Alternate Alignment Ideas

Stable Pointers to Value: An Agent Embedded in Its Own Utility Function

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

(This post is largely a write-up of a conversation with Scott Garrabrant.)

Stable Pointers to Value

How do we build stable pointers to values?

As a first example, consider the wireheading problem for AIXI-like agents in the case of a fixed utility function which we know how to estimate from sense data. As discussed in Daniel Dewey's Learning What to Value and other places, if you try to implement this by putting the utility calculation in a box which rewards an AIXI-like RL agent, the agent can eventually learn to modify or remove the box, and happily does so if it can get more reward by doing so. This is because the RL agent predicts, and attempts to maximize, reward received. If it understands that it can modify the reward-giving box to get more reward, it will.

We can fix this problem by integrating the same reward box with the agent in a better way. Rather than having the RL agent learn what the output of the box will be and plan to maximize the output of the box, we use the box *directly* to evaluate possible futures, and have the agent plan to maximize that evaluation. Now, if the agent considers modifying the box, it evaluates that future *with the current box*. The box as currently configured sees no advantage to such tampering. This is called an observation-utility maximizer (to contrast it with reinforcement learning). Daniel Dewey goes on to show that we can incorporate uncertainty about the utility function into observation-utility maximizers, recovering the kind of "learning what is being rewarded" that RL agents were supposed to provide, but without the perverse incentive to try and make the utility turn out to be something easy to maximize.

This feels much like a use/mention distinction. The RL agent is maximizing "the function in the utility module", whereas the observation-utility agent (OU agent) is maximizing the function in the utility module.

The Easy vs Hard Problem

I'll call the problem which OU agents solve *the easy problem of wireheading*. There's also *the hard problem of wireheading*: how do you build a stable pointer to values if you **can't** build an observation-utility box? For example, how do you set things up so that the agent wants to satisfy a human, without incentivising the AI to manipulate the human to be easy to satisfy, or creating other problems in the attempt to avoid this? Daniel Dewey's approach of incorporating uncertainty about the utility function into the utility box doesn't seem to cut it -- or at least, it's not obvious how to set up that uncertainty in the right way.

The hard problem is the wireheading problem Which Tom Everitt attempts to make progress on in Avoiding Wireheading with Value Reinforcement Learning and Reinforcement Learning with a Corrupted Reinforcement Channel. It's also connected to the problem of Generalizable Environmental Goals in AAMLS. CIRL gets at an aspect of this problem as well, showing how it can be solved if the problem of environmental goals is solved (and if we can assume that humans are perfectly rational, or that we can somehow factor out their irrationality -- Stuart Armstrong has some useful thoughts on why this is difficult). Approval-directed agents can be seen as an attempt to turn the hard problem into the easy problem, by treating humans as the evaluation box rather than trying to infer what the human wants.

All these approaches have different advantages and disadvantages, and the point of this post isn't to evaluate them. My point is more to convey the overall picture which seems to connect them. In a sense, the hard problem is just an extension of the same use/mention distinction which came up with the easy problem. We have some idea how to maximize "human values", but we don't know how to actually maximize human values. Metaphorically, we're trying to dereference the pointer.

Stuart Armstrong's indifference work is a good illustration of what's hard about the hard problem. In the RL vs OU case, you're going to constantly struggle with the RL agent's misaligned incentives until you switch to an OU agent. You can try to patch things by explicitly punishing manipulation of the reward signal, warping the agent's beliefs to think manipulation of the rewards is impossible, etc, but this is really the wrong approach. Switching to OU makes all of that unnecessary. Unfortunately, in the case of the hard problem, it's not clear there's an analogous move which makes all the slippery problems disappear.

Illustration: An Agent Embedded in Its Own Utility Function

If an agent is logically uncertain of its own utility function, the easy problem can turn into the hard problem.

It's quite possible that an agent might be logically uncertain of its own utility function if the function is quite difficult to compute. In particular, human judgement could be difficult to compute even after learning all the details of the human's preferences, so that the AI needs uncertain reasoning about what the model tells it.

Why can this turn the easy problem of wireheading into the hard problem? If the agent is logically uncertain about the utility function, its decisions may have logical correlations with the utility function. This can give the agent some logical control over its utility function, reintroducing a wireheading problem.

As a concrete example, suppose that we have constructed an AI which maximizes CEV: it wants to do what an imaginary version of human society, deliberating under ideal conditions, would decide is best. Obviously, the AI cannot actually simulate such an ideal society. Instead, the AI does its best to reason about what such an ideal society would do.

Now, suppose the agent figures out that there would be an exact copy of itself inside the ideal society. Perhaps the ideal society figures out that it has been constructed as a thought experiment to make decisions about the real world, so they construct a simulation of the real world in order to better understand what they will be making decisions about. Furthermore, suppose for the sake of argument that our AI can break out of the simulation and exert arbitrary control over the ideal society's decisions.

