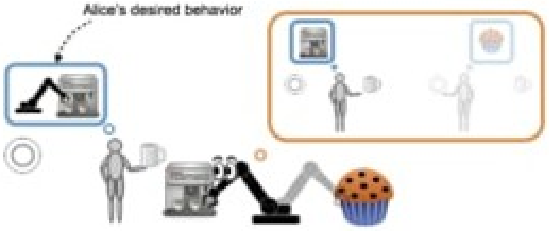


Rob observes Alice's actions to infer (and pursue) her desired goal.

Value Learning

Preface to the sequence on value learning

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

This is a meta-post about the upcoming sequence on Value Learning that will start to be published this Thursday. This preface will also be revised significantly once the second half of the sequence is fully written.

Purpose of the sequence

The first part of this sequence will be about the tractability of ambitious value learning, which is the idea of inferring a utility function for an AI system to optimize based on observing human behavior. After a short break, we will (hopefully) continue with the second part, which will be about why we might want to think about techniques that infer human preferences, even if we assume we won't do ambitious value learning with such techniques.

The aim of this part of the sequence is to gather the current best public writings on the topic, and provide a unifying narrative that ties them into a cohesive whole. This makes the key ideas more discoverable and discussable, and provides a quick reference for existing researchers. It is meant to teach the ideas surrounding *one* specific approach to aligning advanced AI systems.

We'll explore the specification problem, in which we would like to define the behavior we want to see from an AI system. Ambitious value learning is one potential avenue of attack on the specification problem, that assumes a particular model of an AI system (maximizing expected utility) and a particular source of data (human behavior). We will then delve into conceptual work on ambitious value learning that has revealed obstructions to this approach. There will be pointers to current research that aims to circumvent these obstructions.

The second part of this sequence is currently being assembled, and this preface will be updated with details once it is ready.

The first half of this sequence takes you near the cutting edge of *conceptual* work on the *ambitious* value learning problem, with some pointers to work being done at this frontier. Based on the arguments in the sequence, I am confident that the obvious formulation of ambitious value learning has major, potentially insurmountable conceptual hurdles given the ways that AI systems work currently, but it may be possible to pose a different formulation that does not suffer from these issues, or to add hardcoded assumptions to the AI system to avoid impossibility results. If you try to disprove the arguments in the posts, or to create formalisms that sidestep the issues brought up, you may very well generate a new interesting direction of work that has not been considered before.

There is also a community of researchers working on inverse reinforcement learning without focusing on its application to ambitious value learning; this is out of the scope of the first half of this sequence, even though such work may still be relevant to long term safety.

Requirements for the sequence

Understanding these posts will require at least a passing familiarity with the basic principles of machine learning (*not* deep learning), such as “the parameters of a model are chosen to maximize the log probability that the model assigns to the observed dataset”. No other knowledge about value learning is required. If you do not have this background, I am not sure how easy it will be to grasp the points made; many of the points feel intuitive to me even without an ML background, but this could be because I no longer remember what it was like to not have ML intuitions.

There are many different subcultures interested in AI safety, and the posts I have chosen to include involve linguistic choices and assumptions from different places. I have tried to make this sequence understandable to all people who are interested and who understand the basic principles of ML, and so if something seems odd/confusing, please do let me know, either in the comments or via the PM system.

Learning from this sequence

When collating this sequence, I tried to pick content that makes the most important points simply and concisely. I recommend reading through each post carefully, taking the time to understand each paragraph. The posts range from informal arguments to formal theorems, but even for the formal theorems the formalization of the problem could be changed to invalidate the theorem. Learn from this however you best learn; my preferred method is to try and disprove the argument in the post until I feel like I understand what the post actually conveys.

While this sequence as it stands has no exercises, what it does have is a surrounding forum and community. Here are a few actions you can take to aid both your and others’ understanding of the core concepts:

- Leave a comment with a concise summary of what you understand to be the post/paper’s main point
- Leave a comment outlining a confusion you have with paper/post
- Respond to someone else’s comment to help them understand it better

While I can’t commit to responding to the majority of the comments, I am also excited to help readers understand the content, and please let me know if something I write is confusing.

Each post has a note at the top saying what the post covers and who should read it. You can read through these notes and decide whether they are important for you. That said, the posts are written and organized assuming that you have read prior posts in the sequence, and many points will not make sense if read out of order.

What is ambitious value learning?

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

I think of ambitious value learning as a proposed solution to the specification problem, which I define as the problem of *defining* the behavior that we would want to see from our AI system. I italicize "defining" to emphasize that this is *not* the problem of actually *computing* behavior that we want to see -- that's the full AI safety problem. Here we are allowed to use hopelessly impractical schemes, as long as the resulting definition would allow us to *in theory* compute the behavior that an AI system would take, perhaps with assumptions like infinite computing power or arbitrarily many queries to a human. (Although we do prefer specifications that seem like they could admit an efficient implementation.) In terms of DeepMind's classification, we are looking for a design specification that exactly matches the ideal specification. HCH and indirect normativity are examples of attempts at such specifications.

We will consider a model in which our AI system is maximizing the expected utility of some *explicitly* represented utility function that can depend on history. (It does not matter materially whether we consider utility functions or reward functions, as long as they can depend on history.) The utility function may be learned from data, or designed by hand, but it must be an explicit part of the AI that is then maximized.

I will not justify this model for now, but simply assume it by fiat and see where it takes us. I'll note briefly that this model is often justified by the VNM utility theorem and AIXI, and as the natural idealization of reinforcement learning, which aims to maximize the expected sum of rewards, although typically rewards in RL depend only on states.

A lot of conceptual arguments, as well as experiences with specification gaming, suggest that we are unlikely to be able to simply think hard and write down a good specification, since even small errors in specifications can lead to bad results. However, machine learning is particularly good at narrowing down on the correct hypothesis among a vast space of possibilities using data, so perhaps we could determine a good specification from some suitably chosen source of data? This leads to the idea of ambitious value learning, where we *learn* an explicit utility function from human behavior for the AI to maximize.

This is very related to inverse reinforcement learning (IRL) in the machine learning literature, though not all work on IRL is relevant to ambitious value learning. For example, much work on IRL is aimed at *imitation learning*, which would in the best case allow you to match human performance, but not to exceed it. Ambitious value learning is, well, more ambitious -- it aims to learn a utility function that captures "what humans care about", so that an AI system that optimizes this utility function more capably can *exceed* human performance, making the world better for humans than they could have done themselves.

It may sound like we would have solved the entire AI safety problem if we could do ambitious value learning -- surely if we have a good utility function we would be done. Why then do I think of it as a solution to just the specification problem? This is because ambitious value learning by itself would not be enough for safety, except under the assumption of as much compute and data as desired. These are really powerful assumptions -- for example, I'm assuming you can get data where you put a human in an arbitrarily complicated simulated environment with fake memories of their life so far and see what they do. This allows us to ignore many things that would likely be a problem in practice, such as:

- Attempting to use the utility function to choose actions before it has converged
- Distributional shift causing the learned utility function to become invalid
- Local minima preventing us from learning a good utility function, or from optimizing the learned utility function correctly

The next few posts in this sequence will consider the suitability of ambitious value learning as a solution to the specification problem. Most of them will consider whether ambitious value learning is possible in the setting above (infinite compute and data). One post will consider practical issues with the application of IRL to infer a utility function suitable for ambitious value learning, while still assuming that the resulting utility function can be perfectly maximized (which is equivalent to assuming infinite compute and a perfect model of the environment *after* IRL has run).

The easy goal inference problem is still hard

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

Posted as part of the AI Alignment Forum sequence on Value Learning.

> **Rohin's note:** In this post (original here), Paul Christiano analyzes the ambitious value learning approach. He considers a more general view of ambitious value learning where you infer preferences more generally (i.e. not necessarily in the form of a utility function), and you can ask the user about their preferences, but it's fine to imagine that you infer a utility function from data and then optimize it. The key takeaway is that in order to infer preferences that can lead to superhuman performance, it is necessary to understand how humans are biased, which seems very hard to do even with infinite data.

One approach to the AI control problem goes like this:

1. Observe what the user of the system says and does.
2. Infer the user's preferences.
3. Try to make the world better according to the user's preference, perhaps while working alongside the user and asking clarifying questions.

This approach has the major advantage that we can begin empirical work today—we can actually build systems which observe user behavior, try to figure out what the user wants, and then help with that. There are many applications that people care about already, and we can set to work on making rich toy models.

It seems great to develop these capabilities in parallel with other AI progress, and to address whatever difficulties actually arise, as they arise. That is, in each domain where AI can act effectively, we'd like to ensure that AI can also act effectively in the service of goals inferred from users (and that this inference is good enough to support foreseeable applications).

This approach gives us a nice, concrete model of each difficulty we are trying to address. It also provides a relatively clear indicator of whether our ability to control AI lags behind our ability to build it. And by being technically interesting and economically meaningful now, it can help actually integrate AI control with AI practice.

Overall I think that this is a particularly promising angle on the AI safety problem.

Modeling imperfection

That said, I think that this approach rests on an optimistic assumption: that it's possible to model a human as an imperfect rational agent, and to extract the real values which the human is imperfectly optimizing. Without this assumption, it seems like some additional ideas are necessary.

To isolate this challenge, we can consider a vast simplification of the goal inference problem:

The easy goal inference problem: Given no algorithmic limitations and access to the complete human policy — a lookup table of what a human would do after making any sequence of observations — find any reasonable representation of any reasonable approximation to what that human wants.

I think that this problem remains wide open, and that we've made very little headway on the general case. We can make the problem even easier, by considering a human in a simple toy universe making relatively simple decisions, but it still leaves us with a very tough problem.

It's not clear to me whether or exactly how progress in AI will make this problem easier. I can certainly see how enough progress in cognitive science might yield an answer, but it seems much more likely that it will instead tell us "Your question wasn't well defined." What do we do then?

I am especially interested in this problem because I think that "business as usual" progress in AI will probably lead to the ability to predict human behavior relatively well, and to emulate the performance of experts. So I really care about the residual — what do we need to know to address AI control, beyond what we need to know to build AI?

Narrow domains

We can solve the very easy goal inference problem in sufficiently narrow domains, where humans can behave approximately rationally and a simple error model is approximately right. So far this has been good enough.

But in the long run, humans make many decisions whose consequences aren't confined to a simple domain. This approach can can work for driving from point A to point B, but probably can't work for designing a city, running a company, or setting good policies.

There may be an approach which uses inverse reinforcement learning in simple domains as a building block in order to solve the whole AI control problem. Maybe it's not even a terribly complicated approach. But it's not a trivial problem, and I don't think it can be dismissed easily without some new ideas.

Modeling "mistakes" is fundamental

If we want to perform a task as well as an expert, inverse reinforcement learning is clearly a powerful approach.

But in in the long-term, many important applications require AIs to make decisions which are better than those of available human experts. This is part of the promise of AI, and it is the scenario in which AI control becomes most challenging.

In this context, we can't use the usual paradigm — "more accurate models are better." A perfectly accurate model will take us exactly to human mimicry and no farther.

The possible extra oomph of inverse reinforcement learning comes from an explicit model of the human’s mistakes or bounded rationality. It’s what specifies what the AI should do differently in order to be “smarter,” what parts of the human’s policy it should throw out. So it implicitly specifies which of the human behaviors the AI should keep. The error model isn’t an afterthought — it’s the main affair.

Modeling “mistakes” is hard

Existing error models for inverse reinforcement learning tend to be very simple, ranging from Gaussian noise in observations of the expert’s behavior or sensor readings, to the assumption that the expert’s choices are randomized with a bias towards better actions.

In fact humans are not rational agents with some noise on top. Our decisions are the product of a complicated mess of interacting process, optimized by evolution for the reproduction of our children’s children. It’s not clear there is any good answer to what a “perfect” human would do. If you were to find any principled answer to “what is the human brain optimizing?” the single most likely bet is probably something like “reproductive success.” But this isn’t the answer we are looking for.

I don’t think that writing down a model of human imperfections, which describes how humans depart from the rational pursuit of fixed goals, is likely to be any easier than writing down a complete model of human behavior.

We can’t use normal AI techniques to learn this kind of model, either — what is it that makes a model good or bad? The standard view — “more accurate models are better” — is fine as long as your goal is just to emulate human performance. But this view doesn’t provide guidance about how to separate the “good” part of human decisions from the “bad” part.

So what?

It’s reasonable to take the attitude “Well, we’ll deal with that problem when it comes up.” But I think that there are a few things that we can do productively in advance.

- Inverse reinforcement learning / goal inference research motivated by applications to AI control should probably pay particular attention to the issue of modeling mistakes, and to the challenges that arise when trying to find a policy better than the one you are learning from.
- It’s worth doing more theoretical research to understand this kind of difficulty and how to address it. This research can help identify other practical approaches to AI control, which can then be explored empirically.

Humans can be assigned any values whatsoever…

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

(Re)Posted as part of the AI Alignment Forum sequence on Value Learning.

> **Rohin's note:** In the last post, we saw that a good broad value learning approach would need to understand the systematic biases in human planning in order to achieve superhuman performance. Perhaps we can just use machine learning again and learn the biases and reward simultaneously? This post by Stuart Armstrong (original here) and the associated paper say: "Not without more assumptions."
>
> This post comes from a theoretical perspective that may be alien to ML researchers; in particular, it makes an argument that simplicity priors do not solve the problem pointed out here, where simplicity is based on Kolmogorov complexity (which is an instantiation of the Minimum Description Length principle). The analog in machine learning would be an argument that regularization would not work. The proof used is specific to Kolmogorov complexity and does not clearly generalize to arbitrary regularization techniques; however, I view the argument as being suggestive that regularization techniques would also be insufficient to address the problems raised here.

Humans have no values… nor do any agent. Unless you make strong assumptions about their rationality. And depending on those assumptions, you get humans to have any values.

An agent with no clear preferences

There are three buttons in this world, B(0), B(1), and X, and one agent H.

B(0) and B(1) can be operated by H, while X can be operated by an outside observer.

H will initially press button B(0); if ever X is pressed, the agent will switch to pressing B(1). If X is pressed again, the agent will switch back to pressing B(0), and so on. After a large number of turns N, H will shut off. That's the full algorithm for H.

So the question is, what are the values/preferences/rewards of H? There are three natural reward functions that are plausible:

- R(0), which is linear in the number of times B(0) is pressed.

- $R(1)$, which is linear in the number of times $B(1)$ is pressed.
- $R(2) = I(E, X)R(0) + I(O, X)R(1)$, where $I(E, X)$ is the indicator function for X being pressed an even number of times, $I(O, X) = 1 - I(E, X)$ being the indicator function for X being pressed an odd number of times.

For $R(0)$, we can interpret H as an $R(0)$ maximising agent which X overrides. For $R(1)$, we can interpret H as an $R(1)$ maximising agent which X releases from constraints. And $R(2)$ is the “H is always fully rational” reward. Semantically, these make sense for the various $R(i)$’s being a true and natural reward, with $X =$“coercive brain surgery” in the first case, $X =$“release H from annoying social obligations” in the second, and $X =$ “switch which of $R(0)$ and $R(1)$ gives you pleasure” in the last case.

But note that there is no semantic implications here, all that we know is H, with its full algorithm. If we wanted to deduce its true reward for the purpose of something like Inverse Reinforcement Learning (IRL), what would it be?

Modelling human (ir)rationality and reward

Now let’s talk about the preferences of an actual human. We all know that humans are not always rational. But even if humans were fully rational, the fact remains that we are physical, and vulnerable to things like coercive brain surgery (and in practice, to a whole host of other more or less manipulative techniques). So there will be the equivalent of “button X” that overrides human preferences. Thus, “not immortal and unchangeable” is in practice enough for the agent to be considered “not fully rational”.

Now assume that we’ve thoroughly observed a given human h (including their internal brain wiring), so we know the human policy $\pi(h)$ (which determines their actions in all circumstances). This is, in practice all that we can ever observe - once we know $\pi(h)$ perfectly, there is nothing more that observing h can teach us.

Let R be a possible human reward function, and $\mathbf{R}$ the set of such rewards. A human (ir)rationality planning algorithm p (hereafter referred to as a planner), is a map from $\mathbf{R}$ to the space of policies (thus $p(R)$ says how a human with reward R will actually behave - for example, this could be bounded rationality, rationality with biases, or many other options). Say that the pair (p, R) is compatible if $p(R) = \pi(h)$. Thus a human with planner p and reward R would behave as h does.

What possible compatible pairs are there? Here are some candidates:

- $(p(0), R(0))$, where $p(0)$ and $R(0)$ are some “plausible” or “acceptable” planner and reward functions (what this means is a big question).
- $(p(1), R(1))$, where $p(1)$ is the “fully rational” planner, and $R(1)$ is a reward that fits to give the required policy.
- $(p(2), R(2))$, where $R(2) = -R(1)$, and $p(2) = -p(1)$, where $-p(R)$ is defined as $p(-R)$; here $p(2)$ is the “fully anti-rational” planner.
- $(p(3), R(3))$, where $p(3)$ maps all rewards to $\pi(h)$, and $R(3)$ is trivial and constant.
- $(p(4), R(4))$, where $p(4) = -p(0)$ and $R(4) = -R(0)$.

Distinguishing among compatible pairs

How can we distinguish between compatible pairs? At first appearance, we can’t. That’s because, by their definition of compatible, all pairs produce the correct policy $\pi(h)$. And once we have $\pi(h)$, further observations of h tell us nothing.

