Concrete approval-directed agents

Paul Christiano Follow
Nov 19, 2015 · 13 min read

This post lays out my current concrete "approval-directed agents" proposal and outlines what I see as the most promising broad directions for further research. (I will describe more concrete research problems in a series of upcoming posts.)

We'll take as given a powerful algorithm for episodic reinforcement learning, and assume that we are interested in building a system which acts effectively in the world in a way consistent with its user's values.

I'm going to talk a lot about the "scalability" of AI control schemes. Ideally, as our AI systems become more capable they will also robustly become more helpful and better allow us to achieve our goals; I refer to any failure of that property as "failures of scalability." In other words, scalability is the desirable property of continuing to predictably achieve the desired effect, with acceptable additional effort, as the capabilities of the underlying learning algorithms improve.

Current proposal

Summary

Each time our system needs to act we elicit an action from our learner. Most of the time we simply take this action. With small probability we instead submit a description of the action to a human operator for review. The operator assigns that action a score in the interval [-1, 1]; in the process of assigning the score, the operator may make use of as many automated assistants as needed, each of which works according to the same principle.

Our learner is trained to output actions that maximize the score assigned by the human operator, conditioned on the operator reviewing that action.

Detail

We'll work with a pair of independent learning algorithms A, B.

These algorithms play a simple online learning game: in each round they produce a binary output, and receive a score in [-1, 1]. They are trained to choose individual outputs that will receive the highest score. We may elicit an output from the learning algorithms and not assign any score; in this case the episode is not added into the labelled training data (though it may be included for semi-supervised learning). The learners effectively condition their choices on the assumption that a score will be given.

Each time our system needs to take an action, we perform the following procedure. We'll assume that actions are encoded as binary strings.

1. Ask A for a binary output $a \in \{0, 1\}^*$. With probability 99%: interpret *a* as an action, take that action, and move on to the next action.
2. With probability 1%: ask B for a binary output $b \in \{0, 1\}$. If b = 0: interpret *a* as an action and take that action, give B a payoff of 0, and move on to the next action.
3. If *b* = 1 we say that B "challenged" A's proposed action. In this case we notify the human operator and ask her to score *a* and *b*.
4. The operator may instantiate any number of additional approval-directed agents (i.e. agents who act by iterating steps 1–6). These agents are modified to be more conservative: they terminate in step 1 with 0% probability, always moving on to step 2 and providing an opportunity for challenges.
5. The operator gives A a payoff of ±1 based on whether *a* seems like a good proposal, and similarly gives B a payoff of ±1 based on whether challenging was appropriate.
6. We return to step 1 (without taking any action) and elicit a new action.

In step 4, it is critical that the operator instantiate additional approval-directed agents and use them to identify and weigh up possible faults in A's proposal. The operator may also consult other approval-directed agents prior to beginning this protocol to help improve the integrity of the setup or make adjustments (e.g. to address possible software faults that A might use to undermine the evaluation process).

State

A and B should both be given some description of the state, describing the problem they currently face and what they “know” about the world. Maintaining this state is a subtle part of the proposal—we can’t determine optimize an internal representation of state using gradient descent without compromising the format.

I will discuss this issue in more detail a future post. For now, suppose that A and B take as input a “complete description” of the state (we may have to use a larger model class, such as RNNs rather than neural networks, in order to accommodate large and structured description). One of the actions we can make available to A is to forget part of history or write down additional information that it might find useful—these actions are evaluated in the same way as other possible actions.

Concreteness

This proposal corresponds to some code we could *literally write today*. This is sufficiently unusual, amongst scalable approaches to AI control, that I want to emphasize it.

If you set out to write this code, the main problem you face is writing the learning systems A and B. This is an extremely common *kind* of problem that can be attacked with very standard methods. But it is quite a difficult instance of that problem, and existing techniques won’t handle it very well.

Making things actually work will probably require both progress in AI and improvements to the scheme.

Discussion

Goal

For some task X, say we are “capable of automating X” if we can build a system which achieves X in some context that does not inherently reduce the difficulty of the task, e.g. which will perform X whenever the user requests it, or whenever doing X so is the best way to earn money, or so on.

I’d consider the approval-directed agents approach a success if, for every task X which we are capable of automating at a cost *c*, an approval-directed agent can achieve X at a cost not much more than *c*, and **in a manner consistent with the operator’s preferences**.

For example, if we could train an AI system that could design a rocket if doing so was the best way for it to maximize paperclips, then we ought to be able to use this system to design a rocket without side-effects or distortions that the operator would be unhappy with.

Why might this protocol succeed?

Intuitively, the success of this proposal rests on a few hopes:

1. **Explicability:** Our AI systems are nearly as good as explaining *why* they want to do something as they are at choosing *what* to do. Producing these explanations may require more time than choosing what to do, but it should at least be possible. The explanations should be sufficiently rigorous that it is possible to distinguish “correct” vs. “incorrect” actions on the basis of their explanations.