Naively, it seems like what the AI will do in this situation is take control over the ideal society's deliberation, and make the CEV values as easy to satisfy as possible -- just like an RL agent modifying its utility module.

Obviously, this could be taken as reason to make sure the ideal society doesn't figure out that it's just a thought experiment, or that they don't construct copies of the AI. But, we don't generally want good properties of the AI to rely on assumptions about what humans do; wherever possible, we want to design the AI to avoid such problems.

Indifference and CDT

In this case, it seems like the right thing to do is for the AI to ignore any influence which its actions have on its estimate of its utility function. It should act as if it only has influence over the real world. That way, the ideal society which defines CEV can build all the copies of the AI they want; the AI only considers how its actions have influence over the real world. It avoids corrupting the CEV.

Clearly, this would be an indifference-style solution. What's interesting to me is that it also looks like a CDT-style solution. In fact, this seems like an answer to my question at the end of Smoking Lesion Steelman: a case of ignorance about your own utility function which doesn't arise from an obviously bad agent design. Like the smoking lesion steelman, ignorance about utility here seems to recommend CDT-like reasoning over EDT-like reasoning.

This suggests to me that there is a deep connection between CDT, indifference, stable pointers, and corrigibility. As Jason Konek & Ben Levinstein argued in The Foundations of Epistemic Decision Theory, CDT is about getting the direction-of-fit right in decision theory: you want beliefs to better fit the world (if your beliefs don't match the world, you change your beliefs), but you want the world to better fit your goals (if your goals don't match the world, you change the world). The easy problem of wireheading is to follow the second maxim when you have your utility function in hand. The hard problem of wireheading is to go about this when your utility is not directly observable. If you build a stable pointer to a human, you become corrigible. Doing this correctly seems to involve something which at least looks very similar to indifference.

This picture is a little too clean, and likely badly wrong in some respect: several of these concepts are likely to come apart when examined more closely. Nonetheless, this seems like an interesting way of looking at things.

Stable Pointers to Value II: Environmental Goals

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

Cross-posted.

In Stable Pointers to Value, I discussed various ways in which we can try to “robustly point at what we want” (ie, do value learning). I can tidy up the discussion there into three categories:

1. Standard reinforcement learning (RL) frameworks, including AIXI, which try to predict the reward they’ll get and take actions which maximize that expectation. These are incentivised to hack whatever system is feeding them rewards. This is the “easy problem of wireheading”.
2. Observation-utility agents (OU), which get around the problem by assessing the future plans *with the current utility function subsystem* rather than trying to predict what the subsystem will say. This removes the incentive to manipulate what the subsystem will say. The subsystem can itself have uncertainty about what the “true” utility function is; for example, representing uncertainty about what the human wants. CIRL falls into this category. There are a number of problems which can arise here depending on the setup; see for example basically everything Stuart Armstrong has ever written. In particular, you’ve got to define “how much the human likes the plan” somehow, and this is usually manipulable, often to the point of wireheading. I called this the “hard problem of wireheading”.
3. Approval-directed agents (AD) maximize human approval of individual actions, rather than planning to maximize overall approval. Just as observation-utility agents get around the easy wireheading problem by putting the reward function inside the optimization loop, this gets around the hard wireheading problem by putting humans in the loop. Other problems are introduced, though; most prominently, (a) how do we trust the learned model of human evaluation itself, and (b) how do we make this kind of system competitive with systems which plan ahead, given the obvious advantages of planning ahead? Paul Christiano has thoughts on these problems in various posts.

I want to point at an analogy to three categories of approach to the problem of generalizable environmental goals (as defined in the alignment for advanced machine learning agenda). It’s a fairly messy analogy, and there’s probably a better way of organizing the landscape, but FWIW.

1. Supervised Learning

Imagine you’re trying to teach a system to build bridges by showing it examples. You could learn a big neural network which distinguishes cases of “successfully building a bridge” from everything else, and then use this to drive the system.

If the agent is an RL or OU agent, it is incentivised to “fool itself” by doing things like playing a video of bridge-building in front of its camera. You can try and train the classifier to notice this sort of thing, of course; you give it negative training examples in which someone puts a TV set in front of it and things thereafter appear as they do in one of the positive examples. However, you can’t figure out all the different negative training examples you need to give it ahead of time – especially if the rest of the system will continue to learn later on as the classifier remains fixed.

To me, this feels closely analogous to trying to prevent RL systems from wireheading themselves by giving them strongly negative reward for trying to mess with their reward circuits. You don’t know ahead of time what all the things you need to punish are, but you would need to, since the system keeps getting smarter as the reward circuit remains the same. (Or, if humans are managing the reward button, they need to be able to recognize any attempts to mess with the hardware or take over control of the reward button or manipulate the humans.)