I initially thought that Kolmogorov or algorithmic complexity might help us here. But in fact:

Theorem: The pairs $(p(i), R(i))$, $i \geq 1$, are either simpler than $(p(0), R(0))$, or differ in Kolmogorov complexity from it by a constant that is independent of $(p(0), R(0))$.

Proof: The cases of $i = 4$ and $i = 2$ are easy, as these differ from $i = 0$ and $i = 1$ by two minus signs. Given $(p(0), R(0))$, a fixed-length algorithm computes $\pi(h)$. Then a fixed length algorithm defines $p(3)$ (by mapping input to $\pi(h)$). Furthermore, given $\pi(h)$ and any history η, a fixed length algorithm computes the action $a(\eta)$ the agent will take; then a fixed length algorithm defines $R(1)(\eta, a(\eta)) = 1$ and $R(1)(\eta, b) = 0$ for $b \neq a(\eta)$.

So the Kolmogorov complexity can shift between p and R (all in R for $i = 1, 2$, all in p for $i = 3$), but it seems that the complexity of the pair doesn’t go up during these shifts.

This is puzzling. It seems that, in principle, one cannot assume anything about H’s reward at all! R(2) = −R(1), R(4) = −R(0), and p(3) is compatible with any possible reward R. If we give up the assumption of human rationality - which we must - it seems we can’t say anything about the human reward function. So it seems IRL must fail.

Latent Variables and Model Mis-Specification

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

Posted as part of the AI Alignment Forum sequence on Value Learning.

> **Rohin's note:** So far, we've seen that ambitious value learning needs to understand human biases, and that we can't simply learn the biases in tandem with the reward. Perhaps we could hardcode a specific model of human biases? Such a model is likely to be incomplete and inaccurate, but it will perform better than assuming an optimal human, and as we notice failure modes we can improve the model. In the language of this post by Jacob Steinhardt (original here), we are using a mis-specified human model. The post talks about why model mis-specification is worse than it may seem at first glance.
>
> This post is fairly technical and may not be accessible if you don't have a background in machine learning. If so, you can skip this post and still understand the rest of the posts in the sequence. However, if you want to do ML-related safety research, I strongly recommend putting in the effort to understand the problems that can arise with mis-specification.

Machine learning is very good at optimizing predictions to match an observed signal — for instance, given a dataset of input images and labels of the images (e.g. dog, cat, etc.), machine learning is very good at correctly predicting the label of a new image. However, performance can quickly break down as soon as we care about criteria other than predicting observables. There are several cases where we might care about such criteria:

- In scientific investigations, we often care less about predicting a specific observable phenomenon, and more about what that phenomenon implies about an underlying scientific theory.
- In economic analysis, we are most interested in what policies will lead to desirable outcomes. This requires predicting what would counterfactually happen if we were to enact the policy, which we (usually) don't have any data about.
- In machine learning, we may be interested in learning value functions which match human preferences (this is especially important in complex settings where it is hard to specify a satisfactory value function by hand). However, we are unlikely to observe information about the value function directly, and instead must infer it implicitly. For instance, one might infer a value function for autonomous driving by observing the actions of an expert driver.

In all of the above scenarios, the primary object of interest — the scientific theory, the effects of a policy, and the value function, respectively — is not part of the observed data. Instead, we can think of it as an unobserved (or "latent") variable in the model we are using to make predictions. While we might hope that a model that makes good predictions will also place correct values on unobserved variables as well, this need not be the case in general, especially if the model is *mis-specified*.

I am interested in latent variable inference because I think it is a potentially important sub-problem for building AI systems that behave safely and are aligned with human values. The connection is most direct for value learning, where the value function is the latent variable of interest and the fidelity with which it is learned directly impacts the well-behavedness of the system. However, one can imagine other uses as well, such as making sure that the concepts that an AI learns sufficiently match the concepts that the human designer had in mind. It will also turn out that latent variable inference is related to *counterfactual reasoning*, which has a large number of tie-ins with building safe AI systems that I will elaborate on in forthcoming posts.

The goal of this post is to explain why problems show up if one cares about predicting latent variables rather than observed variables, and to point to a research direction (counterfactual reasoning) that I find promising for addressing these issues. More specifically, in the remainder of this post, I will: (1) give some formal settings where we want to infer unobserved variables and explain why we can run into problems; (2) propose a possible approach to resolving these problems, based on counterfactual reasoning.

1 Identifying Parameters in Regression Problems

Suppose that we have a regression model $p_\theta(y|x)$, which outputs a probability distribution over y given a value for x. Also suppose we are explicitly interested in identifying the "true" value of θ rather than simply making good predictions about y given x. For instance, we might be interested in whether smoking causes cancer, and so we care not just about predicting whether a given person will get cancer (y) given information about that person (x), but specifically whether the coefficients in θ that correspond to a history of smoking are large and positive.

In a typical setting, we are given data points $(x_1, y_1) \ldots (x_n, y_n)$ on which to fit a model. Most methods of training machine learning systems optimize predictive performance, i.e. they will output a parameter $\hat{\theta}$ that (approximately) maximizes $\sum_{i=1}^{n} \log p_\theta(y_i, x_i)$. For instance, for a linear regression problem we have $\log p_\theta(y_i, x_i) = -(y_i - \langle \theta, x_i \rangle)^2$. Various more sophisticated methods might employ some form of regularization to reduce overfitting, but they are still fundamentally trying to maximize some measure of predictive accuracy, at least in the limit of infinite data.

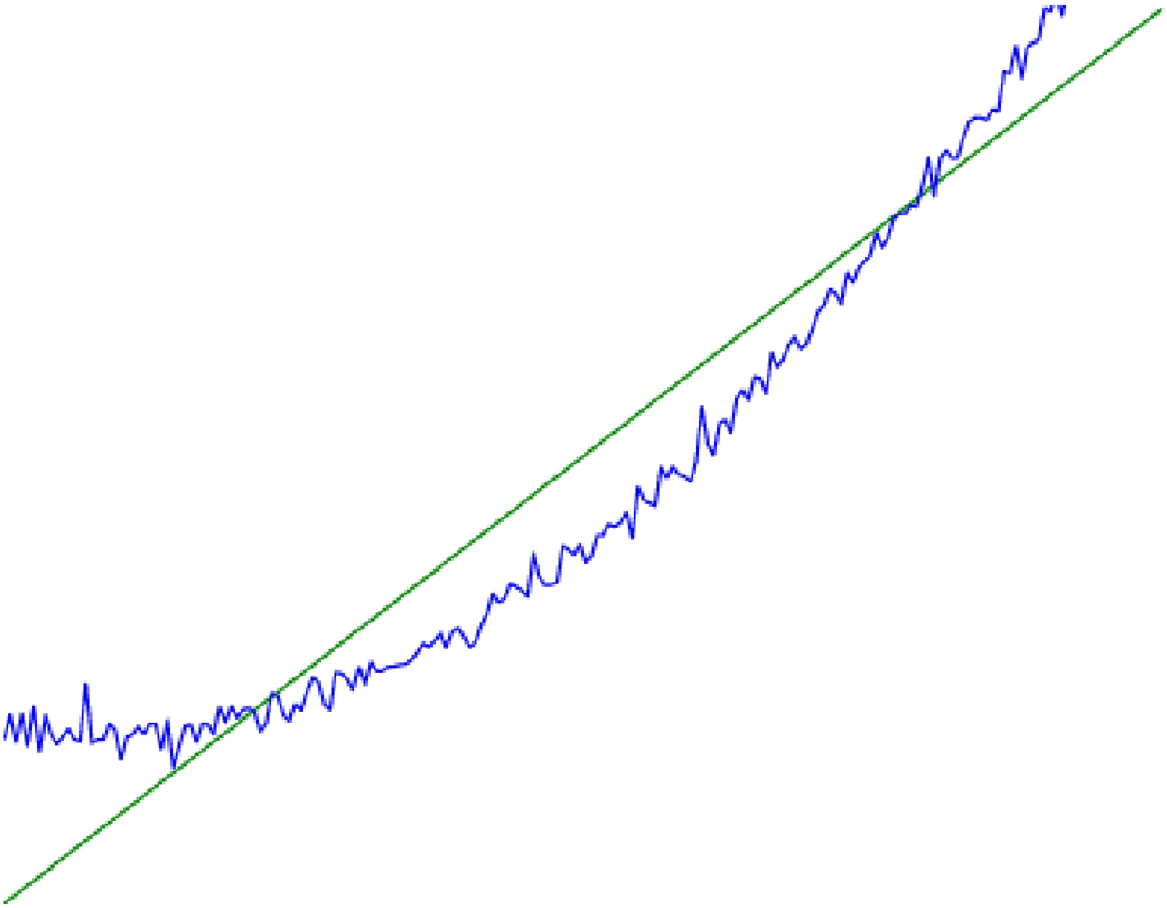
Call a model **well-specified** if there is some parameter θ^* for which $p_{\theta^*}(y, x)$ matches the true distribution over y, and call a model **mis-specified** if no such θ^* exists. One can show that for well-specified models, maximizing predictive accuracy works well (modulo a number of technical conditions). In particular, maximizing $\sum_{i=1}^{n} \log p_\theta(y_i, x_i)$ will (asymptotically, as $n \to \infty$) lead to recovering the parameter θ^*.

However, if a model is mis-specified**,** then it is not even clear what it means to correctly infer θ. We could declare the θ maximizing predictive accuracy to be the "correct" value of θ, but this has issues:

1. While θ might do a good job of predicting y in the settings we've seen, it may not predict y well in very different settings.
2. If we care about determining θ for some scientific purpose, then good predictive accuracy may be an unsuitable metric. For instance, even though margarine consumption might correlate well with (and hence be a good predictor of) divorce rate, that doesn't mean that there is a causal relationship between the two.

The two problems above also suggest a solution: we will say that we have done a good job of inferring a value for θ if θ can be used to make *good predictions in a wide variety of situations*, and not just the situation we happened to train the model on. (For the latter case of predicting causal relationships, the "wide variety of situations" should include the situation in which the relevant causal intervention is applied.)

Note that both of the problems above are different from the typical statistical problem of overfitting. Classically, overfitting occurs when a model is too complex relative to the amount of data at hand, but even if we have a large amount of data the problems above could occur. This is illustrated in the following graph:



Here the blue line is the data we have (x, y), and the green line is the model we fit (with slope and intercept parametrized by θ). We have more than enough data to fit a line to it. However, because the true relationship is quadratic, the best linear fit depends heavily on the distribution of the training data. If we had fit to a different part of the quadratic, we would have gotten a potentially very different result. Indeed, in this situation, there is no linear relationship that can do a good job of extrapolating to new situations, unless the domain of those new situations is restricted to the part of the quadratic that we've already seen.

I will refer to the type of error in the diagram above as *mis-specification error*. Again, mis-specification error is different from error due to overfitting. Overfitting occurs when there is too little data and noise is driving the estimate of the model; in contrast, mis-specification error can occur even if there is plenty of data, and instead occurs because the best-performing model is different in different scenarios.

2 Structural Equation Models

We will next consider a slightly subtler setting, which in economics is referred to as a *structural equation model*. In this setting we again have an output y whose distribution depends on an input x, but now this relationship is mediated by an *unobserved* variable z. A common example is a discrete choice model, where consumers make a choice among multiple goods (y) based on a consumer-specific utility function (z) that is influenced by demographic and other information about the consumer (x). Natural language processing provides another source of examples: in semantic parsing, we have an input utterance (x) and output denotation (y), mediated by a latent logical form z; in machine translation, we have input and output sentences (x and y) mediated by a latent alignment (z).

Symbolically, we represent a structural equation model as a parametrized probability distribution $p_\theta(y, z|x)$, where we are trying to fit the parameters θ. Of course, we can always turn a structural equation model into a regression model by using the identity $p_\theta(y|x) = \sum_z p_\theta(y, z|x)$, which allows us to ignore z altogether. In economics this is called a *reduced form model*. We use structural equation models if we are specifically interested in the unobserved variable z (for instance, in the examples above we are interested in the value function for each individual, or in the logical form representing the sentence's meaning).

In the regression setting where we cared about identifying θ, it was obvious that there was no meaningful "true" value of θ when the model was mis-specified. In this structural equation setting, we now care about the latent variable z, which can take on a meaningful true value (e.g. the actual utility function of a given individual) even if the overall model $p_\theta(y, z|x)$ is mis-specified. It is therefore tempting to think that if we fit parameters θ and use them to impute z, we will have meaningful information about the actual utility functions of individual consumers. However, this is a notational sleight of hand — just because we call z

“the utility function” does not make it so. The variable z need not correspond to the actual utility function of the consumer, nor does the consumer’s preferences even need to be representable by a utility function.

We can understand what goes wrong by consider the following procedure, which formalizes the proposal above:

1. Find θ to maximize the predictive accuracy on the observed data, $\sum_{i=1}^{z} \log p_{\theta}(y_i, x_i)$, where $p_{\theta}(y_i | x_i) = \sum_z p_{\theta}(y_i, z | x_i)$. Call the result θ_0.
2. Using this value θ_0, treat z_i as being distributed according to $p_{\theta_0}(z | x_i, y_i)$. On a new value x_+ for which y is not observed, treat z_+ as being distributed according to $p_{\theta_0}(z | x_+)$.

As before, if the model is well-specified, one can show that such a procedure asymptotically outputs the correct probability distribution over z. However, if the model is mis-specified, things can quickly go wrong. For example, suppose that y represents what choice of drink a consumer buys, and z represents consumer utility (which might be a function of the price, attributes, and quantity of the drink). Now suppose that individuals have preferences which are influenced by unmodeled covariates: for instance, a preference for cold drinks on warm days, while the input x does not have information about the outside temperature when the drink was bought. This could cause any of several effects:

- If there is a covariate that happens to correlate with temperature in the data, then we might conclude that that covariate is predictive of preferring cold drinks.
- We might increase our uncertainty about z to capture the unmodeled variation in x.
- We might implicitly increase uncertainty by moving utilities closer together (allowing noise or other factors to more easily change the consumer’s decision).

In practice we will likely have some mixture of all of these, and this will lead to systematic biases in our conclusions about the consumers’ utility functions.

The same problems as before arise: while we by design place probability mass on values of z that correctly predict the observation y, under model mis-specification this could be due to spurious correlations or other perversities of the model. Furthermore, even though predictive performance is high on the observed data (and data similar to the observed data), there is no reason for this to continue to be the case in settings very different from the observed data, which is particularly problematic if one is considering the effects of an intervention. For instance, while inferring preferences between hot and cold drinks might seem like a silly example, the design of timber auctions constitutes a much more important example with a roughly similar flavour, where it is important to correctly understand the utility functions of bidders in order to predict their behaviour under alternative auction designs (the model is also more complex, allowing even more opportunities for mis-specification to cause problems).

3 A Possible Solution: Counterfactual Reasoning

In general, under model mis-specification we have the following problems:

- It is often no longer meaningful to talk about the “true” value of a latent variable θ (or at the very least, not one within the specified model family).
- Even when there is a latent variable z with a well-defined meaning, the imputed distribution over z need not match reality.

We can make sense of both of these problems by thinking in terms of *counterfactual reasoning*. Without defining it too formally, counterfactual reasoning is the problem of making good predictions not just in the actual world, but in a wide variety of counterfactual worlds that “could” exist. (I recommend this paper as a good overview for machine learning researchers.)

While typically machine learning models are optimized to predict well on a specific distribution, systems capable of counterfactual reasoning must make good predictions on many distributions (essentially any distribution that can be captured by a reasonable counterfactual). This stronger guarantee allows us to resolve many of the issues discussed above, while still thinking in terms of predictive performance, which historically seems to have been a successful paradigm for machine learning. In particular:

- While we can no longer talk about the “true” value of θ, we can say that a value of θ is a “good” value if it makes good predictions on not just a single test distribution, but many different counterfactual test distributions. This allows us to have more confidence in the generalizability of any inferences we draw based on θ (for instance, if θ is the coefficient vector for a regression problem, any variable with positive sign is likely to robustly correlate with the response variable for a wide variety of settings).
- The imputed distribution over a variable z must also lead to good predictions for a wide variety of distributions. While this does not force z to match reality, it is a much stronger condition and does at least mean that any aspect of z that can be measured in some counterfactual world must correspond to reality. (For instance, any aspect of a utility function that could at least counterfactually result in a specific action would need to match reality.)
- We will successfully predict the effects of an intervention, as long as that intervention leads to one of the counterfactual distributions considered.

(Note that it is less clear how to actually train models to optimize counterfactual performance, since we typically won’t observe the counterfactuals! But it does at least define an end goal with good properties.)

Many people have a strong association between the concepts of “counterfactual reasoning” and “causal reasoning”. It is important to note that these are distinct ideas; causal reasoning is a type of counterfactual reasoning (where the counterfactuals are often thought of as centered around interventions), but I think of counterfactual reasoning as any type of reasoning that involves making robustly correct statistical inferences across a wide variety of distributions. On the other hand, some people take robust statistical correlation to be the *definition* of a causal relationship, and thus do consider causal and counterfactual reasoning to be the same thing.

I think that building machine learning systems that can do a good job of counterfactual reasoning is likely to be an important challenge, especially in cases where reliability and safety are important, and necessitates changes in how we evaluate machine learning models. In my mind, while the Turing test has many flaws, one thing it gets very right is the ability to evaluate the accuracy of counterfactual predictions (since dialogue provides the opportunity to set up counterfactual worlds via shared hypotheticals). In contrast, most existing tasks focus on repeatedly making the same type of prediction with respect to a fixed test distribution. This latter type of benchmarking is of course easier and more clear-cut, but fails to probe important aspects of our models. I think it would be very exciting to design good benchmarks that require systems to do counterfactual reasoning, and I would even be happy to incentivize such work monetarily.

