Approval-directed agents

Paul Christiano

November 22, 2018

Note: This is the first post from ***part two: basic intuitions*** *of the sequence on iterated amplification. The second part of the sequence outlines the basic intuitions that motivate iterated amplification. I think that these intuitions may be more important than the scheme itself, but they are considerably more informal.*

Research in AI is steadily progressing towards more flexible, powerful, and autonomous goal-directed behavior. This progress is likely to have significant economic and humanitarian benefits: it helps make automation faster, cheaper, and more effective, and it allows us to automate *deciding what to do.*

Many researchers expect goal-directed machines to predominate, and so have considered the long-term implications of this kind of automation. Some of these implications are worrying: if sophisticated artificial agents pursue their own objectives and are as smart as we are, then the future may be shaped as much by their goals as by ours.

Most thinking about "AI safety" has focused on the possibility of goal-directed machines, and asked how we might ensure that their goals are agreeable to humans. But there are other possibilities.

In this post I will flesh out one alternative to goal-directed behavior. I think this idea is particularly important from the perspective of AI safety.

Approval-directed agents

Consider a human Hugh, and an agent Arthur who uses the following procedure to choose each action:

Estimate the expected rating Hugh would give each action if he considered it at length. Take the action with the highest expected rating.

I'll call this "approval-directed" behavior throughout this post, in contrast with goal-directed behavior. In this context I'll call Hugh an "overseer."

Arthur's actions are rated more highly than those produced by any alternative procedure. That's comforting, but it doesn't mean that Arthur is optimal. An optimal agent may make decisions that have *consequences* Hugh would approve of, even if Hugh can't anticipate those consequences himself. For example, if Arthur is playing chess he should make moves that are actually good—not moves that Hugh thinks are good.

The quality of approval-directed decisions is limited by the *minimum* of Arthur's ability and Hugh's ability: Arthur makes a decision only if it looks good to both Arthur and Hugh. So why would Hugh be interested in this proposal, rather than doing things himself?

- Hugh doesn't actually rate actions, he just participates in a hypothetical rating process. So Hugh can oversee many agents like Arthur at once (and spend his actual time relaxing on the beach). In many cases, this is the whole point of automation.

- Hugh can (hypothetically) think for a very long time about each decision—longer than would be practical or cost-effective if he had to actually make the decision himself.
- Similarly, Hugh can think about Arthur's decisions at a very low level of detail. For example, Hugh might rate a chess-playing AI's choices about how to explore the game tree, rather than rating its final choice of moves. If Arthur is making billions of small decisions each second, then Hugh can think in depth about each of them, and the resulting system can be much smarter than Hugh.
- Hugh can (hypothetically) use additional resources in order to make his rating: powerful computers, the benefit of hindsight, many assistants, very long time periods.
- Hugh's capabilities can be gradually escalated as needed, and one approval-directed system can be used to bootstrap to a more effective successor. For example, Arthur could advise Hugh on how to define a better overseer; Arthur could offer advice in real-time to help Hugh be a better overseer; or Arthur could directly act as an overseer for his more powerful successor.

In most situations, I would expect approval-directed behavior to capture the benefits of goal-directed behavior, while being easier to define and more robust to errors.

Advantages

Facilitate indirect normativity

Approval-direction is closely related to what Nick Bostrom calls "indirect normativity"—describing what is good indirectly, by describing how to tell what is good. I think this idea encompasses the most credible proposals for defining a powerful agent's goals, but has some practical difficulties.

Asking an overseer to evaluate *outcomes* directly requires defining an **extremely** intelligent overseer, one who is equipped (at least in principle) to evaluate the entire future of the universe. This is probably impractical overkill for the kinds of agents we will be building in the near future, who *don't* have to think about the entire future of the universe.

Approval-directed behavior provides a more realistic alternative: start with simple approval-directed agents and simple overseers, and scale up the overseer and the agent in parallel. I expect the approval-directed dynamic to converge to the desired limit; this requires only that the simple overseers approve of scaling up to more powerful overseers, and that they are able to recognize appropriate improvements.

Avoid lock-in

Some approaches to AI require "locking in" design decisions. For example, if we build a goal-directed AI with the wrong goals then the AI might never correct the mistake on its own. For sufficiently sophisticated AI's, such mistakes may be very expensive to fix. There are also more subtle forms of lock-in: an AI may also not be able to fix a bad choice of decision-theory, sufficiently bad priors, or a bad attitude towards infinity. It's hard to know what other properties we might inadvertently lock-in.