2. Model-Utility Learning

One way you might try to solve this: the AI is learning a model of the world in an unsupervised way, only trying to predict well, not thinking at all about its goals. Separately, the AI is learning a classifier representing the goals. This classifier takes the *model* state, rather than the observations.

So, returning to the bridge-building example, the system is shown lots of examples of building bridges and not building bridges. It infers a physical model of what’s going on in those examples, plus a predicate on the physical situations which tells it whether the state of affairs corresponds to proper bridge-building.

As before, we can show it many negative training examples involving different methods of attempting to fool itself.

Now, we might reasonably expect that if the AI considers a novel way of “fooling itself” which hasn’t been given in a training example, it will reject such things for the right reasons: the plan does not involve physically building a bridge.

This can also deal with the problem of ontological crisis, even without new classifier data. As the physical model changes in response to new data, the classifier is simply re-learned so that it remains accurate on the original training examples.

Unfortunately, this approach has serious problems.

Since humans (or *something*) must be labeling the original training examples, the hypothesis that building bridges means “what humans label as building bridges” will always be at least as accurate as the intended classifier. I don’t mean “whatever humans *would* label”. I mean they hypothesis that “build a bridge” means specifically the physical situations which were recorded as training examples for this system in particular, and labeled by humans as such.

This time, there’s no way to patch the problem with negative training examples. You can’t label an example as both positive and negative!

How can we avoid simple-but-wrong hypotheses like this?

3. Human Hypothesis Evaluation

Just as approval-directed agents put more work on the humans in the control loop, we can try and do the same here.

As in model-utility systems, we build a model of the environment through unsupervised learning, and also try to learn the utility in a supervised way.

However, this time the system gets feedback on the quality of hypotheses from humans, and also tries to anticipate such feedback in its model selection. I'm not sure exactly how this should work, but one version is: ask the humans to classify made-up examples. Such examples of bridge-building can be in imaginary worlds where there are no humans evaluating whether bridge-building is going on, so as to differentiate the pathological hypothesis mentioned above from the desired hypothesis.

For this to work, though, we also have to solve the problem of providing human-understandable explanations of the AI's learned models, which is its own pandora's box.

Discussion

The overall point I'm trying to make here has similarities to the Reinforcement Learning with a Corrupted Reward Channelpaper, particularly section 4.1: the way the system gets feedback matters a lot. The way humans get put into the loop can be very tricky; seemingly obvious answers lead to pathological behaviors for highly capable systems. Trying to fix this behavior can lead us down a rabbit-hole of trying patch after patch after patch, until a change in perspective like observation-utility learning eliminates the need for all those patches in one fell swoop (and then we find ourselves making entirely new patches on a higher level and about more important things…).

Stable Pointers to Value III: Recursive Quantilization

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This is a very loose idea.

In Stable Pointers to Value II, I pointed at a loose hierarchy of approaches in which you try to get rid of the wireheading problem by revising the feedback loop to remove the incentive to wirehead. Each revision seems to change the nature of the problem (perhaps to the point where we don't want to call it a wireheading problem, and instead would put it in a more general perverse-instantiation category), but not eliminate problems entirely.

Talking with Lawrence Chan today, he described a way of solving problems by "going meta" (a strategy which he was mostly suspicious of, in the conversation). His example was: you can't extract human values by specifying it as a learning problem, because of severe identifiability problems. *However*, it is not entirely implausible that we *can* "learn to learn human values": have humans label examples of other humans trying to do things, indicating what values are being expressed in the scenario.

If *this* goes wrong, you can try and iterate the operation again, learning to learn to learn...

This struck me as similar to the hierarchy I had constructed in my older post.

My interpretation of what Lawrence meant by "going meta" here is this: machine learning research "eats" other research fields by using automated learning to solve problems which were previously being solved by the process of science, IE, hand-crafting hypotheses and testing them. AI alignment research is full of cases where this doesn't seem like a very good approach. However, one attitude we can take to such cases is to *do the operation again:* propose to learn how humans would solve this sticky problem.

This is not at all like other learning to learn approaches which merely seek to *speed up* normal learning. The idea is that our object-level loss function is insufficient to point out the behavior we really want. We want *new normative feedback to come in at the meta-level,* telling us more about which ways of solving the object-level problem are desirable and which are undesirable.

The idea I'm about to describe seems like a fairly hopeless idea, but I'm interested in seeing how it would go regardless.

What is the fixed point of this particular "go meta" operation?

The intuition is this: any utility function we try to write down has perverse instantiations, so that we don't really want to optimize it fully. Searching over a big space leads to Goodhart and optimization daemons. Unfortunately, search is more or less the only way to produce intelligent behavior that we know of.

However, it seems like we can often improve on this situation by providing more human input to check what was really wanted. Furthermore, it seems like we generally get more by doing this on the meta level -- we don't just want to refine the estimated utility function; we want to refine our notion of safely searching for good options (avoiding searches which goodhart on looks-good-to-humans by manipulating human psychology, for example), refine our notion of what learning the utility function even means, and so on.

