Efficient and safely scalable

Paul Christiano Follow

Mar 23, 2016 · 15 min read

Precisely defining the goal of AI control research seems quite difficult. This post gives preliminary definitions of **safe scalability** and **efficiency** for AI control protocols, taking a step towards formalization. Roughly, these properties say that "using better machine learning primitives results in better systems" and "the control scheme does not impose significant overhead."

I think these properties are probably sufficient conditions for success, but they are also probably too ambitious to be realistic goals. I discuss a few possible ways to weaken these definitions.

Both scalability and efficiency are defined with respect to a preference order $>^D$ which tells us when one algorithm is "better" another on some distribution D, according to the user's preferences. I won't offer any precise definition of $>^D$, but I'll discuss a few informal candidates.

Motivation

I'm interested in defining alignment formally for at least three reasons:

- Having a precise goal makes it easier to do good and well-targeted research. The AI control problem would feel much easier to me (both to work on and to talk to others about) if there were a precise, satisfactory, and achievable goal.
- A precise definition of alignment might be helpful when analyzing AI control schemes. For example, the analysis of ALBA calls for maintaining alignment as an inductive invariant as the agent becomes more powerful. Right now, there is little hope of making that argument formal.
- Trying to formalize alignment may shed light on what the key difficulties are, what assumptions are likely to be necessary, and so on. Trying to pin down slippery concepts is often a good idea .

Definitions

What is a control protocol?

Our AI control protocols will use machine learning primitives as building blocks, and construct a (hopefully aligned) AI out of them.

To instantiate a control protocol Align, we provide some set of learning primitives that are required by the protocol. Align then instantiates any number of copies of each of those primitives. Align may choose what inputs to provide to those instances, and may use their outputs however it likes. Align may also interact with the user arbitrarily.

For simplicity, throughout the post we will assume that Align is built from an RL algorithm, and write Align(A^{RL}) for the algorithm obtained by using A^{RL}. Note that Align can instantiate any number of distinct instances of A^{RL}, can provide each of them distinct rewards, and so on.

All of our definitions can be easily extended to any set of machine learning primitives, as long as we can define what it means for one implementation of a primitive to "outperform" another on a given distribution. I think that the definitions are most interesting when we can efficiently test whether one implementation outperforms another, and amongst such primitives RL is essentially universal (since we can use the test itself as a reward function).

Betterness

What does it mean for one algorithm to be better than another?

We won't answer that question. Instead, we take as given a family of preorders $<^D$ indexed by distributions D. These orders define when one program "outperforms" another on the distribution D, according to the user's preferences.

Intuitively, we can imagine some (unobserved) utility function U characterizing the user's preferences. U takes as input an (x, y) pair, and outputs a real number reflecting how good it is, according to the user's preferences, for a program to output y given input x.

Then we could define:

- $A \preccurlyeq^D B \iff \mathbb{E}[U(x, A(x))] \leq \mathbb{E}[U(x, B(x))]$,

where the expectations are taken over x drawn from D. We could also strengthen the condition by allowing A to score nearly as well on average over many episodes, or by requiring B to do at least as well as A for every utility function in some big set $\mathscr{U}$. These changes make $<^D$ a preorder rather than a total order.

We'll give a range of more plausible definitions in the final section, but this is a good concrete example to have in mind.

Safely scalable and efficient

Intuitively, a control scheme is *safely scalable* if using better RL algorithms never results in worse performance. A control scheme is *efficient* if it can realize any level of performance with low overhead.

- **Safely scalable (with respect to** $>^D$). Given any A^{RL}, B^{RL}, and the ability to sample from some distribution D with $\textsc{Align}(A^{RL}) >^D \textsc{Align}(B^{RL})$, we can sample from a distribution D^{RL} over RL episodes on which A^{RL} outperforms B^{RL}.
- **Efficient (with respect to** $>^D$). Given any B, we can implement an RL agent B^{RL} such that, for every distribution D from which we can sample, $\textsc{Align}(B^{RL}) \geq^D B$.

Efficiency is quantified by how much more time $\textsc{Align}(B^{RL})$ takes than B itself, and how often it queries the user. I'll say that a scheme is "**asymptotically efficient**" if the difference in running times, and the number of queries to the user, is sublinear in the number of episodes T.

In order to be achievable this definition probably needs to be weakened. I think the most plausible way to weaken it is to make additional assumptions about the agent B in **efficiency**. For example, we could focus our attention on a particular approach to building AI systems, and assume that B is the kind of agent that might be produced by that approach. Particularly interesting are structural assumptions about how B itself is built out of the same building blocks that are available to $\textsc{Align}$.