Approval-direction involves only extremely minimal commitments. If an approval-directed AI encounters an unforeseen situation, it will respond in the way that we most approve of. We don't need to make a decision until the situation actually arises.

Perhaps most importantly, an approval-directed agent can correct flaws in its own design, and will search for flaws if we want it to. It can change its own decision-making procedure, its own reasoning process, and its own overseer.

Fail gracefully

Approval-direction seems to “fail gracefully:” if we slightly mess up the specification, the approval-directed agent probably won’t be actively malicious. For example, suppose that Hugh was feeling extremely apathetic and so evaluated proposed actions only superficially. The resulting agent would not aggressively pursue a flawed realization of Hugh’s values; it would just behave lackadaisically. The mistake would be quickly noticed, unless Hugh deliberately approved of actions that concealed the mistake.

This looks like an improvement over misspecifying goals, which leads to systems that are actively opposed to their users. Such systems are motivated to conceal possible problems and to behave maliciously.

The same principle sometimes applies if you define the right overseer but the agent reasons incorrectly about it, if you misspecify the entire rating process, or if your system doesn’t work quite like you expect. Any of these mistakes could be serious for a goal-directed agent, but are probably handled gracefully by an approval-directed agent.

Similarly, if Arthur is smarter than Hugh expects, the only problem is that Arthur won’t be able to use all of his intelligence to devise excellent plans. This is a serious problem, but it can be fixed by trial and error—rather than leading to surprising failure modes.

Is it plausible?

I’ve already mentioned the practical demand for goal-directed behavior and why I think that approval-directed behavior satisfies that demand. There are other reasons to think that agents might be goal-directed. These are all variations on the same theme, so I apologize if my responses become repetitive.

Internal decision-making

We assumed that Arthur can predict what actions Hugh will rate highly. But in order to make these predictions, Arthur might use goal-directed behavior. For example, Arthur might perform a calculation because he believes it will help him predict what actions Hugh will rate highly. Our apparently approval-directed decision-maker may have goals after all, on the inside. Can we avoid this?

I think so: Arthur’s internal decisions could also be approval-directed. Rather than performing a calculation because it will help make a good prediction, Arthur can perform that calculation because Hugh would rate this decision highly. If Hugh is coherent, then taking individual steps that Hugh rates highly leads to overall behavior that Hugh would approve of, just like taking individual steps that maximize X leads to behavior that maximizes X.

In fact the result may be more desirable, from Hugh’s perspective, than maximizing Hugh’s approval. For example, Hugh might incorrectly rate some actions highly, because he doesn’t understand them. An agent maximizing Hugh’s approval might find those actions and take them. But if the agent was internally approval-directed, then it wouldn’t try to exploit errors in Hugh’s ratings. Actions that lead to reported approval but not real approval, don’t lead to approval for approved reasons

Turtles all the way down?

Approval-direction stops making sense for low-level decisions. A program moves data from register A into register B because that’s what the next instruction says, not because that’s what Hugh would approve of. After all, deciding whether Hugh would approve itself requires moving data from one register to another, and we would be left with an infinite regress.

The same thing is true for goal-directed behavior. Low-level actions are taken because the programmer chose them. The programmer may have chosen them because she thought they would help the system achieve its goal, but the actions themselves are performed because that's what's in the code, not because of an explicit belief that they will lead to the goal. Similarly, actions might be performed because a simple heuristic suggests they will contribute to the goal—the heuristic was chosen or learned because it was expected to be useful for the goal, but the action is motivated by the heuristic. Taking the action doesn't involve thinking about the heuristic, just following it.

Similarly, an approval-directed agent might perform an action because it's the next instruction in the program, or because it's recommended by a simple heuristic. The program or heuristic might have been chosen to result in approved actions, but the taking the action doesn't involve reasoning about approval. The aggregate effect of using and refining such heuristics is to effectively do what the user approves of.

In many cases, perhaps a majority, the heuristics for goal-directed and approval-directed behavior will coincide. To answer "what do I want this function to do next?" I very often ask "what do I want the end result to be?" In these cases the difference is in how we think about the behavior of the overall system, and what invariants we try to maintain as we design it.

Relative difficulty?

Approval-directed subsystems might be harder to build than goal-directed subsystems. For example, there is much more data of the form "X leads to Y" than of the form "the user approves of X." This is a typical AI problem, though, and can be approached using typical techniques.

Approval-directed subsystems might also be easier to build, and I think this is the case today. For example, I recently wrote a function to decide which of two methods to use for the next step of an optimization. Right now it uses a simple heuristic with mediocre performance. But I could also have labeled some examples as "use method A" or "use method B," and trained a model to predict what I would say. This model could then be used to decide when to use A, when to use B, and when to ask me for more training data.