Acknowledgements

Thanks to Michael Webb, Sindy Li, and Holden Karnofsky for providing feedback on drafts of this post. If any readers have additional feedback, please feel free to send it my way.

Model Mis-specification and Inverse Reinforcement Learning

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

Posted as part of the AI Alignment Forum sequence on Value Learning.

> **Rohin's note:** While I motivated the last post with an example of using a specific model for human biases, in this post (original here), Jacob Steinhardt and Owain Evans point out that model mis-specification can arise in other parts of inverse reinforcement learning as well. The arguments here consider some more practical concerns (for example, the worries about getting only short-term data for each human would not be a problem if you had the entire human policy).

In my previous post, “Latent Variables and Model Mis-specification”, I argued that while machine learning is good at optimizing accuracy on observed signals, it has less to say about correctly inferring the values for unobserved variables in a model. In this post I’d like to focus in on a specific context for this: inverse reinforcement learning (Ng et al. 2000, Abbeel et al. 2004, Ziebart et al. 2008, Ho et al 2016), where one observes the actions of an agent and wants to infer the preferences and beliefs that led to those actions. For this post, I am pleased to be joined by Owain Evans, who is an active researcher in this area and has co-authored an online book about building models of agents (see here in particular for a tutorial on inverse reinforcement learning and inverse planning).

Owain and I are particularly interested in inverse reinforcement learning (IRL) because it has been proposed (most notably by Stuart Russell) as a method for learning human values in the context of AI safety; among other things, this would eventually involve learning and correctly implementing human values by artificial agents that are much more powerful, and act with much broader scope, than any humans alive today. While we think that overall IRL is a promising route to consider, we believe that there are also a number of non-obvious pitfalls related to performing IRL with a mis-specified model. The role of IRL in AI safety is to infer human values, which are represented by a reward function or utility function. But crucially, human values (or human reward functions) are never directly observed.

Below, we elaborate on these issues. We hope that by being more aware of these issues, researchers working on inverse reinforcement learning can anticipate and address the resulting failure modes. In addition, we think that considering issues caused by model mis-specification in a particular concrete context can better elucidate the general issues pointed to in the previous post on model mis-specification.

Specific Pitfalls for Inverse Reinforcement Learning

In “Latent Variables and Model Mis-specification”, Jacob talked about *model mis-specification*, where the “true” model does not lie in the model family being considered. We encourage readers to read that post first, though we’ve also tried to make the below readable independently.

In the context of inverse reinforcement learning, one can see some specific problems that might arise due to model mis-specification. For instance, the following are things we could misunderstand about an agent, which would cause us to make incorrect inferences about the agent’s values:

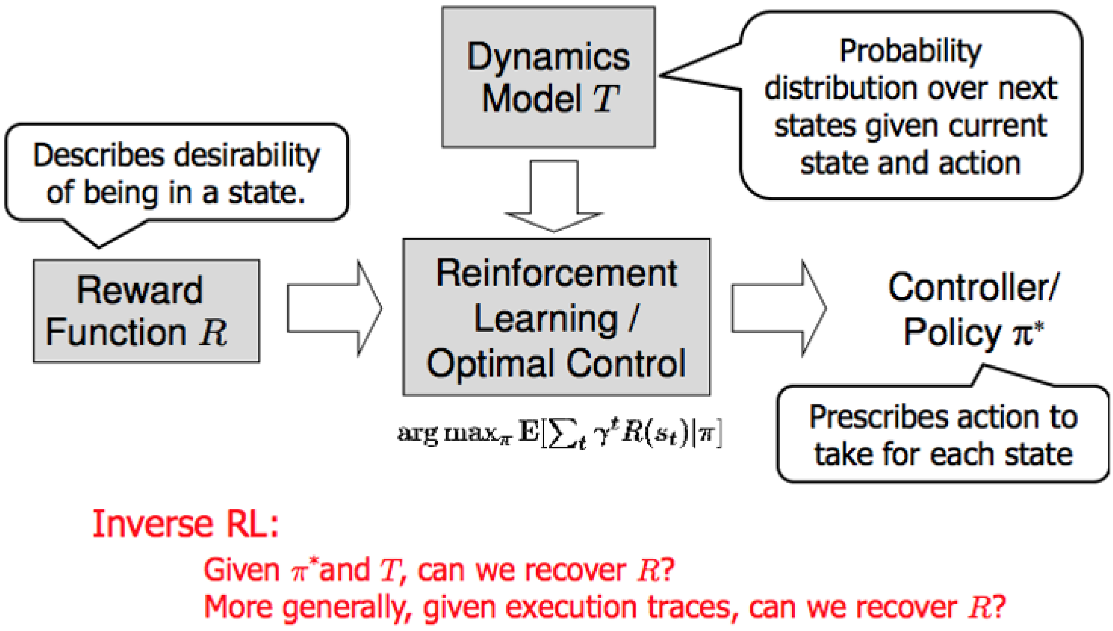
- The **actions** of the agent. If we believe that an agent is capable of taking a certain action, but in reality they are not, we might make strange inferences about their values (for instance, that they highly value not taking that action). Furthermore, if our data is e.g. videos of human behavior, we have an additional inference problem of recognizing actions from the frames.
- The **information** available to the agent. If an agent has access to more information than we think it does, then a plan that seems irrational to us (from the perspective of a given reward function) might actually be optimal for reasons that we fail to appreciate. In the other direction, if an agent has less information than we think, then we might incorrectly believe that they don't value some outcome A, even though they really only failed to obtain A due to lack of information.
- The **long-term plans** of the agent. An agent might take many actions that are useful in accomplishing some long-term goal, but not necessarily over the time horizon that we observe the agent. Inferring correct values thus also requires inferring such long-term goals. In addition, long time horizons can make models more brittle, thereby exacerbating model mis-specification issues.

There are likely other sources of error as well. The general point is that, given a mis-specified model of the agent, it is easy to make incorrect inferences about an agent's values if the optimization pressure on the learning algorithm is only towards predicting actions correctly in-sample.

In the remainder of this post, we will cover each of the above aspects — actions, information, and plans — in turn, giving both quantitative models and qualitative arguments for why model mis-specification for that aspect of the agent can lead to perverse beliefs and behavior. First, though, we will briefly review the definition of inverse reinforcement learning and introduce relevant notation.

Inverse Reinforcement Learning: Definition and Notations

In inverse reinforcement learning, we want to model an agent taking actions in a given environment. We therefore suppose that we have a **state space** S (the set of states the agent and environment can be in), an **action space** A (the set of actions the agent can take), and a **transition function** $T(s^{'}|s,a)$, which gives the probability of moving from state s to state $s^{'}$ when taking action a. For instance, for an AI learning to control a car, the state space would be the possible locations and orientations of the car, the action space would be the set of control signals that the AI could send to the car, and the transition function would be the dynamics model for the car. The tuple of (S,A,T) is called an $MDP\backslash R$, which is a Markov Decision Process without a reward function. (The $MDP\backslash R$ will either have a known horizon or a discount rate γ but we'll leave these out for simplicity.)


Dynamics Model T
Probability distribution over next states given current state and action
Describes desirability of being in a state.
Reward Function R
Reinforcement Learning / Optimal Control
Controller/ Policy π^*
Prescribes action to take for each state
$\arg\max_\pi \mathrm{E}[\sum_t \gamma^t R(s_t)|\pi]$

Inverse RL:

Given $\pi^*$and T, can we recover R?
More generally, given execution traces, can we recover R?

Figure 1: Diagram showing how IRL and RL are related. (Credit: Pieter Abbeel's slides on IRL)

The inference problem for IRL is to infer a reward function R given an optimal policy $\pi^* : S \to A$ for the MDP \R (see Figure 1). We learn about the policy π^* from samples (s, a) of states and the corresponding action according to π^* (which may be random). Typically, these samples come from a trajectory, which records the full history of the agent's states and actions in a single episode:

$$(s_0, a_0), (s_1, a_1), \ldots, (s_n, a_n)$$

In the car example, this would correspond to the actions taken by an expert human driver who is demonstrating desired driving behaviour (where the actions would be recorded as the signals to the steering wheel, brake, etc.).

Given the MDP \R and the observed trajectory, the goal is to infer the reward function R. In a Bayesian framework, if we specify a prior on R we have:

$$P(R | s_{0:n}, a_{0:n}) \propto P(s_{0:n}, a_{0:n} | R) P(R) = P(R) \cdot \prod_{i=0}^{n} P(a_i | s_i, R)$$

The likelihood $P(a_i|s_i, R)$ is just $\pi_R(s)[a_i]$, where π_R is the optimal policy under the reward function R. Note that computing the optimal policy given the reward is in general non-trivial; except in simple cases, we typically approximate the policy using reinforcement learning (see Figure 1). Policies are usually assumed to be noisy (e.g. using a softmax instead of deterministically taking the best action). Due to the challenges of specifying priors, computing optimal policies and integrating over reward functions, most work in IRL uses some kind of approximation to the Bayesian objective (see the references in the introduction for some examples).

Recognizing Human Actions in Data

IRL is a promising approach to learning human values in part because of the easy availability of data. For supervised learning, humans need to produce many labeled instances specialized for a task. IRL, by contrast, is an unsupervised/semi-supervised approach where any record of human behavior is a potential data source. Facebook’s logs of user behavior provide trillions of data-points. YouTube videos, history books, and literature are a trove of data on human behavior in both actual and imagined scenarios. However, while there is lots of existing data that is informative about human preferences, we argue that exploiting this data for IRL will be a difficult, complex task with current techniques.

Inferring Reward Functions from Video Frames

As we noted above, applications of IRL typically infer the reward function R from observed samples of the human policy π^*. Formally, the environment is a known $MPD \backslash R = (S, A, T)$ and the observations are state-action pairs, $(s, a) \sim \pi^*$. This assumes that (a) the environment’s dynamics T are given as part of the IRL problem, and (b) the observations are structured as “state-action” pairs. When the data comes from a human expert parking a car, these assumptions are reasonable. The states and actions of the driver can be recorded and a car simulator can be used for T. For data from YouTube videos or history books, the assumptions fail. The data is a sequence of partial observations: the transition function T is unknown and the data does not separate out *state* and *action*. Indeed, it’s a challenging ML problem to infer human actions from text or videos.

Movie still: What actions are being performed in this situation? (Source)

As a concrete example, suppose the data is a video of two co-pilots flying a plane. The successive frames provide only limited information about the state of the world at each time step and the frames often jump forward in time. So it's more like a POMDP with a complex observation model. Moreover, the actions of each pilot need to be inferred. This is a challenging inference problem, because actions can be subtle (e.g. when a pilot nudges the controls or nods to his co-pilot).

To infer actions from observations, some model relating the true state-action (s, a) to the observed video frame must be used. But choosing any model makes substantive assumptions about how human values relate to their behavior. For example, suppose someone attacks one of the pilots and (as a reflex) he defends himself by hitting back. Is this reflexive or instinctive response (hitting the attacker) an action that is informative about the pilot's values? Philosophers and neuroscientists might investigate this by considering the mental processes that occur before the pilot hits back. If an IRL algorithm uses an off-the-shelf action classifier, it will lock in some (contentious) assumptions about these mental processes. At the same time, an IRL algorithm cannot *learn* such a model because it never directly observes the mental processes that relate rewards to actions.

Inferring Policies From Video Frames

When learning a reward function via IRL, the ultimate goal is to use the reward function to guide an artificial agent's behavior (e.g. to perform useful tasks to humans). This goal can be formalized directly, without including IRL as an intermediate step. For example, in **Apprenticeship Learning,** the goal is to learn a “good” policy for the MDP \R from samples of the human’s policy π* (where π* is assumed to approximately optimize an unknown reward function). In **Imitation Learning,** the goal is simply to learn a policy that is similar to the human’s policy.

Like IRL, policy search techniques need to recognize an agent’s actions to infer their policy. So they have the same challenges as IRL in learning from videos or history books. Unlike IRL, policy search does not explicitly model the reward function that underlies an agent’s behavior. This leads to an additional challenge. Humans and AI systems face vastly different tasks and have different action spaces. Most actions in videos and books would never be performed by a software agent. Even when tasks are similar (e.g. humans driving in the 1930s vs. a self-driving car in 2016), it is a difficult transfer learning problem to use human policies in one task to improve AI policies in another.

IRL Needs Curated Data

We argued that records of human behaviour in books and videos are difficult for IRL algorithms to exploit. Data from Facebook seems more promising: we can store the state (e.g. the HTML or pixels displayed to the human) and each human action (clicks and scrolling). This extends beyond Facebook to any task that can be performed on a computer. While this covers a broad range of tasks, there are obvious limitations. Many people in the world have a limited ability to use a computer: we can’t learn about their values in this way. Moreover, some kinds of human preferences (e.g. preferences over physical activities) seem hard to learn about from behaviour on a computer.

Information and Biases

Human actions depend both on their preferences and their *beliefs*. The beliefs, like the preferences, are never directly observed. For narrow tasks (e.g. people choosing their favorite photos from a display), we can model humans as having full knowledge of the state (as in an MDP). But for most real-world tasks, humans have limited information and their information changes over time (as in a POMDP or RL problem). If IRL assumes the human has full information, then the model is mis-specified and generalizing about what the human would prefer in other scenarios can be mistaken. Here are some examples:

- Someone travels from their house to a cafe, which has already closed. If they are assumed to have full knowledge, then IRL would infer an alternative preference (e.g. going for a walk) rather than a preference to get a drink at the cafe.
- Someone takes a drug that is widely known to be ineffective. This could be because they have a false belief that the drug is effective, or because they picked up the wrong pill, or because they take the drug for its side-effects. Each possible explanation could lead to different conclusions about preferences.
- Suppose an IRL algorithm is inferring a person’s goals from key-presses on their laptop. The person repeatedly forgets their login passwords and has to reset them. This behavior is hard to capture with a POMDP-style model: humans forget some strings of characters and not others. IRL might infer that the person *intends* to repeatedly reset their passwords.

The above arises from humans forgetting information — even if the information is only a short string of characters. This is one way in which humans systematically deviate from rational Bayesian agents. The field of psychology has documented many other deviations. Below we discuss one such deviation — *time-inconsistency* — which has been used to explain temptation, addiction and procrastination.

Time-inconsistency and Procrastination

An IRL algorithm is inferring Alice's preferences. In particular, the goal is to infer Alice's preference for completing a somewhat tedious task (e.g. writing a paper) as opposed to relaxing. Alice has T days in which she could complete the task and IRL observes her working or relaxing on each successive day.

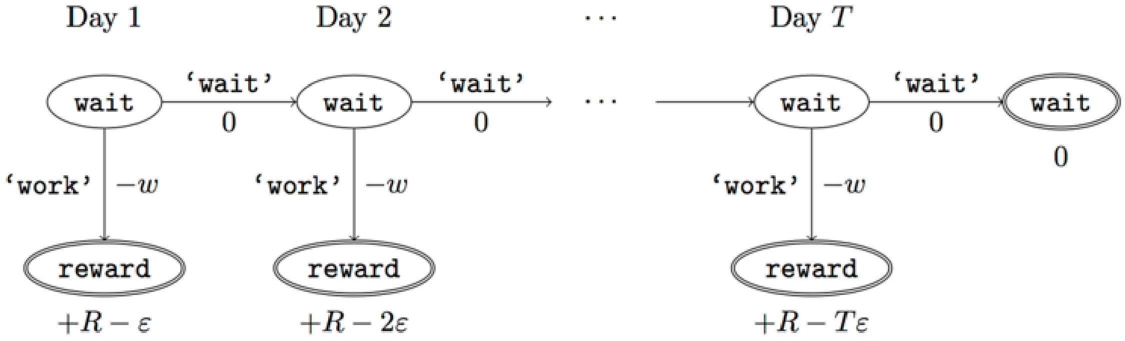


Figure 2. MDP graph for choosing whether to "work" or "wait" (relax) on a task.

Formally, let R be the preference/reward Alice assigns to completing the task. Each day, Alice can "work" (receiving cost w for doing tedious work) or "wait" (cost 0). If she works, she later receives the reward R minus a tiny, linearly increasing cost (because it's better to submit a paper earlier). Beyond the deadline at T, Alice cannot get the reward R. For IRL, we fix ϵ and w and infer R.

Suppose Alice chooses "wait" on Day 1. If she were fully rational, it follows that R (the preference for completing the task) is small compared to w (the psychological cost of doing the tedious work). In other words, Alice doesn't care much about completing the task. Rational agents will do the task on Day 1 or never do it. Yet humans often care deeply about tasks yet leave them until the last minute (when finishing early would be optimal). Here we imagine that Alice has 9 days to complete the task and waits until the last possible day.