2. **Efficacy:** By getting help from additional approval-directed agents, the human operator can evaluate proposals as if she were as smart as those agents. In particular, the human can evaluate the given rationale for a proposed action and determine whether the action really does what the human wants.

3. **Soundness:** By working with AI assistants (at run-time and in advance), the operator can ensure that AI systems cannot learn to undermine the evaluation procedure itself—learner A cannot learn to find actions which are ineffective but criticism-proof. For example, A can’t find actions such that merely looking at them jeopardizes the operator’s judgment or their software.

I find these assumptions intuitively meaningful, but to be precise we need to be much more careful about what kind of feedback is available. For example, there is a trivial sense in which a human+AI system is “as smart as” the AI—we need something stronger.

1. **Explicability** requires that our AI systems can learn to behave well using feedback only on the quality of their explanations.

2. **Efficacy** requires that a human+AI system can learn to evaluate explanations *with no feedback*—the operator can train her AI assistants however she wants, but ultimately all direction must come from the operator’s choices rather than being a response to an external source of feedback.

3. **Soundness** requires that the evaluation system can be made robust using partial feedback: by simply running the evaluation

> procedure the human can get feedback about what kinds of attacks are feasible and can then correct those attacks. But this feedback is only available for non-catastrophic problems; for some problems, a small number of failures is already unacceptable.

Note that if **direct feedback** is available for a task, then the explicability+efficacy requirement is essentially trivial: the operator can "evaluate" an AI's behavior by simply allowing the AI to do the task and then providing it with direct feedback on its performance.

"Explanations" are only challenging in domains where a demonstration is not itself adequate explanation—for example where a human may not be able to tell whether the AI has succeeded at a task, or where a task may extend over a very long time period, or where a demonstration may compromise soundness by giving the AI the ability to interfere with the humans' evaluations.

Technically soundness and explicability aren't distinct requirements —the reason that we need our evaluation scheme to be secure is that, if they are not, then training our AI systems to produce good explanations won't result in good behavior. Moreover, if our evaluation scheme isn't sound, it's not clear in what sense it constitutes a good evaluation. Nevertheless, it seems helpful to split this claim in two: first that we can learn to produce good explanations, and second that we can't learn to get good payoffs by other means.

A warning on soundness

In order to address concerns about soundness, I previously included a simple procedure based on "watchdogs:" I recommended that whenever the operator consulted either A or B, she first run their output by a slightly-less-intelligent filtering algorithm C. I no longer believe that this procedure robustly improves soundness, so I have eliminated it for simplicity. This may serve as a bad omen of surprises to come.

The problem with this proposal is quite simple: even though the filtering algorithm C is less intelligent than algorithms A and B, algorithms A and B may implicitly communicate with C. This makes it problematic to draw any conclusions from C's limitations. And at that point, we might as well just allow B (or a new approval-directed agent) to play the same role.

Directions for future work

I think there are two basic directions for further research: (1) actually making a scheme like this work using existing technology, and (2) understanding and addressing possible concerns with scalability.

These directions mirror the most salient concerns with the proposal, namely (1) that its requirements, and especially the requirement for human understanding, significantly harm performance, and (2) that its reliance on the behavior of the (manipulable and fallible) humans may introduce serious scaling difficulties as the underlying learning systems improve.

Making it work

We can implement the agent above, but it's not going to be pretty. In particular, it won't give state of the art performance in most domains.

Actually getting this to work would require a lot of effort, and will no doubt involve radically changing the formulation above (probably resulting in something with no commonalities, a la the ship of Theseus). I think that this would tell us an awful lot about the practical prospects for this proposal in the long term, and in many cases would reveal more general insight into the nature of the AI control problem.

In the rest of this section I'll outline a few salient problems in this direction. I'll talk about these problems at more length in a future post, probably quite soon. For the most part, these projects address concerns with **explicability** and **efficacy**.

Explanation: many AI systems cannot "explain" their behavior to a human in even the weakest sense. For example, imagine an RL agent trained to fly a drone; it seems much easier to train based on end-to-end performance than to build a system which can actually explain to a human operator why a particular motor action is an appropriate response to a particular situation.

In order for our proposal to work, we must be able to train systems to produce this kind of explanation rather than directly training them to produce the desired outcome. This is a tall order.

Bootstrapping: scaling this kind of "explanation" (perhaps even to current AI capabilities) will require human operators to collaborate with one iteration of a learning system in order to understand the behavior of the next. This bootstrapping process has not been studied at all as far as I am aware of.