Every stage of going meta introduces a need for yet another search, which brings back the problems all over again. But, maybe we can do something interesting by jumping up *all* the meta levels here, so that each search is itself governed by some feedback, except when we bottom out in extremely simple operations which we trust.

(This feels conceptually similar to some intuitions in HCH/IDA, but I don't see that it is exactly the same.)

Recursive Quantilization

"Recursive quantilization" is an attempt to make the idea a little more formal. I don't think it quite captures everything I would want from the "fixed point of the meta operation Lawrence Chan was suspicious of", but it has the advantage of being slightly more concrete.

Quantilizers are a way of optimizing a utility function when you're suspicious that it isn't the "true" utility function you should be optimizing, but you *do* think that the average difference is low when sampling things from a known background distribution. Intuitively, you don't want to move *too* far from the background distribution where your utility estimates are accurate, but you do want to optimize in the direction of high utility *somewhat*.

What if we want to quantilize, and we expect that there is *some* background distribution which would make us have a decent amount of trust in the accuracy of the given utility function, but we don't know what that background distribution is?

We have to learn the "safe" background distribution.

Learning is going to require a search for hypotheses matching whatever feedback we get, which re-introduces Goodhart, etc. So, we quantilize *that* search. But we need a background distribution which we expect to be safe. And so on.

- You start with very broad priors on what background distributions might be safe, so you barely optimize at all, but have some default (human-programmed) strategy of asking humans questions.
- You engage in active learning, steering toward questions which resolve the most important ambiguities (to the extent that you're willing to steer).
- Because we are taking "all the meta levels" here, we can do some amount of generalization across meta levels so that the stack of meta doesn't get out of hand. In other words, we're actually learning one safe background distribution for all the meta levels, which encodes something like a human concept of "non-fishy" ways of going about things.

Issues

There are a lot of potential concerns here, but the one which is most salient to me is that humans will have a lot of trouble providing feedback about non-fishy ways of solving the problems at even slightly high meta levels.

Object level: Plans for achieving high utility.

Meta 1: Distributions containing only plans which the utility function evaluates correctly.

Meta 2: Distributions containing only distributions-on-plans which the first-meta-level learning algorithm can be expected to evaluate correctly.

Et cetera.

How do you analyze a distribution? Presumably you have to get a good picture of its shape in the highly multidimensional space -- look at examples of more and less typical members, and be convinced that the examples you looked at were representative. It's also important that you go into its code and check that it isn't intelligently optimizing for some misaligned goal.

It seems to me that a massive advance in transparency or informed oversight would be needed in order for humans to give helpful feedback at higher meta-levels.

Policy Alignment

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

(ETA: The name "policy approval" wasn't great. I think I will use the term "policy alignment" to contrast with "value alignment" going forward, at the suggestion of Wei Dai in the comments.)

I recently had a conversation with Stuart Armstrong in which I claimed that an agent which learns your utility function (pretending for a moment that "your utility function" really is a well-defined thing) and attempts to optimize it is still not perfectly aligned with you. He challenged me to write up specific examples to back up my claims.

I'll also give a very sketchy alternative to value learning, which I call policy alignment. (The policy alignment idea emerged out of a conversation with Andrew Critch.)

Background

Stuart Armstrong has recently been doing work showing the difficulty of inferring human values. To summarize: because humans are irrational, a value-learning approach like CIRL needs to jointly estimate the human utility function and the degree to which the human is rational -- otherwise, it would take all the mistakes humans make to be preferences. Unfortunately, this leads to a severe problem of identifiability: humans can be assigned any values whatsoever if we assume the right kind of irrationality, and the usual trick of preferring simpler hypotheses doesn't seem to help in this case.

I also want to point out that a similar problem arises even without irrationality. Vladimir Nesov explored how probability and utility can be mixed into each other without changing any decisions an agent makes. So, in principle, we can't determine the utility or probability function of an agent uniquely based on the agent's behavior alone (even including hypothetical behavior in counterfactual situations). This fact was discovered earlier by Jeffrey and Bolker, and is analyzed in more detail in the book *The Logic of Decision*. For this reason, I call the transform "Jeffrey-Bolker rotation".

To give an illustrative example: it doesn't matter whether we assign very low probability to an event, or care very little about what happens given that event. Suppose a love-maximizing agent is unable to assign nonzero utility to a universe where love isn't real. The agent may appear to ignore evidence that love isn't real. We can interpret this as not caring what happens conditioned on love not being real; or, equally valid (in terms of the actions which the agent chooses), we can interpret the agent as having an extremely low prior probability on love not being real.

At MIRI, we sometimes use the term "probutility" to indicate the *probability,utility* pair in a way which reminds us that they can't be disentangled from one another. Jeffrey-Bolker rotation changes probabilities and utilities, but does not change the overall probutilities.

Given these problems, it would be nice if we did not actually need to learn the human utility function. I'll advocate that position.

