Elaborations on apprenticeship learning

Paul Christiano Follow
Nov 20, 2015 · 10 min read

Apprenticeship learning (AL) is an intuitively appealing approach to AI control. In AL, a human expert defines a goal implicitly by demonstrating how to achieve it, and an AI system imitates that behavior. This approach can potentially communicate very complex goals under minimal assumptions. And even if the system fails to implement the desired behavior, it is unlikely to pursue an adversarial goal.

AL has not yet reached the point where it is a scalable approach to AI control, but it faces a different kind of challenge. For most approaches to AI control, the problem is continuing to behave as intended while capabilities scale up. But for AL, the problem is achieving optimal performance as capabilities scales up.

This is *already* a key problem for AL. Because of this close alignment between capability and control, I think that AL is a promising area for research on AI control.

Even if apprenticeship learning is ultimately unsuitable as an approach to scalable AI control, many of the same issues arise straightforwardly for approval-directed agents and potentially for any approach that eschews broad goal-directed reasoning about the world.

How conservatively to copy?

We might consider two versions of AL:

- **Liberal**. In this formulation, the learner tries to uncover the actual goals of the expert and to pursue those goals. This requires a prior over possible preferences, a model of the constraints and cognitive limits of the expert, and aggregation across the posterior over possible preferences. This may result in the learner pursuing policies quite different from the expert.
- **Conservative**. In this formulation, the learner's goal is to do at least as well as the expert as judged by *every* preference function

from some large class of possibilities. This effectively forces the learner to closely imitate the expert .

Of course we can also consider intermediate proposals. For example, we might modify the liberal approach by considering a large class of possible probabilistic models for the expert's preferences and limitations, and guaranteeing good expected performance assuming that any model from this class is accurate. Or we might modify the conservative approach by making increasingly strong assumptions about the expert's preferences.

For concreteness I'll talk about the conservative policy. This particular choice is not important; what is important is that I am **not** willing to trust the value inference process to judge policies wildly different from those adopted by the expert. So before our AI system can perform a task, a human must be able to perform the task in substantially the same way.

Copying and distinguishing

When it matters, I will assume that we implement AL using the general framework proposed in the original paper on AL by Abbeel and Ng, as discussed in this post. In this framework, we train a classifier to distinguish human and AI performance, and simultaneously train our AI to fool the classifier into thinking it is a human.

(Independently, I would be interested in seeing applications of this technique with neural networks in place of the SVM. For example, it would be nice to see if such methods could outperform more direct approaches to imitation learning in Go—it looks to me like they could be an improvement, but trying it out could also reveal weaknesses in this general framework. I'm not aware of any work along these lines, though wouldn't be surprised if it has been done.)

Elaborations

Apprenticeship learning systems could be expanded in many directions. Some of these directions seem especially relevant to assessing and understanding the scalability of AL as an AI control solution. I am optimistic that these directions might be interesting to AI researchers, while being sufficiently distinctive that they would not otherwise receive much attention.

I have discussed all of these elaborations before and talked about their relevance to AI control. Here I want to expand on the problems themselves, and collect them all in one place. I discuss some general issues, including possible application domains, in the final section.

Bootstrapping

The first challenge for scaling AL is: how can imitating human behavior yield superhuman performance?

I think this challenge might be resolvable by a natural form of bootstrapping. Rather than imitating human behavior, the AI system imitates the behavior of a human who has access to a collection of AI assistants. These assistants can also be trained using AL-with-bootstrapping. In principle, such a process could scale well past human level.

There is no need to make this process explicitly iterative: it may be most efficient to continuously give the expert access to the current version of the AI and to do online training.

In order for this bootstrapping to be effective in realistic settings, the task will probably have to be extended. For example, in order to play chess effectively, a human might want access to assistants who can evaluate moves, identify key considerations, and so on. Using assistants with this expanded library of capabilities, the expert can then attempt to demonstrate not only what moves to make, but also how to perform each of the tasks in this expanded library.

This seems like the extension of AL most directly and clearly relevant to understanding its scalability. An impressive demonstration might also be pretty cool, giving a clear intuitive picture of how a human can robustly "control" a process significantly beyond their own abilities. Such a demonstration would probably be possible only in a very carefully chosen domain, where AI performance substantially exceeded human performance and the cost of training data is manageable.

In some domains, this kind of bootstrapping would reduce directly to standard RL algorithms. For example, by training a move evaluator using a natural mechanistic process we can emulate Q-learning, but with a gratuitous human in the loop (greatly increasing the cost of training data). This is in some sense reassuring, but a (1) compelling demonstration would need the human to add significant value beyond this mechanistic process, enough to offset the cost of human involvement, (2) if this mechanistic process is actually optimal, we would need to be able to reduce the human involvement to an acceptable level. I will discuss this issue in much more depth in a subsequent post.

Meeting halfway

In many domains, AI systems cannot imitate human behavior (even if they can achieve comparable performance according to the metrics we care about). In fact in very few domains can AI systems *perfectly* imitate human behavior. It seems worthwhile to study more explicitly how AL degrades when imitation is impossible.

For conservative AL, this necessarily requires the human to adjust their behavior to accommodate the AI's limitations. This dynamic doesn't seem to have been studied much, though I expect some adjustment has happened implicitly behind the scenes.