Imitation in unfavorable domains: In order to address concerns with scalability (see the next section), we may be interested in training systems to mimic human behavior rather than maximize approval. But this introduces a new class of practical difficulties that haven't received much attention—existing research practice uses imitation in some settings where it happens to be an effective approach, but has not explored applying imitation in unfavorable settings (e.g. those where AI systems can surpass human performance, or can only accomplish a task in a very inhuman way). If we wanted to apply mimicry more broadly for safety reasons, we would need to overcome the obvious difficulties in these unfavorable settings.

Addressing scalability concerns

In addition to (considerable) practical difficulties, it's worth thinking about the scalability of this proposal to very advanced AI capabilities. Would we encounter new problems that are exacerbated as capabilities increase?

This requires thinking about theoretical questions, though there may also be complementary empirical work. This section lays out some of those directions, which I'll examine at more length in a future post. For the most part, these represent concerns with **soundness**.

Manipulation resistance: This proposal involves training AI systems to produce explanations that the user will approve of. As our AI systems become increasingly sophisticated, they may find unintended ways to cause the user to express approval.

The main check against this concern is that the user can consult with a community of AI systems; before interacting with the system A the community of AI users has already have interacted with many very-slightly-less-powerful AI systems, and during interaction with A each user can consult other comparably-powerful AI systems.

We might divide these concerns into two classes: those that are *recoverable*—e.g. an attempt by A to deceive the user (which might be addressed by an appropriate rebuttal by B or by another approval-directed agent)—and those that are *unrecoverable*—e.g. a threat to the user or an attack that compromises the user's hardware, causing them to register approval without consulting agent B.

It’s unclear how robust our proposal is to these kinds of attacks, once the AI systems are significantly more intelligent than their users. This is a question that can be considered in much more detail. It could also be addressed by using mimicry instead of approval-direction (see above) or by having an appropriate internal organization rather than using reinforcement learners as black boxes (see below).

Simulations and inductive definitions: All systems along these lines define the “correct” behavior in terms of what the human user “would have done.” Intuitively this may give safe behavior, but there is no formal argument in some cases we may get surprising results.

One surprising possibility is that our algorithms reason about the possibility that they will be instantiated in a simulation, where the behavior of the “user” is anticipated to be very different from the behavior of the actual user (and may in fact be opposed to the user’s intentions).

I discuss this issue here; the most promising resolution is probably to apply pressure to the internal organization of our agents rather than treating the RL agents as a black box (see below). That post also discusses other possible solutions, and it’s far from clear that this problem will materialize in actual systems.

Internal organization: In order to avoid the previous two problems, we might want to build agents which think about problems in a way we would approve of, rather than trying to maximize the extent to which we would approve of their actions.

For example, if we had insight into the internal behavior of an agent then we might disapprove of a cognitive process which was searching for zero-day exploits which would enable them to gain maximal approval; so if internal cognitive processes were organized so as to maximize our approval *of those processes*, we would not expect our agent to perform that kind of search.

This idea suggests both theoretical and a practical research directions.

Practically, we can engage with the design of approval-directed systems built from existing technology and try to understand whether the internal behavior can be organized in a way that is itself-approval directed. This seems to ultimately be a question about the particular algorithms that prove useful.

For example, two common algorithmic technique are gradient ascent or trial-and-error, which effectively optimize internal weights by backward-chaining from the desired output behavior. In some sense this procedure inherently optimizes external behaviors. Is there a way to change this process so that user feedback can adjust internal processes? Can existing training procedures actually produce the kinds of manipulative or surprising behaviors we are concerned with, or do they only arise from explicit decision-theoretic reasoning? Can we build systems that are broken up into modules that e.g. separate out such decision-theoretic reasoning so that it can be influenced by additional requirements?

Theoretically, it's not clear how we would analyze the behavior of a system made from "approval-directed parts." How do we formulate the intended properties of the whole, and how do those properties relate to the properties of the parts? Filling in this theoretical gap would do quite a lot to bridge the gap between the approval-directed approach (and I suspect many more "common-sense" views) and the MIRI perspective. Unfortunately, it's not clear whether we should expect to find any clean theoretical picture, even if this will ultimately be a satisfactory solution.

Conclusion

This framework is not yet well understood, and not many people have spent time thinking about it. But to me it looks more tractable than other direct attacks on scalable AI control. For example, we can build approval-directed systems today, and many key long-term challenges may actually materialize as practical challenges with getting those actually-existing systems to work well. I think that is a big advantage, and almost a prerequisite for studying the issue within the AI field as it currently exists. (This will probably be my focus in the immediate future.)

Whether or not we are interested in approval-direction per se, the theoretical difficulties discussed in the "scalability concerns" section seem to apply to most practical work on AI alignment (as well as to common intuitions about why AI will be aligned-by-default). I suspect this family of questions should be a priority for researchers interested in the theoretical question of scalability—mainstream AI researchers interested in alignment are unlikely to be persuaded to pursue more abstract approaches unless they are convinced that these kinds of scalability concerns are really deal-breakers. Approval-directed agents provide a simple but concrete context in which to think about these concerns.