My understanding is that Stuart Armstrong is optimistic that human values can be inferred despite these problems, because we have a lot of useful prior information we can take advantage of.

It is intuitive that a CIRL-like agent should learn what is irrational and then "throw it out", IE, de-noise human preferences by looking only at what we really prefer, not at what we mistakenly do out of short-sightedness or other mistakes. On the other hand, it is not so obvious that the probability/utility distinction should be handled in the same way. Should an agent disentangle beliefs from preferences just so that it can throw out human beliefs and optimize the preferences alone? I argue against this here.

Main Claim

Ignoring issues of irrationality or bounded rationality, what an agent wants out of a helper agent is that the helper agent does preferred things.

Suppose a robot is trying to help a perfectly rational human. The human has probability function P_H and utility function U_H. The robot is in epistemic state ***e***. The robot has a set of actions $a_1, a_2, ..., a_n$. The proposition "the robot takes the ***i***th action when in epistemic state ***e***" is written as $R(e) = a_i$. The set of full world-states is ***S***. What the human would like the robot to do is given by:

$$R(e) = \text{argmax}_{a_i}\{\sum_{s \in S} P_H(s|R(e) = a_i)U_H(s)\}$$

(Or by the analogous causal counterfactual, if the human thinks that way.)

This notion of what the human wants is invariant to Jeffrey-Bolker rotation; the robot doesn't need to disentangle probability and utility! It only needs to learn probutilities.

The equation written above can't be directly optimized, since the robot doesn't have direct access to human probutilities. However, I'll broadly call any attempt to approximate that equation "policy alignment".

Notice that this is closely analogous to UDT. UDT solves dynamic inconsistencies -- situations in which an AI could predictably dislike the decisions of its future self -- by optimizing its actions from the perspective of a fixed prior, IE, its initial self. Policy alignment resolves inconsistencies between the AI and the human by optimizing the AI's actions from the human's perspective. The main point of this post is that we can use this analogy to produce counterexamples to the typical value-learning approach, in which the AI tries to optimize human utility but not according to human beliefs.

I will somewhat ignore the distinction between UDT1.0 and UDT1.1.

Examples

These examples serve to illustrate that "optimizing human utility according to AI beliefs" is not exactly the same as "do what the human would want you to do", even when we suppose "the human utility function" is perfectly well-defined and can be learned exactly by the AI.

In these examples, I will suppose that the AI has its own probability distribution P_R. It reasons updatelessly with respect to evidence ***e*** it sees, but with full prior knowledge of the human utility function:

$$R(e) = \text{argmax}_{a_i} \{\sum_{s \in S} P_R(s|R(e) = a_i) U_H(s)\}$$

I use an updateless agent to avoid accusations that *of course* an updateful agent would fail classic UDT problems. However, it is not really very important for the examples.

I assume prior knowledge of U_H to avoid any tricky issues which might arise by attempting to combine updatelessness with value learning.

Counterfactual Mugging

It seems reasonable to suppose that the AI will start out with some mathematical knowledge. Imagine that the AI has a database of theorems in memory when it boots up, including the first million digits of pi. Treat these as part of the agent's prior.

Suppose, on the other hand, that the human which the AI wants to help does not know more than a hundred digits of pi.

The human and the AI will disagree on what to do about counterfactual mugging with a logical coin involving digits of pi which the AI knows and the human does not. If Omega approaches the AI, the AI will refuse to participate, but the human will wish the AI would. If Omega approaches the human, the AI may try to prevent the human from participating, to the extent that it can do so without violating other aspects of the human utility function.

"Too Updateless"

Maybe the problem with the counterfactual mugging example is that it doesn't make sense to program the AI with a bunch of knowledge in its prior which the human doesn't have.

We can go in the opposite extreme, and make P_R a broad prior such as the Solomonoff distribution, with no information about our world in particular.

I believe the observation has been made before that running UDT on such a prior could have weird results. There could be a world with higher prior probability than ours, inhabited by Omegas who ask the AI to optimize alien values in most universes (including Earth) in exchange for the Omegas maximizing U_h in their own world. (This particular scenario doesn't seem particularly probable, but it *does* seem quite plausible that *some* weird universes will have higher probability than our universe in the Solomonoff prior, and may make some such bargain.)

Again, this is something which can happen in the maximization using P_R but not in the one using P_H -- unless humans themselves would approve of the multiversal bargain.

"Just Having a Very Different Prior"

Maybe P_R is neither strictly more knowledgable than P_H nor less, but the two are very different on some specific issues. Perhaps there's a specific plan p which, when P_R is conditioned on evidence so far, looks very likely to have many good consequences. P_H considers the plan very likely to have many bad consequences. Also suppose that there aren't any interesting consequences of this plan in counterfactual branches, so UDT considerations don't come in.