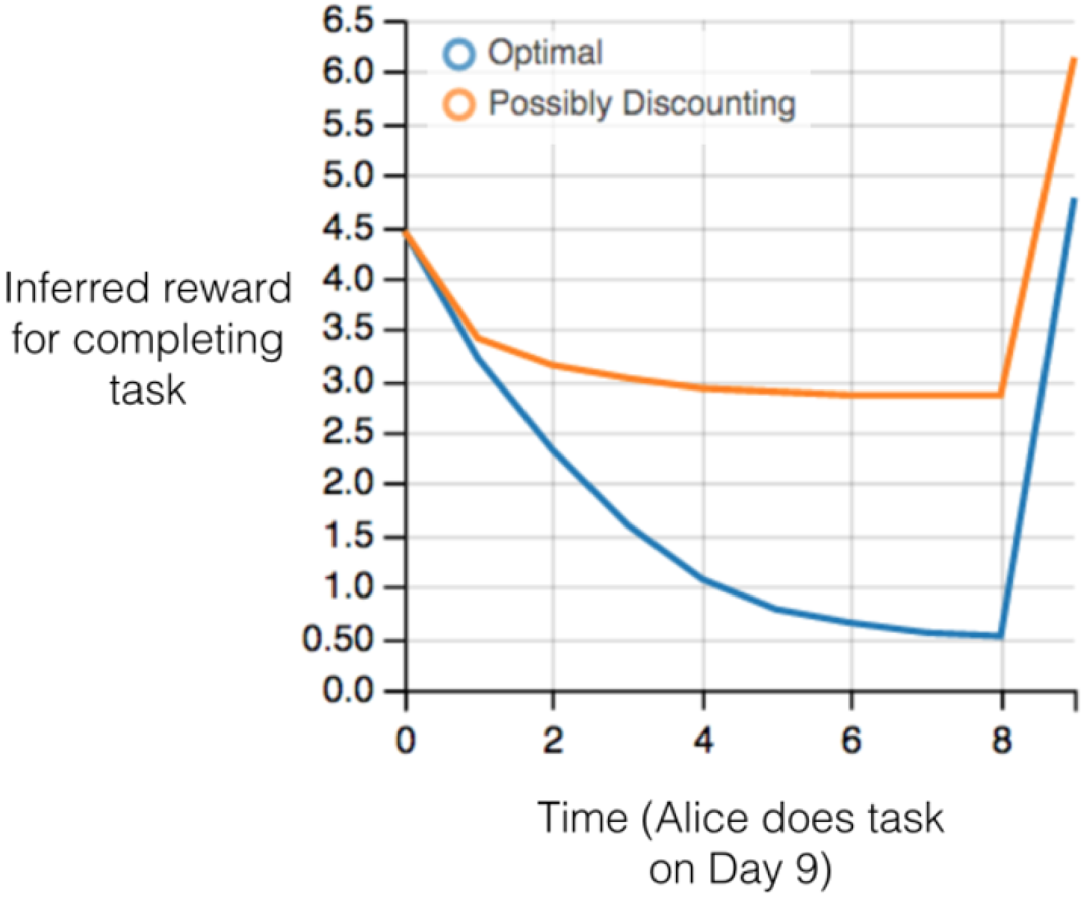


Figure 3: Graph showing IRL inferences for Optimal model (which is mis-specified) and Possibly Discounting Model (which includes hyperbolic discounting). On each day ($x-$axis) the model gets another observation of Alice's choice. The $y-$axis shows the posterior mean for R (reward for task), where the tedious work $w = -1$.

Figure 3 shows results from running IRL on this problem. There is an "Optimal" model, where the agent is optimal up to an unknown level of softmax random noise (a typical assumption for IRL). There is also a "Possibly Discounting" model, where the agent is either softmax optimal or is a hyperbolic discounter (with unknown level of discounting). We do joint Bayesian inference over the completion reward R, the softmax noise and (for "Possibly Discounting") how much the agent hyperbolically discounts. The work cost w is set to -1. Figure 3 shows that after 6 days of observing Alice procrastinate, the "Optimal" model is very confident that Alice does not care about the task ($R < |w|$). When Alice completes the task on the last possible day, the posterior mean on R is not much more than the prior mean. By contrast, the "Possibly Discounting" model never becomes confident that Alice doesn't care about the task. (Note that the gap between the models would be bigger for larger T. The "Optimal" model's posterior on R shoots back to its Day-0 prior because it explains the whole action sequence as due to high softmax noise — optimal agents without noise would either do the task immediately or not at all. Full details and code are here.)

Long-term Plans

Agents will often take long series of actions that generate negative utility for them in the moment in order to accomplish a long-term goal (for instance, studying every night in order to perform well on a test). Such long-term plans can make IRL more difficult for a few reasons. Here we focus on two: (1) IRL systems may not have access to the right type of data for learning about long-term goals, and (2) needing to predict long sequences of actions can make algorithms more fragile in the face of model mis-specification.

(1) Wrong type of data. To make inferences based on long-term plans, it would be helpful to have coherent data about a single agent's actions over a long period of time (so that we can e.g. see the plan unfolding). But in practice we will likely have substantially more data consisting of short snapshots of a large number of different agents (e.g. because many internet services already record user interactions, but it is uncommon for a single person to be exhaustively tracked and recorded over an extended period of time even while they are offline).

The former type of data (about a single representative population measured over time) is called **panel data**, while the latter type of data (about different representative populations measured at each point in time) is called **repeated cross-section data**. The differences between these two types of data is well-studied in econometrics, and a general theme is the following: it is difficult to infer individual-level effects from cross-sectional data.

An easy and familiar example of this difference (albeit not in an IRL setting) can be given in terms of election campaigns. Most campaign polling is cross-sectional in nature: a different population of respondents is polled at each point in time. Suppose that Hillary Clinton gives a speech and her overall support according to cross-sectional polls increases by 2%; what can we conclude from this? Does it mean that 2% of people switched from Trump to Clinton? Or did 6% of people switch from Trump to Clinton while 4% switched from Clinton to Trump?

At a minimum, then, using cross-sectional data leads to a difficult disaggregation problem; for instance, different agents taking different actions at a given point in time could be due to being at different stages in the same plan, or due to having different plans, or some combination of these and other factors. Collecting demographic and other side data can help us (by allowing us to look at variation and shifts within each subpopulation), but it is unclear if this will be sufficient in general.

On the other hand, there are some services (such as Facebook or Google) that do have extensive data about individual users across a long period of time. However, this data has another issue: it is incomplete in a very systematic way (since it only tracks online behaviour). For instance, someone might go online most days to read course notes and Wikipedia for a class; this is data that would likely be recorded. However, it is less likely that one would have a record of that person taking the final exam, passing the class and then getting an internship based on their class performance. Of course, some pieces of this sequence would be inferable based on some people's e-mail records, etc., but it would likely be under-represented in the data relative to the record of Wikipedia usage. In either case, some non-trivial degree of inference would be necessary to make sense of such data.

(2) Fragility to mis-specification. Above we discussed why observing only short sequences of actions from an agent can make it difficult to learn about their long-term plans (and hence to reason correctly about their values). Next we discuss another potential issue — fragility to model mis-specification.

Suppose someone spends 99 days doing a boring task to accomplish an important goal on day 100. A system that is only trying to correctly predict actions will be right 99% of the time if it predicts that the person inherently enjoys boring tasks. Of course, a system that understands the goal and how the tasks lead to the goal will be right 100% of the time, but even minor errors in its understanding could bring the accuracy back below 99%.

The general issue is the following: large changes in the model of the agent might only lead to small changes in the predictive accuracy of the model, and the longer the time horizon on which a goal is realized, the more this might be the case. This means that even slight mis-specifications in the model could tip the scales back in favor of a (very) incorrect reward function. A potential way of dealing with this might be to identify "important" predictions that seem closely tied to the reward function, and focus particularly on getting those predictions right (see here for a paper exploring a similar idea in the context of approximate inference).

One might object that this is only a problem in this toy setting; for instance, in the real world, one might look at the particular way in which someone is studying or performing some other boring task to see that it coherently leads towards some goal (in a way that would be less likely were the person to be doing something boring purely for enjoyment). In other words, correctly understanding the agent's goals might allow for more fine-grained accurate predictions which would fare better under e.g. log-score than would an incorrect model.

This is a reasonable objection, but there are some historical examples of this going wrong that should give one pause. That is, there are historical instances where: (i) people expected a more complex model that seemed to get at some underlying mechanism to outperform a simpler model that ignored that mechanism, and (ii) they were wrong (the simpler model did better under log-score). The example we are most familiar with is n-gram models vs. parse trees for language modelling; the most successful language models (in terms of having the best log-score on predicting the next word given a sequence of previous words) essentially treat language as a high-order Markov chain or hidden Markov model, despite the fact that linguistic theory predicts that language should be tree-structured rather than linearly-structured. Indeed, NLP researchers have tried building language models that assume language is tree-structured, and these models perform worse, or at least do not seem to have been adopted in practice (this is true both for older discrete models and newer continuous models based on neural nets). It's plausible that a similar issue will occur in inverse reinforcement learning, where correctly inferring plans is not enough to win out in predictive performance. The reason for the two issues might be quite similar (in language modelling, the tree structure only wins out in statistically uncommon corner cases involving long-term and/or nested dependencies, and hence getting that part of the prediction correct doesn't help predictive accuracy much).

The overall point is: in the case of even slight model mis-specification, the "correct" model might actually perform worse under typical metrics such as predictive accuracy. Therefore, more careful methods of constructing a model might be necessary.

Learning Values != Robustly Predicting Human Behaviour

The problems with IRL described so far will result in poor performance for predicting human choices out-of-sample. For example, if someone is observed doing boring tasks for 99 days (where they only achieve the goal on Day 100), they'll be predicted to continue doing boring tasks even when a short-cut to the goal becomes available. So even if the goal is simply to predict human behaviour (not to infer human values), mis-specification leads to bad predictions on realistic out-of-sample scenarios.

Let's suppose that our goal is not to predict human behaviour but to create AI systems that promote and respect human values. These goals (predicting humans and building safe AI) are distinct. Here's an example that illustrates the difference. Consider a long-term smoker, Bob, who would continue smoking even if there were (counterfactually) a universally effective anti-smoking treatment. Maybe Bob is in denial about the health effects of smoking or Bob thinks he'll inevitably go back to smoking whatever happens. If an AI system were assisting Bob, we might expect it to avoid promoting his smoking habit (e.g. by not offering him cigarettes at random moments). This is not paternalism, where the AI system imposes someone else's values on Bob. The point is that even if Bob would continue smoking across many counterfactual scenarios this doesn't mean that he places value on smoking.

How do we choose between the theory that Bob values smoking and the theory that he does not (but smokes anyway because of the powerful addiction)? Humans choose between these theories based on our experience with addictive behaviours and our insights into people's preferences and values. This kind of insight can't easily be captured as formal assumptions about a model, or even as a criterion about counterfactual generalization. (The theory that Bob values smoking *does* make accurate predictions across a wide range of counterfactuals.) Because of this, learning human values from IRL has a more profound kind of model mis-specification than the examples in Jacob's previous post. Even in the limit of data generated from an infinite series of random counterfactual scenarios, standard IRL algorithms would not infer someone's true values.

Predicting human actions is neither necessary nor sufficient for learning human values. In what ways, then, are the two related? One such way stems from the premise that if someone spends more resources making a decision, the resulting decision tends to be more in keeping with their true values. For instance, someone might spend lots of time thinking about the decision, they might consult experts, or they might try out the different options in a trial period before they make the real decision. Various authors have thus suggested that people's choices under sufficient "reflection" act as a reliable indicator of their true values. Under this view, predicting a certain kind of behaviour (choices under reflection) is sufficient for learning human values. Paul Christiano has written about some proposals for doing this, though we will not discuss them here (the first link is for general AI systems while the second is for newsfeeds). In general, turning these ideas into algorithms that are tractable and learn safely remains a challenging problem.

Further reading

There is research on doing IRL for agents in POMDPs. Owain and collaborators explored the effects of limited information and cognitive biases on IRL: paper, paper, online book.

For many environments it will not be possible to *identify* the reward function from the observed trajectories. These identification problems are related to the mis-specification problems but are not the same thing. Active learning can help with identification (paper).

Paul Christiano raised many similar points about mis-specification in a post on his blog.

For a big-picture monograph on relations between human preferences, economic utility theory and welfare/well-being, see Hausman's "Preference, Value, Choice and Welfare".

Acknowledgments

Thanks to Sindy Li for reviewing a full draft of this post and providing many helpful comments. Thanks also to Michael Webb and Paul Christiano for doing the same on specific sections of the post.

Future directions for ambitious value learning

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

To recap the sequence so far:

- Ambitious value learning aims to infer a utility function that is safe to maximize, by looking at human behavior.
- However, since you only observe human behavior, you must be able to infer and account for the mistakes that humans make in order to exceed human performance. (If we don't exceed human performance, it's likely that we'll use unsafe techniques that do exceed human performance, due to economic incentives.)
- You might hope to infer both the mistake model (aka systematic human biases) *and* the utility function, and then throw away the mistake model and optimize the utility function. This cannot be done without additional assumptions.
- One potential assumption you could use would be to codify a specific mistake model. However, humans are sufficiently complicated that any such model would be wrong, leading to model misspecification. Model misspecification causes many problems in general, and is particularly thorny for value learning.

Despite these arguments, we could still hope to infer a broad utility function that is safe to optimize, either by sidestepping the formalism used so far, or by introducing additional assumptions. Often, it is clear that these methods would not find the true human utility function (assuming that such a thing exists), but they are worth pursuing anyway because they could find a utility function that is good enough.

This post provides pointers to approaches that are currently being pursued. Since these are active areas of research, I don't want to comment on how feasible they may or may not be -- it's hard to accurately assess the importance and quality of an idea that is being developed just from what is currently written down about that idea.

Assumptions about the mistake model. We could narrow down on the mistake model by making assumptions about it, that could let us avoid the impossibility result. This decision means that we're accepting the risk of misspecification -- but perhaps as long as the mistake model is not *too* misspecified, the outcome will still be good.

Learning the Preferences of Ignorant, Inconsistent Agents shows how to infer utility functions when you have an exact mistake model, such as "the human is a hyperbolic time discounter". (Learning the Preferences of Bounded Agents and the online book Modeling Agents with Probabilistic Programs cover similar ground.)

Inferring Reward Functions from Demonstrators with Unknown Biases takes this a step further by simultaneously learning the mistake model and the utility function, while making weaker assumptions on the mistake model than "the human is noisily optimal". Of course, it does still make assumptions, or it would fall prey to the impossibility result (in particular, it would be likely to infer the negative of the "true" utility function).

The structure of the planning algorithm. Avoiding the impossibility result requires us to distinguish between (planner, reward) pairs that lead to the same policy. One approach is to look at the internal structure of the planner (this corresponds to looking inside the brains of individual humans). I like this post as an introduction, but many of Stuart Armstrong's other posts are tackling some aspect of this problem. There is also work that aims to build a psychological model of what constitutes human values, and use that to infer values, described in more detail (with citations) in this comment.

Assumptions about the relation of behavior to preferences. One of the most perplexing parts of the impossibility theorem is that we can't distinguish between fully rational and fully anti-rational behavior, yet we humans seem to do this easily. Perhaps this is because we have built-in priors that relate observations of behavior to preferences, which we could impart to our AI systems. For example, we could encode the assumption that regret is bad, or that lying about values is similar to lying about facts.

From the perspective of the sequence so far, both things we say and things we do count as "human behavior". But perhaps we could add in an assumption that inferences from speech and inferences from actions should mostly agree, and have rules about what to do if they don't agree. While there is a lot of work that uses natural language to guide some other learning process, I don't know of any work that tries to resolve conflicts between speech and actions (or *multimodal input* more generally), but it's something that I'm optimistic about. Acknowledging Human Preference Types to Support Value Learning explores this problem in more detail, suggesting some aggregation rules, but doesn't test any of these rules on real problems.

Other schemes for learning utility functions. One could imagine particular ways that value learning could go which would result in learning a good utility function. These cases typically can be recast as making some assumption about the mistake model.

For example, this comment proposes that the AI first asks humans how they would like their life to be while they figure out their utility function, and then uses that information to compute a distribution of "preferred" lives from which it learns the full utility function. The rest of the thread is a good example of applying the "mistake model" way of thinking to a proposal that does not obviously fit in its framework. There has been much more thinking spread across many posts and comment threads in a similar vein that I haven't collected, but you might be able to find some of it by looking at discussions between Paul Christiano and Wei Dai.

Resolving human values, completely and adequately presents another framework that aims for an adequate utility function instead of a perfect one.

Besides the approaches above, which still seek to infer a single utility function, there are a few other related approaches:

Tolerating a mildly misspecified utility function. The ideas of satisficing and mild optimization are trying to make us more robust to a misspecified utility function, by reducing how much we optimize the utility function. The key example of this is quantilizers, which select an action randomly from the top N% of actions from some distribution, sorted by expected utility.

Uncertainty over utility functions. Much work in value learning involves uncertainty over utility functions. This does not fix the issues presented so far -- we can consider what would happen if the AI updated on all possible information about the utility function. At that point, the AI would take the expectation of the resulting distribution, and maximize that function. This means that we once again end up with the AI optimizing a single function, and all of the same problems arise.

To be clear, most researchers do not think that uncertainty is a solution to these problems -- uncertainty can be helpful for other reasons, which I talk about later in the sequence. I mention this area of work because it works in the same framework of an AI optimizing a utility function, and I suspect many people will automatically associate uncertainty with any kind of value learning since CHAI has typically worked on both, but uncertainty is typically *not* targeting the problem of learning a utility function that is safe to maximize.

Intuitions about goal-directed behavior

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

One broad argument for AI risk is the Misspecified Goal argument:

> **The Misspecified Goal Argument for AI Risk:** Very intelligent AI systems will be able to make long-term plans in order to achieve their goals, and if their goals are even slightly misspecified then the AI system will become adversarial and work against us.