Reflective stability

Rational goal-directed behavior is reflectively stable: if you want X, you generally want to continue wanting X. Can approval-directed behavior have the same property?

Approval-directed systems inherit reflective stability (or instability) from their overseers. Hugh can determine whether Arthur "wants" to remain approval-directed, by approving or disapproving of actions that would change Arthur's decision-making process.

Goal-directed agents want to be wiser and know more, though their goals are stable. Approval-directed agents also want to be wiser and know more, but they also want their overseers to be wiser and know more. The overseer is not stable, but the overseer's values are. This is a feature, not a bug.

Similarly, an agent composed of approval-directed subsystems overseen by Hugh is *not* the same as an approval-directed agent overseen by Hugh. For example, the composite may make decisions too subtle for Hugh to understand. Again, this is a feature, not a bug.

Black box search

(Note: I no longer agree with the conclusions of this section. I now feel that approval-directed agents can probably be constructed out of powerful black-box search (or stochastic gradient descent); my main priority is now either handling this setting or else understanding exactly what the obstruction is. Ongoing work in this direction is collected at ai-control, and will hopefully be published in a clear format by the end of 2016.)

Some approaches to AI probably can't yield approval-directed agents. For example, we could perform a search which treats possible agents as a black boxes and measures their behavior for signs of intelligence. Such a search could (eventually) find a human-level intelligence, but would give us very crude control over how that intelligence was applied. We could get some kind of goal-directed behavior by selecting for it, but selecting for approval-directed behavior would be difficult:

1. The paucity of data on approval is a huge problem in this setting. (Note: semi-supervised reinforcement learning is an approach to this problem.)
2. You have no control over the internal behavior of the agent, which you would expect to be optimized for pursuing a particular goal: maximizing whatever measure of "approval" that you used to guide your search. (Note: I no longer endorse this argument as written; reward engineering is a response to the substance of this concern.)
3. Agents who maximized your reported approval in test cases need not do so in general, any more than humans are reliable reproductive-fitness-maximizers. (Note: red teaming is an approach to this problem.)

But [1] and especially [3] are also problems when designing a goal-directed agent with **agreeable** goals, or indeed any particular goals at all. Though approval-direction can't deal with these problems, they aren't new problems.

Such a black-box search—with little insight into the internal structure of the agents—seems worrying no matter how we approach AI safety. Fortunately, it also seems unlikely (though not out of the question).

A similar search is more likely to be used to produce internal components of a larger system (for example, you might train a neural network to identify objects, as a component of a system for navigating an unknown environment). This presents similar challenges, concerning robustness and unintended behaviors, whether we are designing a goal-directed or approval-directed agent.

"Implementation" details

So far I've talked about approval-direction imprecisely. Maybe I'm talking about something incoherent, which has desirable properties only in the same sense as a four-sided triangle—vacuously. I won't really be able to dispel this concern here, but I'll at least take some steps.

How do you define approval?

Eventually you would have to actually write code implementing approval-directed behavior. What might that code look like? I want to set aside the problem "what does a sophisticated AI look like?" since I obviously don't know. So let's suppose we had some black box that did all of the hard work. I'll consider a few cases for what the black box does, ranging from "easy to work with" to "very hard to work with."

(Note: I now believe that we can target AI systems trained (nearly) end-to-end with gradient descent, which is most similar to "learning from examples.")

Natural language

As an easy case, suppose we have a natural language question-answering system, which can assign a probability to any natural language sentence. In this case, we ask the question:

> "Suppose that Hugh understood the current situation, was asked 'on a scale from 0 to 1, how good is the action a_?_' and was given a few hours to determine his answer. What would his answer be?"

We then loop over each action a and take the action with the highest expected answer.

In this framework, it is easy to replace Hugh by a more powerful overseer—all you have to do is specify the replacement in natural language.

“Math intuition module”

At an opposite extreme, suppose we have a “math intuition module,” a system which can assign probabilities only to *perfectly* precise statements—perhaps of the form “algorithm A returns output y on input x.”

I’ve written about defining “approval upon reflection” algorithmically (see here, here). These definition can be used to define approval-directed behavior completely precisely. I’m pretty hesitant about these definitions, but I do think it is promising that we can get traction even in such an extreme case.

