to highlight some of its drawbacks.

2 Motivating Example

The concept of imitation learning is powerful and motivated by how humans learn. Most of our understanding of the world comes from reading between the lines and making inferences about the actions we observe in our environment.

Mathematics is a field developed on axioms, where theorems and propositions are developed through logical deductions made from first principles. The transfer of mathematical knowledge from expert to student often involves reciting statements with their proofs and observing the subsequent consequences of these arguments. This is mainly achieved through the medium of a lecture where the results and their derivations are given. At some point, the student is expected to think beyond the actions of the expert and start to develop a new theory that builds upon what they have previously learned. This a step that is challenging for a student that merely observes the facts. Therefore, a mathematics student is often supplemented with problems that utilise the theory they have learned and get them to use it in a way that highlights its properties and gives some guidance to its motivation. It's the combination of these two forms of learning that enables the student to develop an understanding to then be able to develop upon the statements. In this way, mathematics as a field can progress.

Underlying this process, the student is implicitly reading between the lines of the actions taken by their professor during the lecture to try and gauge the motivation of the actions. Then they are trialing their beliefs and refining them by tackling various problems relating to the theory. From this, the student has developed an implicit understanding of the theory and can then utilize it in an exploratory and progressive way.

This sort of understanding is often difficult for an expert to articulate in a form that fully conveys their thoughts to an audience. Furthermore, an expert may not be fully aware of the exact ideas that have led to their actions, it may just be intuition that guides them. Intuition is something that is developed over time and is difficult to encode into an agent. However, as we see in mathematics, small amounts of intuition can be transferred to the agent through implicit observation of the expert's actions.

This notion gave rise to imitation learning which sought to outline a framework by which an agent can learn from an expert. There are various forms that this idea has taken in practice and in this article, we explore the notion of procedure cloning (PC), and how this develops upon the idea of behaviour cloning (BC).

3 Behavioural Cloning

The idea behind this method is to give an agent access to the states observed by an expert and the subsequent action taken. For example, to train an agent to play chess using BC we may show them various board positions and the resulting moves made by high-level players.

Formally, if we suppose that the expert follows a policy $\pi_* : S \to A$ (where S is the state space and A is the action space) we want our agent's policy, π, to approximate π_*. To do this we sample π_* to produce a data set of tuples

$$\mathcal{D}_* = \{(s, a)\}$$

where s is the observed state and a is the action taken. We then use this to train our agent to approximate π_*. We require some metric to quantify this approximation, the choice of such a metric varies and which we choose will not discuss here.

Behavioural cloning is aptly named as the agent is trained to mimic the behaviours taken by the expert. You could argue that the agent hasn't really learned the ideas of the expert but rather has identified patterns in their actions that portray a sense of understanding. In fact, this is seen empirically as this method fails to generalize well to out-of-distribution states. To rectify this we need to provide the agent with a richer source of information, which is the motivation behind procedure cloning.

4 Procedure Cloning

Procedural cloning extends behavioural cloning from simple imitation learning to chain of thought imitation learning. That is, we now expose the agents to the intermediate computations made by the expert to arrive at their action. If we wished to train a chess-playing agent using PC then in addition to the state and actions made by the expert, we would need to include the intermediate computations made by the expert. This could include calculations about the subsequent moves of the other player, or be the strategic reasoning made by the expert to set up a closing move further down the line.

To formalize this, we break down the action of an expert into a procedure of specified granularity. A procedure is defined as a sequence of computations

$$(\Pi_0, \Pi_1, \dots \Pi_L, \Pi_{L+1})$$

where each Π_i is a function from the state space to the action space. We develop our data set by sampling the expert policy at a particular state, s, and noting down the intermediate actions taken, $x_i = \Pi_i(s)$, along with the final action $a = \Pi_{L+1}(s)$

$$\mathcal{D}_\Pi = \{(s^{(i)}, \mathbf{x}^{(i)}, a^{(i)})\}_{i=1}^n.$$

Ultimately, the agent is still trained to approximate π_* which takes only the state as input. We are not trying to replicate the intermediate steps taken by the expert in our trained policy π, but rather using them to help generalize the agent's reasoning to get better performance on out-of-distribution states. During training, we are trying to develop a policy to maximize

$$p(a, \mathbf{x}|s) = p(a|\mathbf{x}, s) \cdot \prod_{l=1}^{L} p(x_l|\mathbf{x}_{<l}, s) \cdot (x_0|s) \quad (\star)$$

which is the probability of taking expert action along with the intermediate steps provided we know the state that we are in. Often we can presume that each computation in a procedure only depends on the previous computation which reduces $(\star)$ to

$$p(a, \mathbf{x}|s) = p(a|x_L) \cdot \prod_{l=1}^{L} p(x_l|\mathbf{x}_{l-1}, s) \cdot (x_0|s).$$

Consequently, this framework requires the agent to develop a more complex narrative of the expert policy with the intention that this will aid the generalisation of the agent's knowledge, providing better performance in previously unseen states. The paper illustrates this framework in practice by providing some empirical results. The experiments conducted in the paper show the performance gain experienced because of using the PC, but also highlights its drawbacks.