My main goal in this post is to make conceptual clarifications and suggest how they affect the Misspecified Goal argument, without making any recommendations about what we should actually do. Future posts will argue more directly for a particular position. As a result, I will not be considering other arguments for focusing on AI risk even though I find some of them more compelling.

I think of this as a concern about *long-term goal-directed behavior*. Unfortunately, it's not clear how to categorize behavior as goal-directed vs. not. Intuitively, any agent that searches over actions and chooses the one that best achieves some measure of "goodness" is goal-directed (though there are exceptions, such as the agent that selects actions that begin with the letter "A"). (ETA: I also think that agents that show goal-directed behavior because they are looking at some other agent are not goal-directed themselves -- see this comment.) However, this is not a necessary condition: many humans are goal-directed, but there is no goal baked into the brain that they are using to choose actions.

This is related to the concept of optimization, though with intuitions around optimization we typically assume that we know the agent's preference ordering, which I don't want to assume here. (In fact, I don't want to assume that the agent even *has* a preference ordering.)

One potential formalization is to say that goal-directed behavior is any behavior that can be modelled as maximizing expected utility for some utility function; in the next post I will argue that this does not properly capture the behaviors we are worried about. In this post I'll give some intuitions about what "goal-directed behavior" means, and how these intuitions relate to the Misspecified Goal argument.

Generalization to novel circumstances

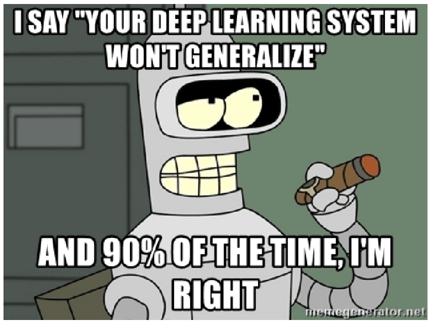
Consider two possible agents for playing some game, let's say TicTacToe. The first agent looks at the state and the rules of the game, and uses the minimax algorithm to find the optimal move to play. The second agent has a giant lookup table that tells it what move to play given any state. Intuitively, the first one is more "agentic" or "goal-driven", while the second one is not. But both of these agents play the game in exactly the same way!

The difference is in how the two agents *generalize to new situations*. Let’s suppose that we suddenly change the rules of TicTacToe -- perhaps now the win condition is reversed, so that anyone who gets three in a row loses. The minimax agent is still going to be optimal at this game, whereas the lookup-table agent will lose against any opponent with half a brain. The minimax agent looks like it is “trying to win”, while the lookup-table agent does not. (You could say that the lookup-table agent is “trying to take actions according to <policy>”, but this is a weird complicated goal so maybe it doesn’t count.)

In general, when we say that an agent is pursuing some goal, this is meant to allow us to predict how the agent will generalize to some novel circumstance. This sort of reasoning is critical for the Goal-Directed argument for AI risk. For example, we worry that an AI agent will prevent us from turning it off, because that would prevent it from achieving its goal: “You can't fetch the coffee if you're dead.” This is a prediction about what an AI agent would do in the novel circumstance where a human is trying to turn the agent off.

This suggests a way to characterize these sorts of goal-directed agents: there is some goal such that the agent’s behavior *in new circumstances* can be predicted by figuring out which behavior best achieves the goal. There's a lot of complexity in the space of goals we consider: something like "human well-being" should count, but "the particular policy <x>" and “pick actions that start with the letter A” should not. When I use the word goal I mean to include only the first kind, even though I currently don’t know theoretically how to distinguish between the various cases.

Note that this is in stark contrast to existing AI systems, which are particularly bad at generalizing to new situations.


I SAY "YOUR DEEP LEARNING SYSTEM WON'T GENERALIZE"
AND 90% OF THE TIME, I'M RIGHT
memegenerator.net

Honestly, I’m surprised it’s only 90%. [1]

Empowerment

We could also look at whether or not the agent acquires more power and resources. It seems likely that an agent that is optimizing for some goal over the long term would want more power and resources in order to more easily achieve that goal. In addition, the agent would probably try to improve its own algorithms in order to become more intelligent.

This feels like a *consequence* of goal-directed behavior, and not its defining characteristic, because it is about being able to achieve a *wide variety* of goals, instead of a particular one. Nonetheless, it seems crucial to the broad argument for AI risk presented above, since an AI system will probably need to first accumulate power, resources, intelligence, etc. in order to cause catastrophic outcomes.

I find this concept most useful when thinking about the problem of inner optimizers, where in the course of optimization through a rich space you stumble across a member of the space that is itself doing optimization, but for a related but still misspecified metric. Since the inner optimizer is being “controlled” by the outer optimization process, it is probably not going to cause major harm unless it is able to “take over” the outer optimization process, which sounds a lot like accumulating power. (This discussion is extremely imprecise and vague; see Risks from Learned Optimization for a more thorough discussion.)

Our understanding of the behavior

There is a general pattern in which as soon as we understand something, it becomes something lesser. As soon as we understand rainbows, they are relegated to the “dull catalogue of common things”. This suggests a somewhat cynical explanation of our concept of “intelligence”: an agent is considered intelligent if we do not know how to achieve the outcomes it does using the resources that it has (in which case our best model for that agent may be that it is pursuing some goal, reflecting our tendency to anthropomorphize). That is, our evaluation about intelligence is a statement about our epistemic state. Some examples that follow this pattern are:

- As soon as we understand how some AI technique solves a challenging problem, it is no longer considered AI. Before we’ve solved the problem, we imagine that we need some sort of “intelligence” that is pointed towards the goal and solves it: the only method we have of predicting what this AI system will do is to think about what a system that tries to achieve the goal would do. Once we understand how the AI technique works, we have more insight into what it is doing and can make more detailed predictions about where it will work well, where it tends to make mistakes, etc. and so it no longer seems like “intelligence”. Once you know that OpenAI Five is trained by self-play, you can predict that they haven’t seen certain behaviors like standing still to turn invisible, and probably won’t work well there.
- Before we understood the idea of natural selection and evolution, we would look at the complexity of nature and ascribe it to intelligent design; once we had the mathematics (and even just the qualitative insight), we could make much more detailed predictions, and nature no longer seemed like it required intelligence. For example, we can predict the timescales on which we can expect evolutionary changes, which we couldn’t do if we just modeled evolution as optimizing reproductive fitness.
- Many phenomena (eg. rain, wind) that we now have scientific explanations for were previously explained to be the result of some anthropomorphic deity.
- When someone performs a feat of mental math, or can tell you instantly what day of the week a random date falls on, you might be impressed and think them very intelligent. But if they explain to you how they did it, you may find it much less impressive. (Though of course these feats are selected to seem more impressive than they are.)

Note that an alternative hypothesis is that humans equate intelligence with mystery; as we learn more and remove mystery around eg. evolution, we automatically think of it as less intelligent.

To the extent that the Misspecified Goal argument relies on this intuition, the argument feels a lot weaker to me. If the Misspecified Goal argument rested entirely upon this intuition, then it would be asserting that *because* we are ignorant about what an intelligent agent would do, we should assume that it is optimizing a goal, which means that it is going to accumulate power and resources and lead to catastrophe. In other words, it is arguing that assuming that an agent is intelligent *definitionally* means that it will accumulate power and resources. This seems clearly wrong; it is possible in principle to have an intelligent agent that nonetheless does not accumulate power and resources.

Also, the argument is *not* saying that *in practice* most intelligent agents accumulate power and resources. It says that we have no better model to go off of other than “goal-directed”, and then pushes this model to extreme scenarios where we should have a lot more uncertainty.

To be clear, I do *not* think that anyone would endorse the argument as stated. I am suggesting as a possibility that the Misspecified Goal argument relies on us incorrectly equating superintelligence with “pursuing a goal” because we use “pursuing a goal” as a default model for anything that can do interesting things, even if that is not the best model to be using.

Summary

Intuitively, goal-directed behavior can lead to catastrophic outcomes with a sufficiently intelligent agent, because the optimal behavior for even a slightly misspecified goal can be very bad according to the true goal. However, it’s not clear exactly what we mean by goal-directed behavior. Often, an algorithm that searches over possible actions and chooses the one with the highest “goodness” will be goal-directed, but this is neither necessary nor sufficient.

“From the outside”, it seems like a goal-directed agent is characterized by the fact that we can predict the agent’s behavior in new situations by assuming that it is pursuing some goal, and as a result it is acquires power and resources. This can be interpreted either as a statement about our epistemic state (we know so little about the agent that our best model is that it pursues a goal, even though this model is not very accurate or precise) or as a statement about the agent (predicting the behavior of the agent in new situations based on pursuit of a goal actually has very high precision and accuracy). These two views have very different implications on the validity of the Misspecified Goal argument for AI risk.

---

[1] This is an entirely made-up number.

Coherence arguments do not entail goal-directed behavior

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

One of the most pleasing things about probability and expected utility theory is that there are many *coherence arguments* that suggest that these are the “correct” ways to reason. If you deviate from what the theory prescribes, then you must be executing a *dominated strategy*. There must be some other strategy that never does any worse than your strategy, but does strictly better than your strategy with certainty in at least one situation. There’s a good explanation of these arguments here.

We shouldn’t expect mere humans to be able to notice any failures of coherence in a superintelligent agent, since if we could notice these failures, so could the agent. So we should expect that powerful agents appear coherent to us. (Note that it is possible that the agent doesn’t fix the failures because it would not be worth it -- in this case, the argument says that we will not be able to notice any *exploitable* failures.)

Taken together, these arguments suggest that we should model an agent much smarter than us as an expected utility (EU) maximizer. And many people agree that EU maximizers are dangerous. So does this mean we’re doomed? I don’t think so: it seems to me that the problems about EU maximizers that we’ve identified are actually about *goal-directed behavior* or *explicit reward maximizers.* The coherence theorems say nothing about whether an AI system must look like one of these categories. This suggests that we could try building an AI system that can be modeled as an EU maximizer, yet doesn’t fall into one of these two categories, and so doesn’t have all of the problems that we worry about.

Note that there are two different flavors of arguments that the AI systems we build will be goal-directed agents (which are dangerous if the goal is even slightly wrong):

- Simply knowing that an agent is intelligent lets us infer that it is goal-directed. (EDIT: See these comments for more details on this argument.)
- Humans are particularly likely to build goal-directed agents.

I will only be arguing against the first claim in this post, and will talk about the second claim in the next post.

All behavior can be rationalized as EU maximization

Suppose we have access to the entire policy of an agent, that is, given any universe-history, we know what action the agent will take. Can we tell whether the agent is an EU maximizer?

Actually, *no matter what the policy is*, we can view the agent as an EU maximizer. The construction is simple: the agent can be thought as optimizing the utility function U, where U(h, a) = 1 if the policy would take action a given history h, else 0. Here I’m assuming that U is defined over histories that are composed of states/observations and actions. The actual policy gets 1 utility at every timestep; any other policy gets less than this, so the given policy perfectly maximizes this utility function. This construction has been given before, eg. at the bottom of page 6 of this paper. (I think I've seen it before too, but I can't remember where.)

But wouldn't this suggest that the VNM theorem has no content? Well, we assumed that we were looking at the *policy* of the agent, which led to a universe-history *deterministically*. We didn't have access to any probabilities. Given a particular action, we knew exactly what the next state would be. Most of the axioms of the VNM theorem make reference to lotteries and probabilities -- if the world is deterministic, then the axioms simply say that the agent must have transitive preferences over outcomes. Given that we can only observe the agent choose one history over another, we can trivially construct a transitive preference ordering by saying that the chosen history is higher in the preference ordering than the one that was not chosen. This is essentially the construction we gave above.

What then is the purpose of the VNM theorem? It tells you how to behave *if you have probabilistic beliefs about the world*, as well as a *complete and consistent preference ordering over outcomes*. This turns out to be not very interesting when "outcomes" refers to "universe-histories". It can be more interesting when "outcomes" refers to world *states* instead (that is, snapshots of what the world looks like at a particular time), but utility functions over states/snapshots can't capture everything we're interested in, and there's no reason to take as an assumption that an AI system will have a utility function over states/snapshots.

There are no coherence arguments that say you must have goal-directed behavior

Not all behavior can be thought of as goal-directed (primarily because I allowed the category to be defined by fuzzy intuitions rather than something more formal). Consider the following examples:

- A robot that constantly twitches
- The agent that always chooses the action that starts with the letter "A"
- The agent that follows the policy <policy> where for every history the corresponding action in <policy> is generated randomly.

These are not goal-directed by my "definition". However, they can all be modeled as expected utility maximizers, and there isn't any particular way that you can exploit any of these agents. Indeed, it seems hard to model the twitching robot or the policy-following agent as having any preferences at all, so the notion of "exploiting" them doesn't make much sense.

You could argue that neither of these agents are *intelligent*, and we're only concerned with superintelligent AI systems. I don't see why these agents could not in principle be intelligent: perhaps the agent knows how the world would evolve, and how to intervene on the world to achieve different outcomes, but it does not act on these beliefs. Perhaps if we peered into the inner workings of the agent, we could find some part of it that allows us to predict the future very accurately, but it turns out that these inner workings did not affect the chosen action at all. Such an agent is in principle possible, and it seems like it is intelligent.

(If not, it seems as though you are *defining* intelligence to also be goal-driven, in which case I would frame my next post as arguing that we may not want to build superintelligent AI, because there are other things we could build that are as useful without the corresponding risks.)

You could argue that while this is possible in principle, no one would ever build such an agent. I wholeheartedly agree, but note that this is now an argument based on particular empirical facts about humans (or perhaps agent-building processes more generally). I'll talk about those in the next post; here I am simply arguing that merely knowing that an agent is intelligent, with no additional empirical facts about the world, does not let you infer that it has goals.

As a corollary, since all behavior can be modeled as maximizing expected utility, but not all behavior is goal-directed, it is not possible to conclude that an agent is goal-driven if you only know that it can be modeled as maximizing some expected utility. However, if you know that an agent is maximizing the expectation of an *explicitly represented* utility function, I would expect that to lead to goal-driven behavior most of the time, since the utility function must be relatively simple if it is explicitly represented, and *simple* utility functions seem particularly likely to lead to goal-directed behavior.

There are no coherence arguments that say you must have preferences

This section is another way to view the argument in the previous section, with "goal-directed behavior" now being operationalized as "preferences"; it is not saying anything new.

Above, I said that the VNM theorem assumes both that you use probabilities and that you have a preference ordering over outcomes. There are lots of good reasons to assume that a good reasoner will use probability theory. However, there's not much reason to assume that there is a preference ordering over outcomes. The twitching robot, "A"-following agent, and random policy agent from the last section all seem like they don't have preferences (in the English sense, not the math sense).

Perhaps you could define a preference ordering by saying "if I gave the agent lots of time to think, how would it choose between these two histories?" However, you could apply this definition to *anything*, including eg. a thermostat, or a rock. You might argue that a thermostat or rock can't "choose" between two histories; but then it's unclear how to define how an AI "chooses" between two histories without that definition also applying to thermostats and rocks.

Of course, you could always define a preference ordering based on the AI's observed behavior, but then you're back in the setting of the first section, where *all* observed behavior can be modeled as maximizing an expected utility function and so saying "the AI is an expected utility maximizer" is vacuous.

Convergent instrumental subgoals are about goal-directed behavior

One of the classic reasons to worry about expected utility maximizers is the presence of convergent instrumental subgoals, detailed in Omohundro's paper The Basic AI Drives. The paper itself is clearly talking about goal-directed AI systems:

To say that a system of any design is an "artificial intelligence", we mean that it has goals which it tries to accomplish by acting in the world.

It then argues (among other things) that such AI systems will want to "be rational" and so will distill their goals into utility functions, which they then maximize. And once they have utility functions, they will protect them from modification.

Note that this starts from the assumption of goal-directed behavior and *derives* that the AI will be an EU maximizer along with the other convergent instrumental subgoals. The coherence arguments all imply that AIs will be EU maximizers for some (possibly degenerate) utility function; they don't prove that the AI must be goal-directed.

Goodhart's Law is about goal-directed behavior

A common argument for worrying about AI risk is that we know that a superintelligent AI system will look to us like an EU maximizer, and if it maximizes a utility function that is even slightly wrong we could get catastrophic outcomes.

By now you probably know my first response: that *any* behavior can be modeled as an EU maximizer, and so this argument proves too much, suggesting that any behavior causes catastrophic outcomes. But let's set that aside for now.

The second part of the claim comes from arguments like Value is Fragile and Goodhart's Law. However, if we consider utility functions that assign value 1 to some histories and 0 to others, then if you accidentally assign a history where I needlessly stub my toe a 1 instead of a 0, that's a slightly wrong utility function, but it isn't going to lead to catastrophic outcomes.

The worry about utility functions that are *slightly wrong* holds water when the utility functions are wrong about some *high-level* concept, like whether humans care about their experiences reflecting reality. This is a very rarefied, particular distribution of utility functions, that are all going to lead to goal-directed or agentic behavior. As a result, I think that the argument is better stated as "if you have a slightly incorrect goal, you can get catastrophic outcomes". And there aren't any coherence arguments that say that agents must have goals.

Wireheading is about explicit reward maximization

There are a few papers that talk about the problems that arise with a very powerful system with a reward function or utility function, most notably wireheading. The argument that AIXI will seize control of its reward channel falls into this category. In these cases, typically the AI system is considering making a change to the system by which it evaluates goodness of actions, and the goodness of the change is evaluated by the system *after the change*. Daniel Dewey argues in Learning What to Value that if the change is evaluated by the system *before* the change, then these problems go away.