In reality, I expect the situation to be somewhere in between the simple case of natural language and the hard case of mathematical rigor. Natural language is the case where we share all of our concepts with our machines, while mathematics is the case where we share only the most primitive concepts. In reality, I expect we will share some but not all of our concepts, with varying degrees of robustness. To the extent that approval-directed decisions are robust to imprecision, we can safely use some more complicated concepts, rather than trying to define what we care about in terms of logical primitives.

Learning from examples

In an even harder case, suppose we have a function learner which can take some labelled examples *f(x)* = y and then predict a new value *f(x’).* In this case we have to define “Hugh’s approval” directly via examples. I feel less comfortable with this case, but I’ll take a shot anyway.

In this case, our approval-directed agent Arthur maintains a probabilistic model over sequences **observation**[T] and **approval**[T](a). At each step T, Arthur selects the action a maximizing **approval**[T](a). Then the timer T is incremented, and Arthur records **observation**[T+1] from his sensors. Optionally, Hugh might specify a value **approval**[*t](a’)* for any time t and any action *a’.* Then Arthur updates his models, and the process continues.

Like AIXI, if Arthur is clever enough he eventually learns that **approval**[T](_a)_refers to whatever Hugh will retroactively input. But unlike AIXI, Arthur will make no effort to manipulate these judgments. Instead he takes the action maximizing his expectation of **approval**[T]—i.e., his prediction about what Hugh will say in the future, if Hugh says anything at all. (This depends on his self-predictions, since what Hugh does in the future depends on what Arthur does now.)

At any rate, this is quite a lot better than AIXI, and it might turn out fine if you exercise appropriate caution. I wouldn’t want to use it in a high-stakes situation, but I think that it is a promising idea and that there are many natural directions for improvement. For example, we could provide further facts about **approval** (beyond example values), interpolating continuously between learning from examples and using an explicit definition of the approval function. More ambitiously, we could implement “approval-directed learning,” preventing it from learning complicated undesired concepts.

How should Hugh rate?

So far I’ve been very vague about what Hugh should actually do when rating an action. But the approval-directed behavior depends on how Hugh decides to administer approval. How should Hugh decide?

If Hugh expects action a to yield better consequences than action b, then he should give action a a higher rating than action b. In simple environments he can simply pick the best action, give it a rating of 1, and give the other options a rating of 0.

If Arthur is so much smarter than Hugh that he knows exactly what Hugh will say, then we might as well stop here. In this case, approval-direction amounts to Arthur doing exactly what Hugh instructs: “the minimum of Arthur’s capabilities and Hugh’s capabilities” is equal to “Hugh’s capabilities.”

But most of the time, Arthur won't be able to tell exactly what Hugh will say. The numerical scale between 0 and 1 exists to accomodate Arthur's uncertainty.

To illustrate the possible problems, suppose that Arthur is considering whether to drive across a bridge that may or may not collapse. Arthur thinks the bridge will collapse with 1% probability. But Arthur also think that Hugh knows for sure whether or not the bridge will collapse. If Hugh always assigned the optimal action a rating of 1 and every other action a rating of 0, then Arthur would take the action that was most likely to be optimal—driving across the bridge.

Hugh should have done one of two things:

- Give a bad rating for risky behavior. Hugh should give Arthur a high rating only if he drives across the bridge and *knows that it is safe*. In general, give a rating of 1 to the best action *ex ante*.
- Assign a very bad rating to incorrectly driving across the bridge, and only a small penalty for being too cautious. In general, give ratings that reflect the utilities of possible outcomes—to the extent you know them.

Probably Hugh should do both. This is easier if Hugh understands what Arthur is thinking and why, and what range of possibilities Arthur is considering.

Other details

I am leaving out many other important details in the interest of brevity. For example:

- In order to make these evaluations Hugh might want to understand what Arthur is thinking and why. This might be accomplished by giving Hugh enough time and resources to understand Arthur's thoughts; or by letting different instances of Hugh "communicate" to keep track of what is going on as Arthur's thoughts evolve; or by ensuring that Arthur's thoughts remains comprehensible to Hugh (perhaps by using approval-directed behavior at a lower level, and only approving of internal changes that can be rendered comprehensible).
- It is best if Hugh optimizes his ratings to ensure the system remains robust. For example, in high stakes settings, Hugh should sometimes make Arthur consult the real Hugh to decide how to proceed—even if Arthur correctly knows what Hugh wants. This ensures that Arthur will seek guidance when he *incorrectly* believes that he knows what Hugh wants.

…and so on. The details I *have* included should be considered illustrative at best. (I don't want anyone to come away with a false sense of precision.)