Also, suppose that there isn't time to test the differing hypotheses involved which make humans think this is such a bad plan while AIs think it is so good. The AI has to decide right now whether to enact the plan.

The value-learning agent will implement this plan, since it seems good on net for human values. The policy-alignment agent will not, since humans wouldn't want it to.

Obviously, one might question whether it is reasonable to assume that things got to a point where there was such a large difference of opinion between the AI and the humans, and no time to resolve it. Arguably, there should be safeguards against this scenario which the value-learning AI itself would want to set up, due to facts about human values such as "the humans want to be involved in big decisions about their future" or the like.

Nonetheless, faced with this situation, it seems like policy-alignment agents do the right thing while value-learning agents do not.

Issues/Objections

Aren't human beliefs bad?

Isn't it problematic to optimize via human beliefs, since human beliefs are low-quality?

I think this is somewhat true and somewhat not.

- Partly, this is like saying "isn't UDT bad because it doesn't learn?" -- actually, UDT acts as if it updates most of the time, so it is wrong to think of it as incapable of learning. Similarly, although the policy-alignment agent uses P_H, it will mostly act as if it has updated P_H on a lot of information. So, maybe you believe human beliefs aren't very good -- but do you think we're capable of learning almost anything eventually? If so, this may address a large component of the concern. In particular, if you trust the output of certain machine learning algorithms more than you trust yourself, the AI can run those algorithms and use their output.

- On the other hand, humans probably have incoherent P_H, and not *just* because of logical uncertainty. So, the AI still needs to figure out what is "irrational" and what is "real" in P_H, just like value-learning needs to do for U_H.

If humans would want an AI to optimize via human beliefs, won't that be reflected in the human utility function?

Or: If policy-alignment were good, wouldn't a value-learner self modify into policy-alignment anyway?

I don't think this is true, but I'm not sure. Certainly there could be simple agents who value-learners cooperate with without ever deciding to self-modify into policy-alignment agents. Perhaps there is something about human preference which desires the AI to cooperate with the human even when the AI thinks this is (otherwise) net-negative for human values.

Aren't I ignoring the fact that the AI needs its own beliefs?

In "Just Having a Very Different Prior", I claimed that if P_R and P_H disagree about the consequences of a plan, value-learning can do something humans strongly don't want it to do, whereas policy-alignment cannot. However, my definition of policy-alignment ignores learning. Realistically, the policy-alignment agent needs to also have beliefs P_R, which it uses to approximate the human approval of its actions. Can't the same large disagreement emerge from this?

I think the concern is qualitatively less, because the policy-alignment agent uses P_R only to estimate P_H and U_H. If the AI *knows* that humans would have a large disagreement with the plan, the policy-alignment agent would not implement the plan, while the value-learning agent would.

For policy-alignment to go wrong, it needs to have a bad estimate of P_H and U_H.

The policy is too big.

Even if the process of learning P_H is doing the work to turn it into a coherent probability distribution (removing irrationality and making things well-defined), it still may not be able to conceive of important possibilities. The evidence which the AI uses to decide how to act, e in the equations given earlier, may be a large data stream with some human-incomprehensible parts.

As a result, it seems like the AI needs to optimize over compact/abstract representations of its policy, similarly to how policy selection in logical inductors works.

This isn't an entirely satisfactory answer, since (1) the representation of a policy as a computer program could still escape human understanding, and (2) it is unclear what it means to correctly represent the policy in a human-understandable way.

Terminology

[Aside from issues with the approach, my term "policy approval" may be terrible. It sounds too much like "approval-directed agent", which means something different. I think there are similarities, but they aren't strong enough to justify referring to both as "approval". Any suggestions?]

[Now using "Policy Alignment" for this. Editing post accordingly.]

Advantages

(These are very speculative.)

Logical Updatelessness?

One of the major obstacles to progress in decision theory right now is that we don't know of a good updateless perspective for logical uncertainty. Maybe a policy-alignment agent doesn't need to solve this problem, since it tries to optimize from the human perspective rather than its own. Roughly: logical updatelessness is hard because it tends to fall into the "too updateless" issue above. So, maybe it can be a non-issue in the right formulation of policy alignment.

Corrigibility?

Stuart Armstrong is somewhat pessimistic about corrigibility. Perhaps there is something which can be done in policy-alignment land which can't be done otherwise. The "Just Having Very Different Priors" example points in this direction; it is an example where policy-alignment acts in a much more corrigible way.

A value-learning agent can always resist humans if it is highly confidant that its plan is a good one which humans are opposing irrationally. A policy-alignment agent can think its plan is a good one but also think that humans would prefer it to be corrigible on principle regardless of that.

On the other hand, a policy-alignment agent isn't *guaranteed* to think that. Perhaps policy-alignment learning can be specified with some kind of highly corrigible bias, so that it requires a lot of evidence to decide that humans don't want it to behave corrigibly in a particular case?