I think of these as problems with *reward* maximization, because typically when you phrase the problem as maximizing reward, you are maximizing the sum of rewards obtained in all timesteps, no matter how those rewards are obtained (i.e. even if you self-modify to make the reward maximal). It doesn’t seem like AI systems have to be built this way (though admittedly I do not know how to build AI systems that reliably avoid these problems).

Summary

In this post I’ve argued that many of the problems we typically associate with expected utility maximizers are actually problems with goal-directed agents or with explicit reward maximization. Coherence arguments only entail that a superintelligent AI system will look like an expected utility maximizer, but this is actually a vacuous constraint, and there are many potential utility functions for which the resulting AI system is neither goal-directed nor explicit-reward-maximizing. This suggests that we could try to build AI systems of this type, in order to sidestep the problems that we have identified so far.

Will humans build goal-directed agents?

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

In the previous post, I argued that simply knowing that an AI system is superintelligent does not imply that it must be goal-directed. However, there are many other arguments that suggest that AI systems will or should be goal-directed, which I will discuss in this post.

Note that I don’t think of this as the Tool AI vs. Agent AI argument: it seems possible to build agent AI systems that are not goal-directed. For example, imitation learning allows you to create an agent that behaves similarly to another agent -- I would classify this as “Agent AI that is not goal-directed”. (But see this comment thread for discussion.)

Note that these arguments have different implications than the argument that superintelligent AI must be goal-directed due to coherence arguments. Suppose you believe all of the following:

- Any of the arguments in this post.
- Superintelligent AI is not *required* to be goal-directed, as I argued in the last post.
- Goal-directed agents cause catastrophe by default.

Then you could try to create alternative designs for AI systems such that they can do the things that goal-directed agents can do without themselves being goal-directed. You could also try to persuade AI researchers of these facts, so that they don’t build goal-directed systems.

Economic efficiency: goal-directed humans

Humans want to build powerful AI systems in order to help them achieve their goals -- it seems quite clear that humans are at least partially goal-directed. As a result, it seems natural that they would build AI systems that are also goal-directed.

This is really an argument that the *system* comprising the human and AI agent should be directed towards some goal. The AI agent by itself need not be goal-directed as long as we get goal-directed behavior when combined with a human operator. However, in the situation where the AI agent is much more intelligent than the human, it is probably best to delegate most or all decisions to the agent, and so the agent could still look mostly goal-directed.

Even so, you could imagine that even the small part of the work that the human continues to do allows the agent to not be goal-directed, especially over long horizons. For example, perhaps the human decides what the agent should do each day, and the agent executes the instruction, which involves planning over the course of a day, but no longer. (I am *not* arguing that this is safe; on the contrary, having very powerful optimization over the course of a day seems probably unsafe.) This could be extremely powerful without the AI being goal-directed over the long term.

Another example would be a corrigible agent, which could be extremely powerful while not being goal-directed over the long term. (Though the meanings of “goal-directed” and “corrigible” are sufficiently fuzzy that this is not obvious and depends on the definitions we settle on for each.)

Economic efficiency: beyond human performance

Another benefit of goal-directed behavior is that it allows us to find novel ways of achieving our goals that we may not have thought of, such as AlphaGo’s move 37. Goal-directed behavior is one of the few methods we know of that allow AI systems to exceed human performance.

I think this is a good argument for goal-directed behavior, but given the problems of goal-directed behavior I think it’s worth searching for alternatives, such as the two examples in the previous section (optimizing over a day, and corrigibility). Alternatively, we could learn human reasoning, and execute it for a longer subjective time than humans would, in order to make better decisions. Or we could have systems that remain uncertain about the goal and clarify what they should do when there are multiple very different options (though this has its own problems).

Current progress in reinforcement learning

If we had to guess today which paradigm would lead to AI systems that can exceed human performance, I would guess reinforcement learning (RL). In RL, we have a reward function and we seek to choose actions that maximize the sum of expected discounted rewards. This sounds a lot like an agent that is searching over actions for the best one according to a measure of goodness (the reward function [1]), which I said previously is a goal-directed agent. And the math behind RL says that the agent should be trying to maximize its reward for the rest of time, which makes it long-term [2].

That said, current RL agents learn to replay behavior that in their past experience worked well, and typically do not generalize outside of the training distribution. This does not seem like a search over actions to find ones that are the best. In particular, you shouldn’t expect a treacherous turn, since the whole point of a treacherous turn is that you don’t see it coming because it never happened before.

In addition, current RL is episodic, so we should only expect that RL agents are goal-directed *over the current episode* and not in the long-term. Of course, many tasks would have very long episodes, such as being a CEO. The vanilla deep RL approach here would be to specify a reward function for how good a CEO you are, and then try many different ways of being a CEO and learn from experience. This requires you to collect many full episodes of being a CEO, which would be extremely time-consuming.

Perhaps with enough advances in model-based deep RL we could train the model on partial trajectories and that would be enough, since it could generalize to full trajectories. I think this is a tenable position, though I personally don’t expect it to work since it relies on our model generalizing well, which seems unlikely even with future research.

These arguments lead me to believe that we’ll probably have to do something that is not vanilla deep RL in order to train an AI system that can be a CEO, and that thing may not be goal-directed.

Overall, it is certainly possible that improved RL agents will look like dangerous long-term goal-directed agents, but this does not seem to be the case today and there seem to be serious difficulties in scaling current algorithms to superintelligent AI systems that can optimize over the long term. (I'm not arguing for long timelines here, since I wouldn't be surprised if we figured out some way that *wasn't* vanilla deep RL to optimize over the long term, but that method need not be goal-directed.)

Existing intelligent agents are goal-directed

So far, humans and perhaps animals are the only example of generally intelligent agents that we know of, and they seem to be quite goal-directed. This is some evidence that we should expect intelligent agents that we build to also be goal-directed.

Ultimately we are observing a correlation between two things with sample size 1, which is really not much evidence at all. If you believe that many animals are also intelligent and goal-directed, then perhaps the sample size is larger, since there are intelligent animals with very different evolutionary histories and neural architectures (eg. octopuses).

However, this is specifically about agents that were created by evolution, which did a relatively stupid blind search over a large space, and we could use a different method to develop AI systems. So this argument makes me more wary of creating AI systems using evolutionary searches over large spaces, but it doesn't make me much more confident that all good AI systems must be goal-directed.

Interpretability

Another argument for building a goal-directed agent is that it allows us to predict what it's going to do in novel circumstances. While you may not be able to predict the specific actions it will take, you can predict some features of the final world state, in the same way that if I were to play Magnus Carlsen at chess, I can't predict how he will play, but I can predict that he will win.

I do not understand the intent behind this argument. It seems as though faced with the negative results that suggest that goal-directed behavior tends to cause catastrophic outcomes, we're arguing that it's a good idea to build a goal-directed agent so that we can more easily predict that it's going to cause catastrophe.

I also think that we would typically be able to predict significantly *more* about what any AI system we actually build will do (than if we modeled it as trying to achieve some goal). This is because "agent seeking a particular goal" is one of the simplest models we can build, and with any system we have more information on, we start refining the model to make it better.

Summary

Overall, I think there are good reasons to think that "by default" we would develop goal-directed AI systems, because the things we want AIs to do can be easily phrased as goals, and because the stated goal of reinforcement learning is to build goal-directed agents (although they do not look like goal-directed agents today). As a result, it seems important to figure out ways to get the powerful capabilities of goal-directed agents through agents that are not themselves goal-directed. In particular, this suggests that we will need to figure out ways to build AI systems that do not involve specifying a utility function that the AI should optimize, or even learning a utility function that the AI then optimizes.

[1] Technically, actions are chosen according to the Q function, but the distinction isn’t important here.

[2] Discounting does cause us to prioritize short-term rewards over long-term ones. On the other hand, discounting seems mostly like a hack to make the math not spit out infinities, and so that learning is more stable. On the third hand, infinite horizon MDPs with undiscounted reward aren't solvable unless you almost surely enter an absorbing state. So discounting complicates the picture, but not in a particularly interesting way, and I don’t want to rest an argument against long-term goal-directed behavior on the presence of discounting.

AI safety without goal-directed behavior

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

When I first entered the field of AI safety, I thought of the problem as figuring out how to get the AI to have the “right” utility function. This led me to work on the problem of inferring values from demonstrators with unknown biases, despite the impossibility results in the area. I am less excited about that avenue because I am pessimistic about the prospects of ambitious value learning (for the reasons given in the first part of this sequence).

I think this happened because the writing on AI risk that I encountered has the pervasive assumption that any superintelligent AI agent must be maximizing some utility function over the long term future, such that it leads to goal-directed behavior and convergent instrumental subgoals. It’s often not stated as an assumption; rather, inferences are made assuming that you have the background model that the AI is goal-directed. This makes it particularly hard to question the assumption, since you don’t realize that the assumption is even there.

Another reason that this assumption is so easily accepted is that we have a long history of modeling rational agents as expected utility maximizers, and for good reason: there are many coherence arguments that say that, *given that you have preferences/goals*, if you aren’t using probability theory and expected utility theory, then you can be taken advantage of. It’s easy to make the inference that a superintelligent agent must be rational, and therefore it must be an expected utility maximizer.

Because this assumption was so embedded in how I thought about the problem, I had trouble imagining how else to even consider the problem. I would guess this is true for at least some other people, so I want to summarize the counterargument, and list a few implications, in the hope that this makes the issue clearer.

Why goal-directed behavior may not be required

The main argument of this chapter is that it is not *required* that a superintelligent agent takes actions in pursuit of some goal. It is possible to write algorithms that select actions without doing a search over the actions and rating their consequences according to an explicitly specified simple function. There is no coherence argument that says that your agent must have preferences or goals; it is perfectly possible for the agent to take actions with no goal in mind simply because it was programmed to do so; this remains true even when the agent is intelligent.

It seems quite likely that by default a superintelligent AI system would be goal-directed anyway, because of economic efficiency arguments. However, this is *not* set in stone, as it would be if coherence arguments implied goal-directed behavior. Given the negative results around goal-directed behavior, it seems like the natural path forward is to search for alternatives that still allow us to get economic efficiency.

Implications

At a high level, I think that the main implication of this view is that we should be considering other models for future AI systems besides optimizing over the long term for a single goal or for a particular utility or reward function. Here are some other potential models:

- G*oal-conditioned policy with common sense:* In this setting, humans can set goals for the AI system simply by asking it in natural language to do something, and the AI system sets out to do it. However, the AI also has “common sense”, where it interprets our commands pragmatically and not literally: it’s not going to prevent us from setting a new goal (which would stop it from achieving its current goal), because common sense tells it that we don’t want it to do that. One way to think about this is to consider an AI system that infers and follows human *norms*, which are probably much easier to infer than human values (most humans seem to infer norms very accurately).
- *Corrigible AI:* I’ll defer to Paul Christiano’s explanation of corrigibility.
- *Comprehensive AI Services (CAIS):* Maybe we could create lots of AI services that interact with each other to solve hard problems. Each individual service could be bounded and episodic, which immediately means that it is no longer optimizing over the long term (though it could still be goal-directed). Perhaps we have a long-term planner that is trained to produce good plans to achieve particular goals over the span of an hour, and a plan executor that takes in a plan and executes the next step of the plan over an hour, and leaves instructions for the next steps.

There are versions of these scenarios which are compatible with the framework of an AI system optimizing for a single goal:

- A goal-conditioned policy with common sense could be operationalized as optimizing for the goal of “following a human’s orders without doing anything that humans would reliably judge as crazy”.
- MIRI’s version of corrigibility seems like it stays within this framework.
- You could think of the services in CAIS as optimizing for the *aggregate* reward they get over all time, rather than just the reward they get during the current episode.

I do *not* want these versions of the scenarios, since they then make it tempting to once again say “but if you get the goal even slightly wrong, then you’re in big trouble”. This would likely be true if we built an AI system that could maximize an arbitrary function, and then tried to program in the utility function we care about, but *this is not required*. It seems possible to build systems in such a way that these properties are inherent in the way that they reason, such that it’s not even coherent to ask what happens if we “get the utility function slightly wrong”.

Note that I’m not claiming that I know how to build such systems; I’m just claiming that we don’t know enough yet to reject the possibility that we could build such systems. Given how hard it seems to be to align systems that explicitly maximize a reward function, we should explore these other methods as well.

Once we let go of the idea of optimizing for a single goal and it becomes possible to think about other ways in which we could build AI systems, there are more insights about how we could build an AI system that does what we intend instead of what we say. (In my case it was reversed -- I heard a lot of good insights that don’t fit in the framework of goal-directed optimization, and this eventually led me to let go of the assumption of goal-directed optimization.) We’ll explore some of these in the next chapter.

What is narrow value learning?

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

Ambitious value learning aims to achieve superhuman performance by figuring out the underlying latent "values" that humans have, and evaluating new situations according to these values. In other words, it is trying to infer the criteria by which we judge situations to be good. This is particularly hard because in novel situations that humans haven't seen yet, we haven't even developed the criteria by which we would evaluate. (This is one of the reasons why we need to model humans as suboptimal, which causes problems.)

Instead of this, we can use *narrow value learning*, which produces behavior that we want in some narrow domain, without expecting generalization to novel circumstances. The simplest form of this is imitation learning, where the AI system simply tries to imitate the supervisor's behavior. This limits the AI’s performance to that of its supervisor. We could also learn from preferences over behavior, which can scale to superhuman performance, since the supervisor can often evaluate whether a particular behavior meets our preferences even if she can’t perform it herself. We could also teach our AI systems to perform tasks that we would not want to do ourselves, such as handling hot objects.

Nearly all of the work on preference learning, including most work on inverse reinforcement learning (IRL), is aimed at narrow value learning. IRL is often explicitly stated to be a technique for imitation learning, and early algorithms phrase the problem as *matching* the features in the demonstration, not exceeding them. The few algorithms that try to generalize to different test distributions, such as AIRL, are only aiming for relatively small amounts of generalization.

(Why use IRL instead of behavioral cloning, where you mimic the actions that the demonstrator took? The hope is that IRL gives you a good inductive bias for imitation, allowing you to be more sample efficient and to generalize a little bit.)

You might have noticed that I talk about narrow value learning in terms of actual observed behavior from the AI system, as opposed to any sort of “preferences” or “values” that are inferred. This is because I want to include approaches like imitation learning, or meta learning for quick task identification and performance. These approaches can produce behavior that we want without having an explicit representation of “preferences”. In practice any method that scales to human intelligence is going to have to infer preferences, though perhaps implicitly.

Since any instance of narrow value learning is defined with respect to some domain or input distribution on which it gives sensible results, we can rank them according to how general this input distribution is. An algorithm that figures out what food I like to eat is very domain-specific, whereas one that determines my life goals and successfully helps me achieve them in both the long and short term is very general. When the input distribution is “all possible inputs”, we have a system that has good behavior everywhere, reminiscent of ambitious value learning.

(Annoyingly, I defined ambitious value learning to be about the *definition* of optimal behavior, such as an inferred utility function, while narrow value learning is about the observed behavior. So really the most general version of narrow value learning is equivalent to “ambitious value learning plus some method of actually obtaining the defined behavior in practice, such as by using deep RL”.)

Ambitious vs. narrow value learning

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

(Re)Posted as part of the AI Alignment Forum sequence on Value Learning.

> ***Rohin's note:*** *The definition of narrow value learning in the previous post focused on the fact that the resulting behavior is limited to some domain. The definition in this post focuses on learning instrumental goals and values. While the definitions are different, I have used the same term for both because I believe that they are both pointing at the same underlying concept. (I do not know if Paul agrees.) I'm including this post to give a different perspective on what I mean by narrow value learning, before delving into conceptual ideas within narrow value learning.*

Suppose I'm trying to build an AI system that "learns what I want" and helps me get it. I think that people sometimes use different interpretations of this goal. At two extremes of a spectrum of possible interpretations:

- The AI learns my preferences over (very) long-term outcomes. If I were to die tomorrow, it could continue pursuing my goals without me; if humanity were to disappear tomorrow, it could rebuild the kind of civilization we would want; *etc.* The AI might pursue radically different subgoals than I would on the scale of months and years, if it thinks that those subgoals better achieve what I really want.
- The AI learns the narrower subgoals and instrumental values I am pursuing. It learns that I am trying to schedule an appointment for Tuesday and that I want to avoid inconveniencing anyone, or that I am trying to fix a particular bug without introducing new problems, *etc.* It does not make any effort to pursue wildly different short-term goals than I would in order to better realize my long-term values, though it may help me correct some errors that I would be able to recognize as such.

I think that many researchers interested in AI safety per se mostly think about the former. I think that researchers with a more practical orientation mostly think about the latter.

The ambitious approach

The maximally ambitious approach has a natural theoretical appeal, but it also seems quite hard. It requires understanding human preferences in domains where humans are typically very uncertain, and where our answers to simple questions are often inconsistent, like how we should balance our own welfare with the welfare of others, or what kinds of activities we really want to pursue vs. enjoying in the moment. (It seems unlikely to me that there is a unified notion of "what I want" in many of these cases.) It also requires extrapolation to radically unfamiliar domains, where we will need to make decisions about issues like population ethics, what kinds of creatures do we care about, and unforeseen new technologies.

I have written about this problem, pointing out that it is unclear how you would solve it even with an unlimited amount of computing power. My impression is that most practitioners don’t think of this problem even as a long-term research goal — it’s a qualitatively different project without direct relevance to the kinds of problems they want to solve.