Problems

It would be sloppy to end the post without a sampling of possible pitfalls. For the most part these problems have more severe analogs for goal-directed agents, but it's still wise to keep them in mind when thinking about approval-directed agents in the context of AI safety.

My biggest concerns

I have three big concerns with approval-directed agents, which are my priorities for follow-up research:

- Is an approval-directed agent generally as useful as a goal-directed agent, or does this require the overseer to be (extremely) powerful? Based on the ideas in this post, I am cautiously optimistic.
- Can we actually define approval-directed agents by examples, or do they already need a shared vocabulary with their programmers? I am again cautiously optimistic.

- Is it realistic to build an intelligent approval-directed agent without introducing goal-directed behavior internally? I think this is probably the most important follow-up question. I would guess that the answer will be "it depends on how AI plays out," but we can at least get insight by addressing the question in a variety of concrete scenarios.

Motivational changes for the overseer

"What would I say if I thought for a *very* long time?" might have a surprising answer. The very process of thinking harder, or of finding myself in a thought experiment, might alter my priorities. I may care less about the real world, or may become convinced that I am living in a simulation.

This is a particularly severe problem for my proposed implementation of indirect normativity, which involves a truly outlandish process of reflection. It's still a possible problem for defining approval-direction, but I think it is much less severe.

"What I would say after a few hours," is close enough to real life that I wouldn't expect my thought process to diverge too far from reality, either in values or beliefs. Short time periods are much easier to predict, and give less time to explore completely unanticipated lines of thought. In practice, I suspect we can also define something like "what I would say after a few hours of sitting at my desk under completely normal conditions," which looks particularly innocuous.

Over time we will build more powerful AI's with more powerful (and perhaps more exotic) overseers, but making these changes gradually is much easier than making them all at once: small changes are more predictable, and each successive change can be made with the help of increasingly powerful assistants.

Treacherous turn

If Hugh inadvertently specifies the wrong overseer, then the resulting agent might be motivated to deceive him. *Any* rational overseer will be motivated to approve of actions that look reasonable to Hugh. If they don't, Hugh will notice the problem and fix the bug, and the original overseer will lose their influence over the world.

This doesn't seem like a big deal—a failed attempt to specify "Hugh" probably won't inadvertently specify a different Hugh-level intelligence, it will probably fail innocuously.

There are some possible exceptions, which mostly seem quite obscure but may be worth having in mind. The learning-from-examples protocol seems particularly likely to have problems. For example:

- Someone other than Hugh might be able to enter training data for **approval**$[T](a)$. Depending on how Arthur is defined, these examples might influence Arthur's behavior as soon as Arthur expects them to appear. In the most pathological case, these changes in Arthur's behavior might have been the very reason that someone had the opportunity to enter fraudulent training data.
- Arthur could accept the motivated simulation argument, believing himself to be in a simulation at the whim of a simulator attempting to manipulate his behavior.
- The simplest explanation for Hugh's judgments may be a simple program motivated to "mimic" the series **approval**$[T]$ and **observation**$[T]$ in order to influence Arthur.

Ignorance

An approval-directed agent may not be able to figure out what I approve of.

I'm skeptical that this is a serious problem. It falls under the range of predictive problems I'd expect a sophisticated AI to be good at. So it's a standard objective for AI research, and AI's that can't make such predictions probably have significantly sub-human ability to act in the world. Moreover, even a fairly weak reasoner can learn generalizations like "actions that lead to Hugh getting candy, tend to be approved of" or "actions that take control away from Hugh, tend to be disapproved of."

If there is a problem, it doesn't seem like a serious one. Straightforward misunderstandings will lead to an agent that is inert rather than actively malicious (see the "Fail gracefully" section). And deep misunderstandings can be avoided, by Hugh approving of the decision "consult Hugh."

Conclusion

Making decisions by asking "what **action** would your owner most approve of?" may be more robust than asking "what **outcome** would your owner most approve of?" Choosing actions directly has limitations, but these might be overcome by a careful implementation.

More generally, the focus on achieving safe goal-directed behavior may have partially obscured the larger purpose of the AI safety community, which should be achieving safe and *useful* behavior. It may turn out that goal-directed behavior really is inevitable or irreplaceable, but the case has not yet been settled.

This essay was originally posted here on 1st December 2014.

Tomorrow's AI Alignment Forum sequences post will be 'Fixed Point Discussion' by Scott Garrabrant, in the sequence 'Fixed Points'.

The next posts in this sequence will be 'Approval directed bootstrapping' and 'Humans consulting HCH', two short posts which will come out on Sunday 25th November.