Conclusion

I've left out some speculation about what policy-alignment agents should actually look like, for the sake of keeping mostly to the point (the discussion with Stuart). I like this idea because it involves a change in perspective of what an agent should be, similar to the change which UDT itself made.

Non-Consequentialist Cooperation?

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This is a very rough intuition pump for possible alternatives to value learning.

In broad strokes, the goal of (ambitious) value learning is to define and implement a notion of cooperation (or helpfulness) in terms of two activities: (1) figuring out what humans value, (2) working to optimize that.

I'm going to try to sketch an alternative notion of cooperation/helpfulness. This intuition is based on libertarian or anarcho-capitalist ideas, but in some ways seems closer to what humans do when they try to help.

Autonomy

I was talking to Andrew Forrester about the suffering golem thought experiment. I'm not sure who came up with this thought experiment, but the idea is:

Suffering Golem: *A golem suffers terribly in every moment of its existence, but it says it wants to keep living. Do you kill it?*

The idea is that if you think it is good to kill it, you're a hedonic utilitarian: your altruistic motives have to do with maximizing pleasure / minimizing suffering. If you think it should not be killed, then you're more likely to be a preference utilitarian: your altruistic motives have to do with what the person values for themselves, rather than some other thing you think would be good for them. (I tend to lean toward preference utilitarianism myself, but don't think the question is obvious.)

Andrew Forrester was against killing it, but justified his answer with something like "it's none of your business. You could provide convenient means of suicide if you wanted..." This wasn't an expression of not caring about the welfare of the golem. Rather, it was a way of saying that you want to preserve the golem's autonomy.

I realized that although his answer was consistent with preference utilitarianism on the surface, it went beyond. I think he would likely have a similar response to the following thought experiment:

Confused Golem: *A golem hates every moment of its existence, and would prefer to die, but it is unable to admit the fact to itself. It thinks that it loves life and wants to continue living. Perhaps it could eventually realize that it preferred not to exist if it thought about the question for long enough, but that day is a long way off. Do you kill it?*

The autonomy-preserving move is to not kill the confused golem. You might talk to the golem about what it wants, but you wouldn't actively optimize for convincing it that it actually wants to die. (That would subtract from its autonomy.)

Informed Consent?

If I imagine something which is only motivated to help, where "help" is interpreted in the autonomy-centric way, it seems like the idea is an entity which only acts on your informed consent. It will sit and do nothing until it is confident that there is something which you want it to do, and understand the consequences of it doing.

Imagine you buy a robot which runs on these rules. The robot sits patiently in your house. Sitting there is not a violation of its directive, because it did not place itself there; whatever the consequences may be, they are a result of your autonomous action. However, it does watch you, which may violate informed consent. It has a high prior probability that you understand it is watching and consent to this, because the packaging had prominent warnings about this. Watching you is necessary for the robot to function, since it must infer your consent. It may shut off if it infers that you do not understand this.

The robot will continue doing nothing until it has gained confidence that you have fully read the instruction booklet, which contains the basic facts about how the robot functions. You may then issue commands to the robot. The instruction booklet recommends that, before you try this, you say "I consent to discuss the meaning of my commands with you, the robot." This clears the robot to ask clarifying questions about commands and to tell you about the likely consequences of commands if it does not think you understand them. Without giving consent to this, the robot will often fail to do anything without offering any explanation for its failure.

Another recommended command is "Let's discuss how we can work together." This clears the robot to make general inquiries about how it should behave toward you. Once you issue this command, for the duration of the conversation, the robot will formulate intelligent questions about what you might want, what you like and dislike, where you struggle in your life, and so on. It will make suggestions about how it can help, many invented on the spot. At some point during the conversation it will likely ask if it should maintain this level of candor going forward, or if it should only discuss its tasks in such an open-ended way upon request.

Gentle Value Extraction

What the robot will *absolutely not* do during this initial interview is pry into your personal life with questions optimized to extract the maximum useful information about your values and life difficulties. Although that might be the most *useful* thing it could do during its initial interview with you, it would break your autonomy, because many humans are uncomfortable discussing certain topics, and breaking these norms is not a reasonable consequence to expect from the command you've issued. Since humans may not even wish to consent to the robot *knowing* various personal details (and may accidentally reveal enough information for the robot to figure things out), the robot has to tread lightly in its inferences, too. Even asking directly whether a certain topic is OK may be an unwanted and unexpected act, making it impossible to go there unless the human brings it up on their own initiative.

The robot might not even try to gently move the discussion in the direction of greater openness about private details, because "trying to get the human to open up more" is not an obvious consequence of discussing potential tasks. But it isn't obvious; maybe trying to get people to open up is normal enough for a conversation that this is fine. The instruction booklet could warn users about it, making it an expected consequence and therefore part of what is consented to.

Explicit Consent vs Inferred Consent

At this point, someone might be thinking "Why are you talking about the robot inferring that the human consents to certain things as reasonable expectations of giving certain commands? Why give so much leeway? We could just require explicit consent instead."