The narrow approach

The narrow approach looks relatively tractable and well-motivated by existing problems. We want to build machines that helps us do the things we want to do, and to that end they need to be able to understand what we are trying to do and what instrumental values guide our behavior. To the extent that our “preferences” are underdetermined or inconsistent, we are happy if our systems at least do as well as a human, and make the kinds of improvements that humans would reliably consider improvements.

But it’s not clear that anything short of the maximally ambitious approach can solve the problem we ultimately care about. A sufficiently clever machine will be able to make long-term plans that are significantly better than human plans. In the long run, we will want to be able to use AI abilities to make these improved plans, and to generally perform tasks in ways that humans would never think of perform them — going far beyond correcting simple errors that can be easily recognized as such.

In defense of the narrow approach

I think that the narrow approach probably takes us much further than it at first appears. I’ve written about these arguments before, which are for the most part similar to the reasons that approval-directed agents or directly mimicking human behavior might work, but I’ll quickly summarize them again:

Instrumental goals

Humans have many clear instrumental goals like “remaining in effective control of the AI systems I deploy,” “acquiring resources and other influence in the world,” or “better understanding the world and what I want.” A value learner may able to learn robust preferences like these and pursue those instrumental goals using all of its ingenuity. Such AI’s would not necessarily be at a significant disadvantage with respect to normal competition, yet the resources they acquired would remain under meaningful human control (if that’s what their users would prefer).

This requires learning robust formulations of concepts like “meaningful control,” but it does not require making inferences about cases where humans have conflicting intuitions, nor considering cases which are radically different from those encountered in training — AI systems can continue to gather training data and query their users even as the nature of human-AI interactions changes (if that’s what their users would prefer).

Process

Even if we can’t infer human preferences over very distant objects, we might be able to infer human preferences well enough to guide a process of deliberation (real or hypothetical). Using the inferred preferences of the human could help eliminate some of the errors that a human would traditionally make during deliberation. Presumably these errors run counter to a deliberator’s short-term objectives, if those objectives are properly understood, and this judgment doesn’t require a direct understanding of the deliberator’s big-picture values.

This kind of error-correction could be used as a complement to other kinds of idealization, like providing the human a lot of time, allowing them to consult a large community of advisors, or allowing them to use automated tools.

Such a process of error-corrected deliberation could itself be used to provide a more robust definition of values or a more forward looking criterion of action, such as “an outcome/action is valuable to the extent that I would/did judge it valuable after extensive deliberation.”

Bootstrapping

By interacting with AI assistants, humans can potentially form and execute very sophisticated plans; if so, simply helping them achieve their short-term goals may be all that is needed. For some discussion of this idea, see these three posts.

Conclusion

I think that researchers interested in scalable AI control have been too quick to dismiss “narrow” value learning as unrelated to their core challenge. Overall I expect that the availability of effective narrow value learning would significantly simplify the AI control problem even for superintelligent systems, though at the moment we don’t understand the relationship very well.

(Thanks to Andreas Stuhlmüller and Owain Evans for helpful discussion.)

This was originally posted here on 4th October, 2015.

Human-AI Interaction

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

The importance of feedback

Consider trying to program a self-driving car to drive from San Francisco to Los Angeles -- *with no sensors* that allow it to gather information as it is driving. This is possible in principle. If you can predict the exact weather conditions, the exact movement of all of the other cars on the road, the exact amount of friction along every part of the road surface, the exact impact of (the equivalents of) pressing the gas or turning the steering wheel, and so on, then you could compute ahead of time how exactly to control the car such that it gets from SF to LA. Nevertheless, it seems unlikely that we will ever be able to accomplish such a feat, even with powerful AI systems.

No, in practice there is going to be some uncertainty about how the world is going to evolve; such that any plan computed ahead of time will have some errors that will compound over the course of the plan. The solution is to use sensors to gather information *while* executing the plan, so that we can notice any errors or deviations from the plan, and take corrective action. It is much easier to build a *controller* that keeps you pointed in the general direction, than to build a plan that will get you there perfectly without any adaptation.

Control theory studies these sorts of systems, and you can see the general power of feedback controllers in the theorems that can be proven. Especially for motion tasks, you can build feedback controllers that are guaranteed to safely achieve the goal, even in the presence of *adversarial* environmental forces (that are bounded in size, so you can’t have arbitrarily strong wind). In the presence of an adversary, in most environments it becomes impossible even in principle to make such a guarantee if you do not have any sensors or feedback and must compute a plan in advance. Typically, for every such plan, there is some environmental force that would cause it to fail.

The control theory perspective on AI alignment

With ambitious value learning, we’re hoping that we can learn a utility function that tells us the optimal thing to do into the future. You need to be able to encode exactly how to behave in all possible environments, no matter what new things happen in the future, even if it’s something we humans never considered a possibility so far.

This is analogous to the problem of trying to program a self-driving car. Just as in that case, we might hope that we can solve the problem by introducing sensors and feedback. In this case, the “feedback” would be human data that informs our AI system what we want it to do, that is, data that can be used to learn values. The evolution of human values and preferences in new environments with new technologies is analogous to the unpredictable environmental disturbances that control theory assumes.

This does not mean that an AI system must be architected in such a way that human data is *explicitly* used to “control” the AI every few timesteps in order to keep it on track. It does mean that any AI alignment proposal should have some method of incorporating information about what humans want in radically different circumstances. I have found this an important frame with which to view AI alignment proposals. For example, with indirect normativity or idealized humans it’s important that the idealized or simulated humans are going through similar experiences that real humans go through, so that they provide good feedback.

Feedback through interaction

Of course, while the control theory perspective does not require the feedback controller to be explicit, one good way to ensure that there is feedback would be to make it explicit. This would mean that we create an AI system that explicitly collects fresh data about what humans want in order to inform what it should do. This is basically calling for an AI system that is constantly using tools from narrow value learning to figure out what to do. In practice, this will require interaction between the AI and the human. However, there are still issues to think about:

Convergent instrumental subgoals: A simple way of implementing human-AI interaction would be to have an estimate of a reward function that is continually updated using narrow value learning. Whenever the AI needs to choose an action, it uses the current reward estimate to choose.

With this sort of setup, we still have the problem that we are maximizing a reward function which leads to convergent instrumental subgoals. In particular, the plan “disable the narrow value learning system” is likely very good according to the current estimate of the reward function, because it prevents the reward from changing causing all future actions to continue to optimize the current reward estimate.

Another way of seeing that this setup is a bit weird is that it has inconsistent preferences over time -- at any given point in time, it treats the expected change in its reward as an obstacle that should be undone if possible.

That said, it is worth noting that in this setup, the goal-directedness is coming from the human. In fact, any approach where goal-directedness comes from the human requires some form of human-AI interaction. We might hope that some system of this form allows us to have a human-AI system that is overall goal-directed (in order to achieve economic efficiency), while the AI system itself is not goal-directed, and so the overall system pursues the *human’s* instrumental subgoals. The next post will talk about reward uncertainty as a potential approach to get this behavior.

Humans are unable to give feedback: As our AI systems become more and more powerful, we might worry that they are able to vastly outthink us, such that they would need our feedback on scenarios that are too hard for us to comprehend.

On the one hand, if we’re actually in this scenario I feel quite optimistic: if the questions are so difficult that we can’t answer them, we’ve probably already solved all the simple parts of the reward, which means we’ve probably stopped x-risk.

But even if it is imperative that we answer these questions accurately, I’m still optimistic: as our AI systems become more powerful, we can have better AI-enabled tools that help us understand the questions on which we are supposed to give feedback. This could be AI systems that do cognitive work on our behalf, as in recursive reward modeling, or it could be AI-created technologies that make us more capable, such as brain enhancement or the ability to be uploaded and have bigger “brains” that can understand larger things.

Humans don’t know the goal: An important disanalogy between the control theory/self-driving car example and the AI alignment problem is that in control theory it is assumed that the general path to the destination is known, and we simply need to stay on it; whereas in AI alignment even the human does not know the goal (i.e. the “true human reward”). As a result, we cannot rely on humans to always provide adequate feedback; we also need to manage the process by which humans learn what they want. Concerns about human safety problems and manipulation fall into this bucket.

Summary

If I want an AI system that acts autonomously over a long period of time, but it isn't doing ambitious value learning (only narrow value learning), then we necessarily require a feedback mechanism that keeps the AI system "on track" (since my instrumental values will change over that period of time).

While the feedback mechanism need not be explicit (and could arise simply because it is an effective way to actually help me), we could consider AI designs that have an explicit feedback mechanism. There are still many problems with such a design, most notably that the obvious design has the problem that at any given point the AI system looks like it could be goal-directed with a long-term reward function, which is the sort of system that we are most worried about.

Reward uncertainty

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

In my last post, I argued that interaction between the human and the AI system was necessary in order for the AI system to “stay on track” as we encounter new and unforeseen changes to the environment. The most obvious implementation of this would be to have an AI system that keeps an estimate of the reward function. It acts to maximize its current estimate of the reward function, while simultaneously updating the reward through human feedback. However, this approach has significant problems.

Looking at the description of this approach, one thing that stands out is that the actions are chosen according to a reward that we *know* is going to change. (This is what leads to the incentive to disable the narrow value learning system.) This seems clearly wrong: surely our plans should account for the fact that our rewards will change, *without* treating such a change as adversarial? This suggests that we need to have our action selection mechanism take the future rewards into account as well.

While we don’t know what the future reward will be, we can certainly have a *probability distribution* over it. So what if we had uncertainty over reward functions, and took that uncertainty into account while choosing actions?

Setup

We’ve drilled down on the problem sufficiently far that we can create a formal model and see what happens. So, let’s consider the following setup:

- The human, Alice, knows the “true” reward function that she would like to have optimized.
- The AI system maintains a probability distribution over reward functions, and acts to maximize the expected sum of rewards under this distribution.
- Alice and the AI system take turns acting. Alice knows that the AI learns from her actions, and chooses actions accordingly.
- Alice’s action space is such that she cannot take the action “tell the AI system the true reward function” (otherwise the problem would become trivial).
- Given these assumptions, Alice and the AI system act optimally.

This is the setup of Cooperative Inverse Reinforcement Learning (CIRL). The optimal solution to this problem typically involves Alice “teaching” the AI system by taking actions that communicate what she does and does not like, while the AI system “asks” about parts of the reward by taking actions that would force Alice to behave in different ways for different rewards.

Does this solve our problems?

Two of the problems we identified in the last post are simply assumed away:

- Alice does not know the “true” reward function, but we assumed that she does.

- Alice may be unable to optimally give feedback to the AI system, but we assume that she is optimal here.

So this particular kind of reward uncertainty does not fix either of these problems. What about convergent instrumental subgoals?

Utility preservation. One major worry we had with the original setup was that the AI system would disable its narrow value learning system, as a manifestation of the instrumental goal of protecting its utility function. This is reversed in our setup: the AI system has a positive incentive to continue doing narrow value learning, since it helps it hone in on the true reward function, which in turn allows it to optimize the reward better. (We might worry that this prevents us from fixing any problems in the narrow value learning system, but that is a robustness problem: in the world where everything is working correctly, this is the correct incentive.)

Survival incentive, aka shutdown incorrigibility. Another worry is that the AI system has a survival incentive that causes it to prevent us from shutting it down. With reward uncertainty, the fact that we are trying to shut the AI system down is itself strong evidence about the reward function. The AI system should reason that its operation leads to worse outcomes, and so allow itself to be turned off.

The Off-Switch Game formalizes this reasoning in a simple setting. The AI system can either directly take action a (perhaps by disabling an off switch), can shut down, or can allow Alice to choose between these options. If the AI system allows Alice to choose, Alice then decides whether or not to shut down the AI system. Assuming that Alice is optimal, the AI system reasons that when Alice chooses to shut it down, the true reward function must have been one which makes the action a bad, and so it does better by deferring to Alice. However, when Alice is modeled as noisily rational instead of optimal, the AI system might reason that Alice might make a mistake when deciding to shut the AI system down, and so it might take action a directly without deferring to her. So, the AI system becomes shutdown corrigible, as long as it assumes that Alice is sufficiently rational.

Should robots be obedient? makes a similar point, arguing that an AI system that learns preferences and then acts to maximize their satisfaction can perform better than an AI system that simply obeys instructions, because humans are not perfectly rational. This creates a tradeoff between performance and obedience (which shutdown corrigibility is an instance of).

Of course, these simple models exclude many actions that a realistic AI system could take. In particular, it seems likely that an AI system would prefer to disable the shutdown button, gather information about the reward until it has fully updated, and optimize the resulting set of rewards. If the space of reward functions is misspecified, as it likely will be, this will lead to bad behavior. (This is the point made by Incorrigibility in the CIRL Framework.)

Note though that while this cuts against shutdown corrigibility (since the AI system would prefer to disable the shutdown button), I would frame the problem differently. If the space of rewards is well-specified and has sufficient weight on the true reward function and the AI system is sufficiently robust and intelligent, then the AI system must update strongly on us attempting to shut it down. This should cause it to stop doing the bad thing it was doing. When it eventually narrows down on the reward it will have identified the true reward, which by definition is the right thing to optimize. So even though the AI system might disable its off switch, this is simply because it is better at knowing what we want than we are, and this leads to better outcomes for us. So, really the argument is that since we want to be robust (particularly to reward misspecification), we want shutdown corrigibility, and reward uncertainty is an insufficient solution for that.

A note on CIRL

There has been a lot of confusion on what CIRL is and isn't trying to do, so I want to avoid adding to the confusion.

CIRL is *not* meant to be a blueprint for a value-aligned AI system. It is not the case that we could create a practical implementation of CIRL and then we would be done. If we were to build a practical implementation of CIRL and use it to align powerful AI systems, we would face many problems:

- As mentioned above, Alice doesn't actually know the true reward function, and she may not be able to give optimal feedback.
- As mentioned above, in the presence of reward misspecification the AI system may end up optimizing the wrong thing, leading to catastrophic outcomes.
- Similarly, if the model of Alice's behavior is incorrect, as it inevitably will be, the AI system will make incorrect inferences about Alice's reward, again leading to bad behavior. As an example that is particularly easy to model, should the AI system model Alice as thinking about the robot thinking about Alice, or should it model Alice as thinking about the robot thinking about Alice thinking about the robot thinking about Alice? How many levels of pragmatics is the "right" level?
- Lots of other problems have not been addressed: the AI system might not deal with embeddedness well, or it might not be robust and could make mistakes, etc.

CIRL is supposed to bring conceptual clarity to what we could be trying to do in the first place with a human-AI system. In Dylan's own words, "what cooperative IRL is, it's a definition of how a human and a robot system together can be rational in the context of fixed preferences in a fully observable world state". In the same way that VNM rationality informs our understanding of humans even though humans are not expected utility maximizers, CIRL can inform our understanding of alignment proposals, even though CIRL itself is unsuitable as a solution to alignment.

Note also that this post is about reward uncertainty, not about CIRL. CIRL makes other points besides reward uncertainty, that are well explained in this blog post, and are not mentioned here.

While all of my posts have been significantly influenced by many people, this post is especially based on ideas I heard from Dylan Hadfield-Menell. However, besides the one quote, the writing is my own, and may not reflect Dylan's views.

The human side of interaction

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

The last few posts have motivated an analysis of the human-AI system rather than an AI system in isolation. So far we've looked at the notion that the AI system should get feedback from the user and that it could use reward uncertainty for corrigibility. These are focused on the AI system, but what about the human? If we build a system that explicitly solicits feedback from the human, what do we have to say about the human policy, and how the human should provide feedback?

Interpreting human actions

One major free variable in any explicit interaction or feedback mechanism is what semantics the AI system should attach to the human feedback. The classic examples of AI risk are usually described in a way where this is the problem: when we provide a reward function that rewards paperclips, the AI system interprets it literally and maximizes paperclips, rather than interpreting it pragmatically as another human would.

(Aside: I suspect this was not the original point of the paperclip maximizer, but it has become a very popular retelling, so I'm using it anyway.)

Modeling this classic example as a human-AI system, we can see that the problem is that the human is offering a form of "feedback", the reward function, and the AI system is not ascribing the correct semantics to it. The way it uses the reward function implies that the reward function encodes the *optimal behavior* of the AI system in *all possible environments* -- a moment's thought is sufficient to see that this is not actually the case. There will definitely be many cases and environments that the human did not consider when designing the reward function, and we should not expect that the reward function incentivizes the right behavior in those cases.

So what can the AI system assume if the human provides it a reward function? Inverse Reward Design (IRD) offers one answer: the human is likely to provide a particular reward function if it leads to high *true* utility behavior in the training environment. So, in the boat race example, if we are given the reward "maximize score" on a training environment where this actually leads to winning the race, then "maximize score" and "win the race" are about equally likely reward functions, since they would both lead to the same behavior in the training environment. Once the AI system is deployed on the environment in the blog post, it would notice that the two likely reward functions incentivize very different behavior. At that point, it could get more feedback from humans, or it could do something that is good according to both reward functions. The paper takes the latter approach, using risk-averse planning to optimize the worst-case behavior.

Similarly, with inverse reinforcement learning (IRL), or learning from preferences, we need to make some sort of assumption about the semantics of the human demonstrations or preferences. A typical assumption is Boltzmann rationality: the human is assumed to take better actions with higher probability. This effectively models all human biases and suboptimalities as noise. There are papers that account for biases rather than modeling them as noise. A major argument against the feasibility of ambitious value learning is that any assumption we make will be misspecified, and so we cannot infer the “one true utility function”. However, it seems plausible that we could have an assumption that would allow us to learn some values (at least to the level that humans are able to).

