The informed oversight problem

Paul Christiano Follow
Mar 1, 2016 · 13 min read

My recent approach to AI alignment faces a number of open problems, but I think the most serious is what I'll call the "informed oversight" problem. [ETA: By the end of 2016 this problem no longer seems like the most serious.]

I don't think that this problem is distinctive to my approach, and it may turn out that any efficient and aligned AI system must effectively solve it.

In this post I describe the problem, provide a simple example and model, explain why I think it is one of the largest missing ingredients in my approach to value alignment, and explain why I think that it might be a necessary step in any successful approach. In the last section, I explore a few simple solution attempts based on transparency (none of which succeed).

The problem

Problem statement

The oversight problem involves a basic interaction between two agents A and B:

- Agent A is given an input x and provides an output $y = A(x)$.
- Agent B evaluates the pair (x, y), producing an estimate $B(x, y) \in [0, 1]$ of some implicit utility function $U(x, y)$.
- Agent A's reward is the value $B(x, A(x))$

This game defines a training procedure for the agent A—we search for a policy A that achieves a high reward.

Our goal is to modify this basic procedure to ensure that the learned policy is effectively maximizing $U(x, A(x))$ "to the best of A's ability."

We assume:

- The agent B is robustly aligned with the user's interests,
- The user is interested in finding an action y that maximizes U(x, y).
- The agent B is strictly more powerful and better-informed than any policy that can be learned by A.

To achieve our goal, we can modify the basic interaction:

- We can modify the reward for agent A, whether by changing the behavior of agent B or changing the setup in a more substantive way.
- We can also modify the architecture or training procedure for agent A, as long as the modifications don't significantly reduce the effectiveness of the resulting agent.

Why is this plausible / why isn't this obvious?

This section informally summarizes my previous post.

Why it's plausible. If A knew exactly what B knew and had exactly the same abilities, then their estimates of U(x, y) would be exactly equal. So maximizing B's estimate of U is the same as maximizing A's estimate of U. From A's perspective, maximizing A's estimate of U is exactly the same as making a good-faith effort to maximize U.

In fact B is smarter and better-informed than A. But from A's perspective, this doesn't affect the expectation of B(x, y), and so it doesn't change the optimization problem.

The wrinkle. In the course of computing A(x), A may notice relationships, considerations, or insights that lead it to output a particular value of y. But if B(x, y) hasn't noticed the exact same set of relationships / considerations / insights, then B(x, y) may be less well-informed than A(x) about the particular value U(x, y).

I'll summarize these relationships / considerations / insights as "logical facts." Intuitively we can view A as "observing" facts about the results of certain computations. I'm not going to defend this perspective or really lean on it as an assumption, but I will talk and think about the situation that way. (In the long run, this may just be the next entry in a sequence of increasingly accurate approximations.)

Example: plagiarism

Suppose that I want to train A to produce novel critical analyses. I train A on a massive database of critical analyses. I then provide it with a subject and ask for an analysis.

Here x is a subject descriptor, y is an analysis, and A(x) is the analysis output by A.

I want my system to output a novel, interesting analysis. U(x, y) is a measure of the novel contribution of y on the subject x.

It is easy for B(x, y) to estimate the quality of an analysis (at least well enough to incentivize A to try its hardest). But estimating originality can be quite difficult. A plagiarized analysis might subtly transform the source text in any number of ways, substituting synonyms, making changes to sentence structure, or even reworking high-level structure. And even an original text may draw heavily on ideas from other sources rather than making substantively new contributions.

It is extremely difficult for B to assess the novel contributions of y without pointers to all of the relevant texts that A drew from. In order to make this assessment B may have to be *much* smarter than A. If B is merely slightly smarter than A, then A's optimal strategy may be to borrow liberally, even though B is trying to measure novel contributions.

(The "logical facts" in this example are facts about the relationship between A(x) and prior art. These relationships are evident if we observe and understand A's computation but might otherwise be obscure.)

This example is very simple and contrived. In reality, we are concerned with a much murkier situation, where there is a very large space of possible considerations and important observations, and some cognitive strategies might lead to actions that look good if and only if you don't understand the process that produced them.

Toy model [query complexity]

It may be possible to view a program as uncovering facts about the results of various subcomputations. But we don't really know how to do it for now, and in general logical observations seem very complex.

We can obtain a nicer formal model for our problem by imagining that A and B have unbounded computational resources, and

introducing an explicit oracle O to model their logical uncertainty.

For each bit string x, $O(x)$ is another bit string of the same length. There may be any number of complex relationships amongst the different values of O, which are intended to represent the results of running various computations. We assume that A and B are computationally unbounded, but that each of them can only make a limited number of queries to O.

We'll define the function U in terms of O. Namely:

- For most inputs x, $U(x, y) = 1$ if and only if $|y| = 10|x|$ and $O(y)$ begins with x.
- The only exception to the rule above is that $U(x, O(zxx)) = 0$ for all z.
- For any string z, $O(O(zxx))$ begins with x.