Hard to beat

Together efficiency and safe scalability imply a third property:

- **Hard to beat**. Given any RL agent A^{RL}, any agent B, and the ability to sample from a distribution D with $B >^D \textsc{Align}(A^{RL})$, we can implement an agent B^{RL} and sample from a distribution D^{RL} over RL episodes on which B^{RL} outperforms A^{RL}.

If an algorithm is “hard to beat,” then the only way to make it better (according to $>^{D}$) is to improve the underlying RL algorithms. In some sense this is the strongest form of optimality that we can realistically hope for, since improving our RL algorithms will allow us to build “better” AI systems for any reasonable notion of “better.”

To see that (efficient + scalable → hard to beat), apply efficiency to find an agent $\textsc{Align}(B^{RL}) \geq^{D} B$, use transitivity to infer that $\textsc{Align}(B^{RL}) >^{D} \textsc{Align}(A^{RL})$, and then to use safe scalability to sample from a distribution where B^{RL} outperforms A^{RL}.

Being hard to beat is slightly weaker than being efficient + scalable while being almost as comforting. So it might also be useful as an easier goal.

Restrictions on the building blocks

Rather than working with a generic RL algorithm, we might want to work with an RL algorithm that satisfies some additional property. For example, $\textsc{Align}(A^{RL})$ might only be safely scalable if A^{RL} is able to make good enough predictions about humans, or if A^{RL} is “transparent” in an appropriate sense.

These restrictions can be incorporated into the definition of safe scalability and efficiency—in safe scalability, we can strengthen the hypothesis by assuming that the learning algorithms satisfy the restrictions, and in efficiency we can strengthen the conclusion by assuming that the learning algorithms satisfy the restrictions.

To make the task easiest, we could strengthen the hypothesis of safe scalability *without* strengthening the conclusion of efficiency. For example, if we want to work with “transparent” RL algorithms, we will probably not be able to strengthen the conclusion of efficiency—we won’t be able to turn a black-box algorithm B into a transparent RL algorithm B^{RL}. So we could instead aim for a scheme that is safely scalable when applied with transparent RL algorithms, and that is efficient when we are allowed to use arbitrary RL algorithms. The resulting control scheme would only be practically efficient to the extent that there are transparent nearly-state-of-the-art RL algorithms.

For now I am interested in schemes that work under *any* remotely plausible assumptions:

- Even a conditional result would be a big advance beyond our current understanding.
- Conditional results could clarify our understanding of when AI control will and won't succeed.
- Conditional results present natural targets for differential AI progress. For example, if we could formulate a transparency condition that was sufficient for building safe+efficient AI, this could help clarify the goals of research on transparency.

Related building blocks

We might also work with a number of different building blocks, and require them to have some relationship to each other. For example, we might want to use both a classifier and an RL agent, and assume that the classifier is in some sense more powerful than the RL agent (though it's not yet clear to me what assumption of this form would actually be useful). Or we might work with a sequence of RL agents, and assume that each is only slightly more powerful than the last.

Non-solutions

Plain RL (+counterfactual oversight)

The simplest behavior for Align is to simply consult the underlying RL agent directly, and then to define a reward that induces aligned behavior.

Perhaps the simplest reward function is a human's evaluation: a human watches the agent and presses the reward button when the agent behaves well.

If the human has to monitor the agent during every episode, then this constitutes a huge efficiency hit and so our algorithm won't be reasonably efficient.

More realistically, we could train our agent on a small subset of the data (for which the human does have to monitor the agent and press the reward button) and then use the learned policy to act even when there is no human monitor. If necessary, we could train online as in counterfactual oversight.

This approach is asymptotically efficient (as long as the size of the training set is sublinear in the total number of episodes, as would be typical).

However, this scheme is **not** safely scalable for a reasonable order $>^D$.

The problem is that the user's judgments are not exactly aligned with the user's preferences. For any given input *x*, there are some outputs *y* for which the user assigns *y* an "erroneously" high expected value, whether due to error, to bias, or simply to not having enough time to understand the consequences of a proposed action.

So let A^{RL} be an RL agent which outputs actions which the user mistakenly rates highly, and let B^{RL} be an RL agent which outputs actions that are better than A^{RL}'s outputs but which are rated lower by the user. As long as the user's judgments don't perfectly preserve the ordering of outcomes, then we can construct such a pair. Then we can have $\text{Align}(A^{RL}) >^D \text{Align}(B^{RL})$ for most distributions D, while B^{RL} outperforms A^{RL} on *every* distribution D^{RL}

Plain imitation learning (+counterfactual oversight)

Given observations of a human expert, we can give A^{RL} the problem "predict what output the human expert will produce on input *x*."