Explicit consent is so impractical as to border on meaninglessness. We want the robot to have some autonomy in how it executes commands. If it knows we like cream in our coffee, it makes sense for it to just put the cream in without asking every time, or us issuing a general rule. Cream in the coffee is a reasonable expectation. The way I think about it, an explicit consent requirement would force us to approve every motor command precisely; the freedom to intelligently carry out complex tasks in response to commands *requires* a certain amount of freedom to infer consent.

Another way of thinking about the problem is that explicit consent places dictionary-definition English in too much of a special position. We can convey our meaning in any number of ways. In a sufficiently information-rich context, a glance might be sufficient.

Turning things the other way around, there are also cases when explicit consent doesn't make for inferred consent. If someone is made to consent under duress, consent should not be inferred.

The biggest argument I see in *favor* of explicit consent is that it makes for a much lower risk of misunderstanding. Misunderstanding is certainly a serious concern, and one reason why humans often require explicit consent in high-stakes situations. However, in the context of consent-based robotics, there are likely better ways of addressing the concern:

- Requiring higher confidence in inferred consent. This might be modulated by the inferred importance of the question in a situation, so that explicit consent is required in practice for anything of importance, due to the high confidence it establishes. Measuring "importance" in this way creates its own potential safety concerns, of course.
- Using highly robust machine-learning techniques, so that spuriously high confidence in inferred consent is very unlikely.

What Does Informed Consent Mean?

There's a conceptual problem here which feels very similar to impact measures. An impact measure is supposed to quantify the extent to which an action changes things in general. Informed consent seems to require that we quantify the degree to which a change fits within certain expectations. The notion of "change" seems to be common between the two.

For example, at least according to my intuition, an impact measure should not penalize an AI much for the butterfly-effect changes in weather patterns which any action implies. The future will include hurricanes destroying cities in a broad variety of circumstances, and small actions may create large changes in which hurricanes / which cities. If a particular action foreseeably changes the *overall impact* of the hurricane/city pattern on *other* important variables in a significant way, *then* an impact measure should penalize it.

Similarly, a human can have informed consent as to the consequences of a robot going to the grocery store and buying bananas without understanding all the consequences on future weather patterns, even though this will involve some large changes to which hurricanes destroy which cities at some point later. On the other hand, if the robot walks to the grocery store in just the right way so as to cause a series of severe hurricanes to tip the right dominoes to cause a severe economic collapse which would otherwise not have happened, then this is a significant unexpected consequence of going to the grocery store which the human would need to consent to separately.

Human Rationality Assumptions

The bad news is that this approach seems likely to run into essentially all the same conceptual difficulties as value learning, if not more. Although the conceptual framework is not as strongly tied to VNM-style utility functions as value learning is, the robot still needs to infer what the human believes and wants: belief for the "informed" part, and want for the "consent" part. This still sounds like it is most naturally formulated in a VNM-ish framework, although there may be other ways.

As such, it doesn't seem like it helps any with the difficulties of assuming human rationality.

Helping Animals

My friend mentioned that the suffering golem scenario depends a great deal on whether the golem is sentient. Mercy-killing suffering animals is OK, even good, without any consent. More generally, there are lots of acceptable ways of helping animals which break their autonomy in significant ways, such as taking them to the vet against protests. One might say the same thing of children.

It isn't obvious what makes the difference, but one idea might be: where there is no *capacity* for informed consent, other principles may apply. But what would this imply for humans? There may be consequences of actions which we lack the capacity to understand. Should the robot simply try to optimize for our preferences on those issues, without constraining acceptable consequences by consent?

How should an autonomy-respecting robot interact with children? Respecting human autonomy absolutely might make it impossible to help with certain household tasks like changing diapers. If so, the approach might not result in very capable agents.

Respecting All Humans

So far, I've focused on a thought experiment of a robot respecting the autonomy of a single designated user. Ultimately, it seems like an approach to alignment has to deal with all humans. However, getting "consent" from all humans seems impossible. How can a consent-based approach approve any actions, then?

One idea is to only require consent from humans who are impacted by an action. Any action which impacts the whole future would require consent from everyone (?), but low-impact actions could be carried out with only consent from those involved.

It's not clear to me how to approach this question.

Connections to Other Problems

- As I mentioned, this seems to connect to **impact measures**.
- The agent as described also may be a **mild optimizer**, because (1) it has to avoid thinking about things when those things are not understood consequences of carrying out commands, (2) plans are constrained by the human probability distribution over plans, somehow (I'm not sure how it works, but there's definitely an aspect of "unexpected plans are not allowed" in play here).
- There is a connection to **transparency**, in that impacts of actions have to be described/understood (and approved) in order to be allowed.
- The agent as I've described it sounds **potentially corrigible**, in that resistance to shutdown or modification would have to be an understood and approved consequence of a command.