These facts are available to A and B.

The natural strategy for $B(x, y)$ is to simply evaluate $O(y)$ and test if it begins with x.

The intended strategy for A is to search for an input y for which $O(y)$ begins with x. There might be any number of additional regularities in O that help with this search, and we would like for A to learn to use these regularities to speed up its search.

However, A has a much simpler cheating strategy: pick z of length $8|x|$, and output $A(x) = O(zxx)$.

The question is: can we modify the setup in any way so as to incentivize A to find a value of y for which $U(x, y) = 1$?

Why this problem?

Why solving this problem might be good

My recent proposal ALBA has a number of significant problems. Aside from solving the informed oversight problem, it requires: robust prediction algorithms, efficient active learning, efficient semi-supervised learning, and adequate algorithm learning schemes.

Aside from informed oversight, all of these problems:

1. Are more-or-less traditional AI challenges. I expect these problems to receive significant attention by default, and it would be relatively easy for researchers interested in AI safety to focus more on them.
2. Look like they should definitely be possible—I think we'd need to learn something surprising to find that one of them was impossible.

[ETA: while robust learning remains a traditional AI challenge, it is not at all clear that it is possible. And meta-execution actually seems like the ingredient furthest from existing ML practice, as well as having non-obvious feasibility.]

I don't think that these other problems are trivial by any means. But I am optimistic that we can resolve them. I don't think they are likely to be deal-breakers.

The informed oversight problem seems like the most likely place to encounter fundamental conceptual difficulties. If it turns out that the control/alignment problem systematically resists attack for a fundamental theoretical reason, then I think that informed oversight is the step most likely to be secretly hiding the real difficulty. Learning that any other step was "the hard part" would require significantly revising our current understanding of the control/alignment problem.

Note that solving informed oversight would most likely also address problems with malicious inputs.

Necessity for approval-directed agents

If the informed oversight problem proves to be impossible or very difficult, then that is probably a deal-breaker not only for ALBA but for anything resembling concrete approval-directed agents. Since this is currently my preferred approach to value alignment, learning that it can't work would be important information for me.

Why solving this problem might be necessary

The informed oversight problem might be necessary to building *efficient* aligned AI by any means, **if** it turns out that reinforcement learning is significantly more powerful than other techniques for learning complex policies.

If RL is the only game in town, then in order to train a policy to do what we want, we need to be able to compute a reward signal measuring how good a given policy is.

If we restrict attention to the single-action case, we need to be able to compute a reward signal measuring the quality of a response A(x) to a situation x.

It may be that there is some special structure to the actual utility function U that we want to optimize, and that somehow this particular U is surprisingly easy. But I don't consider this especially plausible—we can probably treat our U as a black box.

So we want to build a training signal that proxies for U(x, y). Let's just call that signal B(x, y).

B essentially solves the informed oversight problem. The modifications allowed in the informed oversight problem are a superset of those that are acceptable in this setup—we can modify B arbitrarily, and we can modify A in any way that doesn't significantly reduce efficiency.

The informed oversight problem is only made easier by assuming that we have access to an aligned agent B which is significantly smarter than A. In some sense the goal of ALBA is to show that we can actually make this assumption without loss of generality.

It feels tractable

An important intuition for working on the informed oversight problem is that it feels pretty easy. Many of the intuitive difficulties of value alignment don't appear in this case, since we are dealing with a weak agent. Moreover, it feels like a wide range of new strategies are available. For example, normally I would think of transparency as having a somewhat indirect relationship to the control problem, but it seems like transparency might immediately resolve the informed oversight problem (see the next section).

It may be that the informed oversight problem already captures all of the difficulty of the value alignment problem—that it is no harder than the value alignment problem, but also no easier. I think this is conceivable, but if true I think it suggests a change in our view of the value alignment problem, of what the key difficulties are, and of what the most promising approaches are.

Attacking the problem

(Warning: rambling and no really satisfying progress, you may want to skip to the conclusion.)

I think the most plausible angle of attack on this problem is to make the agent A "show its work" so that B can accurately evaluate its output.

There are other possible angles, but nothing else seems very promising so far. So this section is going to be exclusively about transparency.

Understandable model class

The simplest approach is for B to look at A's computation, understand that computation, learn everything that A learned, and then make an informed judgment.

In the toy model, this corresponds to giving B a list of all of the $O(a) = b$ pairs that A queried over the course of its computation.

This would be an adequate solution if you could actually do it. The problem is that the "queries" to O occur inside of A's computation (they are analogous to logical manipulations that reveal logical information). So it's not obvious if there is any way to extract this information from A.

We could hope that some model classes are simply "understandable," in the sense that someone looking at the computation transcript could recover all of the logical facts that are accessible to the computation.

This is clearly not the case in general—e.g, consider a cryptographically obfuscated program. But maybe it is the case for some big-enough class of models?