After A^{RL} produces a prediction, with small probability we ask the human to actually produce an output *y*, and then we provide A^{RL} a payoff depending on the quality of its prediction. (There are a number of possible ways to score A^{RL}, most involving two separate learners. For example, we could use a generative adversarial model, with a second instance of A^{RL} trying to distinguish the predicted output from the real human output. Or we could score A^{RL} based on a variational lower bound on the log probability it assigned to the particular prediction made by the user.)

This kind of imitation learning is not safely scalable: an agent which chooses superhuman actions will become strictly worse if we make them a perfect human-predictor, yet their performance in the RL game will improve.

(Intuitively, imitation *is* safely scalable but *is not* efficient—this example highlights a way in which our definitions fail to map well to the intuitive definitions of these concepts.)

ALBA

The current version of ALBA is not safely scalable + efficient, even under the generous assumptions made in the post. The key problem is its reliance on a sequence of throttled RL subagents.

The missing ingredients, roughly speaking, are the following:

- **Speedup**. Given black-box access to the agent B, we can produce a *strictly less capable* agent B' such that Bootstrap(B') $\geq^D$ B.
- **Informed oversight**. See here.

Speedup is impossible without some additional assumptions on B, and informed oversight probably also requires some additional assumptions (in order to make transparency possible). It's not clear what kind of assumptions on B might be sufficient to carry out the analysis, and overall it looks like a daunting project.

Limitations

Too strong

I think the biggest problem with this definition is that it is too strong.

It's not so strong as to be *obviously* impossible. But it looks *almost* obviously impossible. The discussions of RL and ALBA illustrate why the definition is so strong:

- In order to turn a black box agent B into an agent B^{RL} with Align(B^{RL}) $\geq^D$ B, we are essentially forced to take B^{RL} = B (since we can't produce other derivative agents using black-box access to B).
- Then Align is essentially forced to be a training scheme for RL agents.
- So in order to be safely scalable, Align needs to evaluate of the quality of the agent's decisions "well enough" that optimizing its evaluations optimizes $>^D$.
- Moreover, Align can't really use the RL agent's help to make those decisions—if Align is merely a training procedure, the RL agent need not output anything except on the support of D, and so we can't get any useful work out of the agent. Thus Align is using the same evaluations for every agent.
- If Align evaluates the agent's behavior "well enough" for an arbitrary agent, then Align must be evaluating the agent's behavior perfectly.

- It seems infeasible to produce such a perfect evaluations for any interesting $>^D$.

How might we weaken the definition?

- Place some restriction on the set of agents B that we consider in **efficiency**. For example, we may restrict attention to the kinds of agents that could be produced by some particular AI research project in AI. I think that this is by far the most promising approach.
- As discussed in the section **Restrictions on building blocks**, we could only require safe scalability for a certain class of RL agents, thus moving some of the work to ensuring that state-of-the-art RL agents have the required properties.
- We could use relations $>^D$ that evaluate agents holistically in terms of a *description* of the distribution D (see below). For example, we might say that "A $>^D$ B if the human believes that A would outperform B on the distribution D." I don't really see a way to make this work, but it might be worth thinking about.
- We could settle for an agent which is hard to beat instead of both efficient and safely scalable. I don't think this really addresses the difficulty described above, but it does give us a tiny bit more traction.
- We could swap the quantifier order, giving us access to B and D when trying to construct an agent B^{RL} with Align(B^{RL}) $\geq^D$ B. I don't think this will help.

I expect there are many other ways to weaken the definition, and of course we could pursue some combination of the above.

Improving RL algorithms is quite broad

Even if Align is efficient and safely scalable, Align(A^{RL}) isn't necessarily *good* even according to $>^D$. In order to make Align(A^{RL}) actually be good, we may need to improve A^{RL}. In some sense this is obvious and inevitable—it's like saying that even if we solve the control problem, AI progress will still make our AI systems work better.

But in particular, the alignment of Align(A^{RL}) may depend on how A^{RL} performs on some very unnatural distribution over RL problems (e.g.

on how well A^{RL} is able to predict the results of human deliberation about moral questions).

Given how strong safe scalability and efficiency are, I don't think this is a problem for this particular definition. That is, any such "unnatural" distribution over RL problems would be necessary to achieving good behavior, even for very weak agents:

- Given any aligned agent B we can apply efficiency to obtain an RL agent B^{RL}.
- If B^{RL} always does well on these unnatural instances, then they were in some strong sense *necessary* in order to get good behavior. But note that B might be quite weak, so these "unnatural" instances were necessary even to getting good behavior out of weak AI systems.
- If B^{RL} does poorly on these unnatural instances, then we can consider a very powerful agent A^{RL} that does equally poorly.
- By safe scalability, Align(A^{RL}) remains at-least-as-good-as-B for that RL agent A^{RL}.

This argument would still go through if we restricted B to the kinds of AI's that we might actually produce. (The conclusion would be that the unnatural RL instances are in fact necessary, *for the kinds of AI that we might actually build*, which is just as good.)