5 Experiments

The paper details the performance of PC in a variety of settings, including navigating a maze, manipulating robot appendages, and playing strategy games. In each, the implementation of PC is slightly different to accommodate the differing features of each task. However, in all the tasks indicated PC provided better generalization compared to BC as well as showing better overall performance.

One of the experiments involved manipulating robotic effectors to move a group of particles into a set of bowls. As part of the training process, the agent was tasked with predicting the coordinate location of the particles along with their desired end location. The agent then learned a policy for conducting actions that would minimize the approximation metric when conditioned on the coordinates predicted and the end-effector position/orientation as seen in the expert action. When additional information regarding intermediate computations were omitted the agent quickly overfit to the data, it was as if the prediction of coordinates bared no relation to the subsequent action of the agent. Whereas a PC trained agent showed an understanding between the coordinates of the particles and the necessary actions required to move them to their desired location, as a result, it generalized well to the unseen environments provided in testing.

A point of friction for implementing PC is in generating the data set of expert procedures. The set of computations that are to be included in the procedure is determined by the user implementing PC. On the one hand, the length of procedures must be comprehensive enough to capture the underlying intuition of the expert. However, individual computations cannot be too mundane to the point where they provide no context for the agent. Furthermore, sampling the expert policy varies in difficulty depending on the scenario. The first experiment considered by the paper involves an agent navigating a maze. In this case, we can sample the expert policy by using a breadth-first-search algorithm, the computations included in the procedure are taken to be the arrays visited by the algorithm. However, when the paper considered PC to train an agent on MiniAtari games Monte-Carlo Tree Search was run to develop the expert trajectories. This was more convoluted than running BFS, as when sticky actions were considered (to increase the generalization of the agent) MCTS only produced good actions after extensive training. This induces a computational overhead to the entire PC process.

6 Conclusions

All in all, we see that PC provides a framework to establish an agent with a more generalized understanding of the actions taken by an expert. The richer source of data seems to alter the learning process from a simple pattern recognition exercise to an exercise that requires the development of an 'intuition'. I use this term 'intuition' in a broad sense as in the experiments outlined the expert policies were algorithms that I do not believe to be intuitive in the human sense of the word. Human intuition is an elusive concept that even those with it do not fully understand how they acquired it or what it is. Therefore, I believe the only way intuition can be learned is implicitly. There is the potential for frameworks similar to the one discussed above to produce an agent that shows signs of intuition. It seems PC increases the probability that an agent develops this desirable characteristic compared to BC. However, I also believe that intuition is developed and strengthened through self-play and exploration. Perhaps, a combination of chain of thought imitation and learning through trial and error may develop an agent with intuition. This circles back to the motivating example of learning mathematics. A student learns implicitly by being presented with the computations of an expert (in the form of a lecture) and strengthens those ideas through a process of trial and error in applying the ideas presented to them (in the form of problem sheets).

7 Connection to the Alignment Problem

Is intuition the required quality of a super-intelligent AI agent to ensure that it is benevolent to humans? Reinforcement learning agents are notoriously opaque. Especially for embedded agents, as then the optimization problem is dynamic making it difficult to isolate and investigate specific features of the model. However, suppose we have a rational agent that has developed an intuition, as the agent is rational, we could expect that it will follow that intuition. From humans we understand that everyone has a roughly similar intuition, so we can assume that the intuition of the AI may be aligned with our own. On the one hand, the initial actions of an agent may be like our own and so aligned with values. On the other hand, one knows that our intuition is not always correct and we often diverge from it when we notice something is not working or we desire a different outcome. So, intuition may not guarantee a super-intelligent AI will remain aligned with our values, however, its initial actions may not deviate too far from our values. Intuition may provide the key to ensuring that the existential consequences of superintelligent AI on humanity will not happen immediately, allowing us to notice any issues and counter them before it's too late. Although, it will probably not provide a guarantee that such an agent is not inherently malevolent.

References

[1] Mengjiao Yang, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. *Chain of Thought Imitation with Procedure Cloning*. 2022.