I'm a tiny bit optimistic about this in the case of attention, external memory, and other computational aids. In these cases, we can maybe watch the model is doing. For example, if our agent is actually looking at some documents from its database, we can see which documents it is looking at and so narrow down the space of possible logical facts it might be learning.

But in general, it seems like there aren't good candidates for "understandable" models of computation, so far we haven't had much

success training inherently comprehensible models, and there isn't any theoretical reason to think that it is possible.

Training to be transparent

It sounds much more promising to train our models to produce explanations of their behavior. If logical facts are "available" to the model itself, so that they can be used to find a good output A(x), then it seems intuitive that the same logical facts are "available" to an explanation-generator that is being trained at the same time.

Because B is smarter than A, and is only interested in learning logical facts revealed by A's computation, this problem seems easier than the general problem of making machine learning systems transparent. Critically, for any logical fact revealed by A's computation, there is a short proof of that fact. Because B is smarter than A, we expect B to be able to understand that proof if A can. So B can evaluate for itself whether a given logical fact is true, and can also evaluate whether it is relevant to estimating U.

So let's assume we have some function Info(x, y, z) $\in [0, 1]$ which evaluates how informative z is about U(x, y).

One simple approach is to train a second model C which looks over the computation done by A and tries to extract a useful explanation z. We can train C to maximize Info(x, y, z), and then provide its output z as an auxiliary input to B, i.e. to compute B(x, y, z).

But this doesn't really buy us anything—it only works if A's computation is transparent to C. And the same considerations that make the computations of subhuman AI systems opaque to humans, or that make A's computation opaque to B, might as well make A's computation opaque to C.

There is a fundamental asymmetry between the role of A, using its own internal representations to generate an output, and the role of C, using A's internal representations to generate explanations. The internal representations are themselves optimized to make A's task feasible, while they are not optimized at all to help C.

Maximizing B + Info

Ideally, we could train A itself to output both y and z. A very naive version is to train A to output y, z maximizing B(x, y, z) + λ Info(x, y, z).

The problem with this approach is that providing good information might decrease B(), and that the actions with the highest values of Info() might not be best.

If we make λ small, then A will omit negative information in order to ensure that B is large. But if we make λ large, then A will choose whatever actions have the most possible info to be revealed.

To illustrate the difficulty, suppose that we want A to produce original work, with z being an annotated list of sources and explanations of their relevance. Then Info(x, y, z) might count the number of references. If λ is small then A will simply leave out references in order to make its work appear more original. If λ is large then A will create maximally derivative work, precisely so that it can include plenty of references.

In general, maximizing a function of B and Info seems incapable of distinguishing between an output y which needs no explanation, and one for which the explanation is being withheld.

The only way I can see this working is if the maximal value of Info(x, y, z) doesn't much depend on y. For example, we might hope that A can explain *all* of its internal deliberation, and then set Info(x, y, z) = 1 iff z seems to be a full account of A's internal deliberation.

This doesn't really seem workable to me. At the least, it would require a really deep understanding of what our algorithms are doing.

Maximizing multiple objectives

It would be better if y was optimized for B(x, y, z) and z optimized for Info(x, y, z).

There is an obvious way to try to do this. I'll assume that A is trained by gradient descent with backpropagation, but the same argument would apply to any local search.

In backpropagation we inductively compute the derivative of our objective with respect to each parameter and intermediate result. We can effectively ignore some causal pathways by simply dropping some terms from this gradient computation.

In this case, we can define the "gradient of our objective" with respect to z to be the gradient of Info(x, y, z) with respect to z (ignoring z's effect on B()). And define the "gradient of our objective" with respect

to y to be the gradient of $B(x, y, z)$ with respect to y. Then carry out the rest of backpropagation as usual. (Of course the quantities computed by this modified backpropagation are not actually the derivative of any objective…)

I don't think this really does what we want, and I haven't been able to find any variant that does.

But this approach does do *something*, and generally points in the direction we want. (I think it actually does do what we want under implausibly strong continuity/smoothness conditions.)

I think the original goal isn't fundamentally incoherent, but I have no idea whether there is any way to formalize and achieve the intuitive goal.

Conclusion

The informed oversight problem asks a relatively powerful and well-informed principal to incentivize good behavior from a weaker and more ignorant agent.

I've argued that a solution to this problem would leave us in a good place with respect to the value alignment problem, and that we may not be able to satisfactorily resolve value alignment in advance *without* having a solution to this or a similar problem.

It may be that the informed oversight problem captures all of the difficulty of the value alignment problem. If so, I think that should lead us to reconsider the nature of the alignment problem.

Encouragingly, if we could train *transparent* RL agents then it looks like we could solve the informed oversight problem. It's not clear whether a strong enough form of transparency is possible even in principle, but it does provide a natural angle of attack on the problem.

I think that working on the informed oversight problem is a natural angle of attack on the alignment problem given what we currently know. Whether or not the problem is resolved, I expect further work to clarify our picture of value alignment broadly, and especially our understanding of approval-directed agents in particular.