If we weakened the definition enough we might encounter a more troubling version of this unnatural-RL-instances problem, but even then I think that reducing the alignment problem to a concrete RL problem would probably represent significant progress in and of itself. So overall I'm not concerned about this potential bug.

Missing definition of $>^D$

We don't provide any definition of $>^D$; that definition needs to be quite subtle and is doing a lot of the work of formalizing our goals. See the discussion in the next section.

Defining $>^D$

Human judgment

For sufficiently subhuman agents we could define:

- Draw a random *x* from D, compute A(*x*) and B(*x*), and give them both to a human judge (along with the input x) to decide which is better. A $>^{D}$ B if the human prefers A(*x*) with probability at least 2/3.

For this definition of <, the RL agent described above is in fact efficient and safely scalable.

But being safely scalable in this sense doesn't seem very useful, since the human's judgment about "which action is good" need not be correct, and the human would substantially prefer an agent that chooses actions which are *actually* good.

Human judgment on reflection

We could try to fix this definition with an alternative informal definition:

- Draw a random *x* from D, compute A(*x*) and B(*x*). Give a human judge an extremely long time to reflect, and access to extremely powerful tools. A $>^{D}$ B if, upon extensive reflection, the human prefers A(*x*) with probability at least 2/3.

This definition leads to a very strong notion of safe scalability. Unfortunately that strong notion is *too* strong, and it is clearly intractable to build a hard-to-beat agent in this sense.

For example, consider some hard question that bears on what strategies will have good long-term effects. This question is beyond our abilities currently, but we will be able to answer it with sufficient reflection. If B guesses the correct answer to that question and A does not, then A $<^{D}$ B, where D is the a point distribution concentrated on a decision where that question matters. If B guesses the wrong answer, then B $<^{D}$ A. But by efficiency + safe scalability, we can essentially produce a pair of agents A^{RL} and B^{RL} and sample from a distribution D^{RL} such that A $<^{D}$ B iff B^{RL} outperforms A^{RL} on D ^{RL}. But then we can test which of A and B is correct about the hard question, contradicting its supposed hardness.

Holistic judgment

A very different approach would be to directly elicit the human's judgment about the relative merits of A and B. That is:

- Give a human judge a description of A and B, and the sampling procedure for D, and ask them which they expect to perform better. $A >^{D} B$ if the human thinks that A would make significantly better decisions than B, on inputs sampled from D.

In some sense this definition gets very directly at what we care about. Suppose that some alternative AI control research program produced B, and that we would be happier using B than A. Then we are precisely satisfying this definition of $A <^{D} B$.

An obvious problem with this approach is that determining which of A and B is better may itself be a very hard problem—indeed, it’s easy to believe that it includes all of the difficulty of the alignment problem.

The definition of efficiency still has teeth if we restrict our attention to algorithms B that “explain themselves,” e.g. that are accompanied with significant documentation and explanation. The description of the agent B could embody the entire output of the alternative research program that produced it.

So this definition of betterness is adequate if we think that there are efficient and good decision-making protocols which are *demonstrably* good to existing humans. This definition is very unsatisfying if we think that evaluating a possible proposal, even given the best available arguments, is the core difficulty of AI control. This might be either because those arguments are necessarily extremely complex, or because there will be many bad proposals that are also supported by extremely convincing-looking arguments.

What we really want

Intuitively, I would like a definition along the lines of:

- Draw a random x from D, compute A(x) and B(x). Give a human judge the same information, abilities, and insights, that A and B used to compute these quantities. $A >^{D} B$ if the human prefers A(x) with probability at least 2/3.

This definition smuggles all of the complexity into imagining that the human has the same “information, abilities, and insights” as the AI they are evaluating. I don’t have any candidate formalization of this idea, nor am I especially optimistic about being able to formalize it.

I do feel like I can reason about this definition intuitively and that it roughly captures my intuitive desiderata. This makes me more optimistic that there is *some* satisfactory definition of >.

Note that this definition is closely related to the goal in the informed oversight problem, which is roughly to ensure that the overseer “knows everything the AI knows.” In the informed oversight problem we are willing to assume that the overseer is significantly more powerful than the system they are overseeing. That may well be a necessary assumption to actually ensure that the overseer “knows everything the AI knows,” but it probably isn’t needed to define what it would mean for the overseer to “know everything the AI knows.”

Conclusion

We can try to define the goals of AI control by thinking about how AI systems relate to the underlying machine learning primitives. Such a framework wouldn’t cover all possible approaches to AI control, but where applicable it could be a great way to organize research and a useful analysis tool.

This post gave a step in that direction, but did not yet succeed. I would love to see other attempts, and I think there is a good chance that it will be possible to find a satisfying problem statement for AI control.