Recall that we are doing AL by training a classifier to distinguish human and AI behavior. One way to do this adjustment is for the human to inspect that classifier's behavior, to view this classifier as imposing an extra constraint (namely, "look like an AI"), and then to accomplish the underlying task while looking like an AI. This requires that the human be able to understand the classifier's model well enough to understand what kinds of behaviors would look like an AI. This requires some transparency of the classifier, but it's a novel kind of transparency and it's hard to know what to make of it (how to achieve it and how hard it will be) without trying it out.

Another requirement is that the human and AI are *capable* of accomplishing the task in a similar-looking way. In many cases this won't be possible, because the AI can only perform the task by using techniques that the human cannot imitate. To address this problem we would need to apply bootstrapping or some similar measure—this problem is structurally equivalent to achieving superhuman performance. In the long run this is clearly an important question, but we could start by focusing attention on domains where this is not necessary.

Explanation

When imitating a whole "trajectory," AL can in principle guarantee that the learned behavior is roughly as good as the expert's behavior —simultaneously for every notion of "good." We get this guarantee as long as our classifier can learn to distinguish trajectories with different levels of goodness.

If we want to use AL to take only part of a trajectory—for example, in order to play single moves of a board game, or to take the first step towards complex real-world plans—then it seems harder to use this technique to get comparable guarantees. The problem is that the goodness of a step has a very complex relationship to the goodness of a plan, and this relationship may be opaque to our classifier. So, for example, a classifier may not be able to distinguish "good" moves from "bad" moves, even though it can easily distinguish a winning sequence of moves from a losing sequence of moves.

Moreover, even if it is possible to make this determination, actually learning this classification or behavior seems likely to be very challenging.

We could try to address these problems by considering *justified actions*—single steps along a trajectory, together with an explanation for why the proposed step is desirable. This requires gathering more involved training data, but could potentially accelerate the learning process and increase the robustness of the learned behavior.

Of course these explanations need not be in natural language; they could be in a very task-specific format, or could essentially correspond to a broadening of the task rather than being an "explanation" in the conventional sense. (For example, we might train a model to predict human life-and-death judgments in Go at the same time as making moves, or to predict annotations that are especially relevant to the current move.)

These explanations may also involve interactions between several AL systems. For example, we could simultaneously train one AL system to select and justify moves, while we train another system to criticize proposed moves. A human can play either role while an AI plays the other, and the classifier can simultaneously make predictions about both systems. (You could apply a similar dynamic to learning to play a game, having a classifier predict which side is human rather than simply using the outcome of the game as a training signal. That would be interesting for understanding the basic dynamics of the situation, but is of less direct relevance.)

Approach

Some notes on the approach I would take in this area, if I were to work on it (which is a live possibility). These views are less well-considered than the object level views about what elaborations are worthwhile.

Picking directions. These elaborations are connected and probably not exhaustive. I expect an AI-control-focused project would be best served by approaching AL with an open mind about what developments (if any) would be most productive. The resulting project could easily end up exploring any one of these elaborations, all of them at the same time, or none of them.

Training data. All of these projects require idiosyncratic training data, and would require collecting new data (unless you get really creative). In many domains acquiring test data looks likely to be a major difficulty/expense. I expect this comes with the territory—these are fundamentally questions about the relationship between users and machines, to a much greater extent than many AI projects. The mix of data-hungry methods and human-centric questions seems like a serious obstacle.

In the long run these questions might be answered in the context of productive systems that users would be happy to use for free. But in the short term, I wouldn't be at all surprised to find myself paying thousands of dollars for training data, or spending hundreds of hours on data acquisition if done by researchers or volunteers.

A project along these lines may also find itself forced to experiment with combining AL with semi-supervised learning and other techniques to make frugal use of data. These challenges are at least somewhat relevant to AI control, but I think the tight connection is probably an artifact of trying to do this work far in advance.

Picking domains. In principle our goal is to be able to achieve state-of-the-art performance with AL in every domain. This suggests an easy domain-picking methodology: look for anything that we don't currently know how to do with AL, and try to get almost-as-good performance by applying AL.

(In many domains, like classification problems with human-provided labels, this is actually trivial—this domain can automatically be viewed as a special case of AL, and so basically any solution will have the same safety properties.)

We have some extra constraints. Most saliently, real-time domains complicate substantive AI/human collaboration (since it is hard for the expert to make use of AI assistance, or to break trajectories into steps); for now it seems easiest to just set such domains aside.

Some possible domains that I've considered:

- Playing board games.
- Playing turn-based video games or puzzles.
- Producing basic pixel art, line art, or music.

Finding appropriate domains seems likely to be a key part of actually making this project work.

Redundancy. In some domains, where clear external feedback is available, applying AL feels gratuitous. From a scalability perspective there is a clear motivation for avoiding the reward signal provided by nature—we suspect that the use of such rewards is inherently hard-to-scale. And even setting aside safety, it is clear that in the long run an increasing range of domains won't have clear reward signals. Hopefully these motivations can be enough to justify research that explicitly avoids using external feedback from the environment, even in domains where good feedback is available.

It is quite possible for AL to result in improved performance in some domains, even if useful external feedback is available. This is certainly convenient and it may be best to focus on those domains, but I wouldn't want to present the value of the approach as dependent on this happy semi-coincidence.

We could also try to focus on domains where AL is really necessary to define the task. This is the approach most existing research has taken (very understandably). Unfortunately it often involves realtime domains, which complicates collaboration between the user and machines. It might be worth dealing with the realtime issues in order to make these domains accessible.