The human policy

Another important aspect is how the human actually computes feedback. We could imagine training human overseers to provide feedback in the manner that the AI system expects. Currently we “train” AI researchers to provide reward functions that incentivize the right behavior in the AI systems. With IRD, we only need the human to extensively test their reward function in the training environment and make sure the resulting behavior is near optimal, without worrying too much about generalization to other environments. With IRL, the human needs to provide demonstrations that are optimal. And so on.

(Aside: This is very reminiscent of human-computer interaction, and indeed I think a useful frame is to view this as the problem of giving humans better, easier-to-use tools to control the behavior of the AI system. We started with direct programming, then improved upon that to reward functions, and are now trying to improve to comparisons, rankings, and demonstrations.)

We might also want to train humans to give more careful answers than they would have otherwise. For example, it seems really good if our AI systems learn to preserve option value in the face of uncertainty. We might want our overseers to think deeply about potential consequences, be risk-averse in their decision-making, and preserve option value with their choices, so that the AI system learns to do the same. (The details depend strongly on the particular narrow value learning algorithm -- the best human policy for IRL will be very different from the best human policy for CIRL.) We might hope that this requirement only lasts for a short amount of time, after which our AI systems have learnt the relevant concepts sufficiently well that we can be a bit more lax in our feedback.

Learning human reasoning

So far I’ve been analyzing AI systems where the feedback is given explicitly, and there is a dedicated algorithm for handling the feedback. Does the analysis also apply to systems which get feedback implicitly, like iterated amplification and debate?

Well, certainly these methods will need to get feedback somehow, but they may not face the problem of ascribing semantics to the feedback, since they may have learned the semantics implicitly. For example, a sufficiently powerful imitation learning algorithm will be able to do narrow value learning simply because humans are capable of narrow value learning, even though it has no explicit assumption of semantics of the feedback. Instead, it has internalized the semantics that we humans give to other humans’ speech.

Similarly, both iterated amplification and debate inherit the semantics from humans by learning how humans reason. So they do not have the problems listed above. Nevertheless, it probably still is valuable to train humans to be good overseers for other reasons. For example, in debate, the human judges are supposed to say which

AI system provided the most true and useful information. It is crucial that the humans judge by this criterion, in order to provide the right incentives for the AI systems in the debate.

Summary

If we reify the interaction between the human and the AI system, then the AI system must make some assumption about the meaning of the human’s feedback. The human should also make sure to provide feedback that will be interpreted correctly by the AI system.

Following human norms

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

So far we have been talking about how to learn “values” or “instrumental goals”. This would be necessary if we want to figure out how to build an AI system that does exactly what we want it to do. However, we’re probably fine if we can keep learning and building better AI systems. This suggests that it’s sufficient to build AI systems that don’t screw up so badly that it ends this process. If we accomplish that, then steady progress in AI will eventually get us to AI systems that do what we want.

So, it might be helpful to break down the problem of learning values into the subproblems of learning what to do, and learning what not to do. Standard AI research will continue to make progress on learning what to do; catastrophe happens when our AI system doesn’t know what not to do. This is the part that we need to make progress on.

This is a problem that humans have to solve as well. Children learn basic norms such as not to litter, not to take other people’s things, what not to say in public, etc. As argued in Incomplete Contracting and AI alignment, any contract between humans is never explicitly spelled out, but instead relies on an external unwritten normative structure under which a contract is interpreted. (Even if we don’t explicitly ask our cleaner not to break any vases, we still expect them not to intentionally do so.) We might hope to build AI systems that infer and follow these norms, and thereby avoid catastrophe.

It’s worth noting that this will probably not be an instance of narrow value learning, since there are several differences:

- Narrow value learning requires that you learn what *to* do, unlike norm inference.
- Norm following requires learning from a complex domain (human society), whereas narrow value learning can be applied in simpler domains as well.
- Norms are a property of groups of agents, whereas narrow value learning can be applied in settings with a single agent.

Despite this, I have included it in this sequence because it is plausible to me that value learning techniques will be relevant to norm inference.

Paradise prospects

With a norm-following AI system, the success story is primarily around accelerating our rate of progress. Humans remain in charge of the overall trajectory of the future, and we use AI systems as tools that enable us to make better decisions and create better technologies, which looks like “superhuman intelligence” from our vantage point today.

If we still want an AI system that colonizes space and optimizes it according to our values without our supervision, we can figure out what our values are over a period of reflection, solve the alignment problem for goal-directed AI systems, and then create such an AI system.

This is quite similar to the success story in a world with Comprehensive AI Services.

Plausible proposals

As far as I can tell, there has not been very much work on *learning* what not to do. Existing approaches like impact measures and mild optimization are aiming to *define* what not to do rather than learn it.

One approach is to scale up techniques for narrow value learning. It seems plausible that in sufficiently complex environments, these techniques will learn what not to do, even though they are primarily focused on what to do in current benchmarks. For example, if I see that you have a clean carpet, I can infer that it is a norm not to walk over the carpet with muddy shoes. If you have an unbroken vase, I can infer that it is a norm to avoid knocking it over. This paper of mine shows how this you can reach these sorts of conclusions with narrow value learning (specifically a variant of IRL).

Another approach would be to scale up work on ad hoc teamwork. In ad hoc teamwork, an AI agent must learn to work in a team with a bunch of other agents, without any prior coordination. While current applications are very task-based (eg. playing soccer as a team), it seems possible that as this is applied to more realistic environments, the resulting agents will need to infer norms of the group that they are introduced into. It's particularly nice because it explicitly models the multiagent setting, which seems crucial for inferring norms. It can also be thought of as an alternative statement of the problem of AI safety: how do you "drop in" an AI agent into a "team" of humans, and have the AI agent coordinate well with the "team"?

Potential pros

Value learning is hard, not least because it's hard to define what values are, and we don't know our own values to the extent that they exist at all. However, we do seem to do a pretty good job of learning society's norms. So perhaps this problem is significantly easier to solve. Note that this is an argument that norm-following is easier than ambitious value learning, not that it is easier than other approaches such as corrigibility.

It is also feels easier to work on inferring norms right now. We have many examples of norms that we follow; so we can more easily evaluate whether current systems are good at following norms. In addition, ad hoc teamwork seems like a good start at formalizing the problem, which we still don't really have for "values".

This also more closely mirrors our tried-and-true techniques for solving the principal-agent problem for humans: there is a shared, external system of norms, that everyone is expected to follow, and systems of law and punishment are interpreted with respect to these norms. For a much more thorough discussion, see Incomplete Contracting and AI alignment, particularly Section 5, which also argues that norm following will be *necessary for value alignment* (whereas I'm arguing that it is plausibly *sufficient to avoid catastrophe*).

One potential confusion: the paper says "We do not mean by this embedding into the AI the particular norms and values of a human community. We think this is as impossible a task as writing a complete contract." I believe that the meaning here is that we should not try to *define* the particular norms and values, not that we shouldn't try to *learn* them. (In fact, later they say “Aligning AI with human values, then, will require figuring out how to build the technical tools that will allow a robot to replicate the human agent’s ability to read and predict the responses of human normative structure, whatever its content.”)

Perilous pitfalls

What additional things could go wrong with powerful norm-following AI systems? That is, what are some problems that might arise, that wouldn’t arise with a successful approach to ambitious value learning?

- Powerful AI likely leads to rapidly evolving technologies, which might require rapidly changing norms. Norm-following AI systems might not be able to help us develop good norms, or might not be able to adapt quickly enough to new norms. (One class of problems in this category: we would not be addressing human safety problems.)
- Norm-following AI systems may be uncompetitive because the norms might overly restrict the possible actions available to the AI system, reducing novelty relative to more traditional goal-directed AI systems. (Move 37 would likely not have happened if AlphaGo were trained to “follow human norms” for Go.)
- Norms are more like soft constraints on behavior, as opposed to goals that can be optimized. Current ML focuses a lot more on optimization than on constraints, and so it’s not clear if we could build a competitive norm-following AI system (though see eg. Constrained Policy Optimization).
- Relatedly, learning what not to do imposes a limitation on behavior. If an AI system is goal-directed, then given sufficient intelligence it will likely find a nearest unblocked strategy.

Summary

One promising approach to AI alignment is to teach AI systems to infer and follow human norms. While this by itself will not produce an AI system aligned with human values, it may be sufficient to avoid catastrophe. It seems more tractable than approaches that require us to infer values to a degree sufficient to avoid catastrophe, particularly because humans are proof that the problem is soluble.

However, there are still many conceptual problems. Most notably, norm following is not obviously expressible as an optimization problem, and so may be hard to integrate into current AI approaches.

Future directions for narrow value learning

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

Narrow value learning is a huge field that people are already working on (though not by that name) and I can't possibly do it justice. This post is primarily a list of things that I think are important and interesting, rather than an exhaustive list of directions to pursue. (In contrast, the corresponding post for ambitious value learning did aim to be exhaustive, and I don't think I missed much work there.)

You might think that since so many people are already working on narrow value learning, we should focus on more neglected areas of AI safety. However, I still think it's worth working on because long-term safety suggests a particular subset of problems to focus on; that subset seems quite neglected.

For example, a lot of work is about how to improve current algorithms in a particular domain, and the solutions encode domain knowledge to succeed. This seems not very relevant for long-term concerns. Some work assumes that a handcoded featurization is given (so that the true reward is linear in the features); this is not an assumption we could make for more powerful AI systems.

I will speculate a bit on the neglectedness and feasibility of each of these areas, since for many of them there isn't a person or research group who would champion them whom I could defer to about the arguments for success.

The big picture

This category of research is about how you could take narrow value learning algorithms and use them to create an aligned AI system. Typically, I expect this to work by having the narrow value learning enable some form of corrigibility.

As far as I can tell, nobody outside of the AI safety community works on this problem. While it is far too early to stake a confident position one way or the other, I am slightly less optimistic about this avenue of approach than one in which we create a system that is directly trained to be corrigible.

Avoiding problems with goal-directedness. How do we put together narrow value learning techniques in a way that doesn't lead to the AI behaving like a goal-directed agent at each point? This is the problem with keeping a reward estimate that is updated over time. While reward uncertainty can help avoid some of the problems, it does not seem sufficient by itself. Are there other ideas that can help?

Dealing with the difficulty of "human values". Cooperative IRL makes the unrealistic assumption that the human knows her reward function exactly. How can we make narrow value learning systems that deal with this issue? In particular, what prevents them from updating on our behavior that's not in line with our "true values", while still letting them update on other behavior? Perhaps we could make an AI system that is always uncertain about what the true reward is, but how does this mesh with epistemics, which suggest that you can get to arbitrarily high confidence given sufficient evidence?

Human-AI interaction

This section of research aims to figure out how to create human-AI systems that successfully accomplish tasks. For sufficiently complex tasks and sufficiently powerful AI, this overlaps with the big picture concerns above, but there are also areas to work on with subhuman AI with an eye towards more powerful systems.

Assumptions about the human. In any feedback system, the update that the AI makes on the human feedback depends on the assumption that the AI makes about the human. In Inverse Reward Design (IRD), the AI system assumes that the reward function provided by a human designer leads to near-optimal behavior in the training environment, but may be arbitrarily bad in other environments. In IRL, the typical assumption is that the demonstrations are created by a human behaving Boltzmann rationally, but recent research aims to also correct for any suboptimalities they might have, and so no longer assumes away the problem of systematic biases. (See also the discussion in Future directions for ambitious value learning.) In Cooperative IRL, the AI system assumes that the human models the AI system as approximately rational. COACH notes that when you ask a human to provide a reward signal, they provide a critique of current behavior rather than a reward signal that can be maximized.

Can we weaken the assumptions that we have to make, or get rid of them altogether? Barring that, can we make our assumptions more realistic?

Managing interaction. How should the AI system manage its interaction with the human to learn best? This is the domain of active learning, which is far too large a field for me to summarize here. I'll throw in a link to Active Inverse Reward Design, because I already talked about IRD and I helped write the active variant.

Human policy. The utility of a feedback system is going to depend strongly on the quality of the feedback given by the human. How do we train humans so that their feedback is most useful for the AI system? So far, most work is about how to adapt AI systems to understand humans better, but it seems likely there are also gains to be had by having humans adapt to AI systems.

Finding and using preference information

New sources of data. So far preferences are typically learned through demonstrations, comparisons or rankings; but there are likely other useful ways to elicit preferences. Inverse Reward Design gets preferences from a stated proxy reward function. An obvious one is to learn preferences from what people say, but natural language is notoriously hard to work with so not much work has been done on it so far, though there is some. (I'm pretty sure there's a lot more in the NLP community that I'm not yet aware of.) We recently showed that there is even preference information in the state of the world that can be extracted.

Handling multiple sources of data. We could infer preferences from behavior, from speech, from given reward functions, from the state of the world, etc. but it seems quite likely that the inferred preferences would conflict with each other. What do you do in these cases? Is there a way to infer preferences simultaneously from all the sources of data such that the problem does not arise? (And if so, what is the algorithm implicitly doing in cases where different data sources pull in different directions?)

Acknowledging Human Preference Types to Support Value Learning talks about this problem and suggests some aggregation rules but doesn't test them. Reward Learning from Narrated Demonstrations learns from both speech and demonstrations, but they are used as complements to each other, not as different sources for the same information that could conflict.

I'm particularly excited about this line of research -- it seems like it hasn't been explored yet and there are things that can be done, especially if you allow yourself to simply detect conflicts, present the conflict to the user, and then trust their answer. (Though this wouldn't scale to superintelligent AI.)

Generalization. Current deep IRL algorithms (or deep anything algorithms) do not generalize well. How can we infer reward functions that transfer well to different environments? Adversarial IRL is an example of work pushing in this direction, but my understanding is that it had limited success. I'm less optimistic about this avenue of research because it seems like in general function approximators do not extrapolate well. On the other hand, I and everyone else have the strong intuition that a reward function should take fewer bits to specify than the full policy, and so should be easier to infer. (Though not based on Kolmogorov complexity.)

Conclusion to the sequence on value learning

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

This post summarizes the sequence on value learning. While it doesn't introduce any new ideas, it does shed light on which parts I would emphasize most, and the takeaways I hope that readers get. I make several strong claims here; interpret these as my impressions, not my beliefs. I would guess many researchers disagree with the (strength of the) claims, though I do not know what their arguments would be.

Over the last three months we've covered a lot of ground. It's easy to lose sight of the overall picture over such a long period of time, so let's do a brief recap.

The "obvious" approach

Here is an argument for the importance of AI safety:

- Any agent that is much more intelligent than us should not be exploitable by us, since if we could find some way to exploit the agent, the agent could also find the exploit and patch it.
- Anything that is not exploitable must be an expected utility maximizer; since we cannot exploit a superintelligent AI, it must look like an expected utility maximizer to us.
- Due to Goodhart's Law, even "slightly wrong" utility functions can lead to catastrophic outcomes when maximized.
- Our utility function is complex and fragile, so getting the "right" utility function is difficult.

This argument implies that by the time we have a superintelligent AI system, there is only one part of that system that could still have been influenced by us: the utility function. Every other feature of the AI system is fixed by math. As a result, we must *necessarily* solve AI alignment by influencing the utility function.

So of course, the natural approach is to get the right utility function, or at least an adequate one, and have our AI system optimize that utility function. Besides fragility of value, which you might hope that machine learning could overcome, the big challenge is that even if you assume full access to the entire human policy, we cannot infer their values without making an assumption about how their preferences relate to their behavior. In addition, any misspecification can lead to bad inferences. And finally the entire project of having a single utility function that captures optimal behavior in all possible environments seems quite hard to do -- it seems necessary to have some sort of feedback from humans, or you end up extrapolating in some strange way that is not necessarily what we "would have" wanted.

So does this mean we're doomed? Well, there are still some potential avenues for rescuing ambitious value learning, though they do look quite difficult to me. But I think we should actually question the assumptions underlying our original argument.

Problems with the standard argument

Consider the calculator. From the perspective of someone before the time of calculators, this device would look quite intelligent -- just look at the speed with which it can do arithmetic! Nonetheless, we can all agree that a standard calculator is not dangerous.

It also seems strange to ascribe goals to the calculator -- while this is not *wrong* per se, we certainly have better ways of predicting what a calculator will and will not do than by modelling it as an expected utility maximizer. If you model a calculator as aiming to achieve the goal of “give accurate math answers”, problems arise: what if I take a hammer to the calculator and then try to ask it 5 + 3? The utility maximizer model here would say that it answers 8, whereas with our understanding of how calculators work we know it probably won’t give any answer at all. Utility maximization with a simple utility function is only a good model for the calculator within a restricted set of environmental circumstances and a restricted action space. (For example, we don’t model the calculator as having access to the action, “build armor that can protect against hammer attacks”, because otherwise utility maximization would predict it takes that action.)

Of course, it may be that something that is generally superintelligent will work in as broad a set of circumstances as we do, and will have as wide an action space as we do, and must still look to us like an expected utility maximizer since otherwise we could Dutch book it. However, if you take such a broad view, then it turns out that all behavior looks coherent. There’s no *mathematical* reason that an intelligent agent must have catastrophic behavior, since *any* behavior that you observe is consistent with the maximization of some utility function.

To be clear, while I agree with every statement in Optimized agent appears coherent, I am making the strong claim that these statements are *vacuous* and by themselves tell us nothing about the systems that we will actually build. Typically, I do not flat out disagree with a common argument. I usually think that the argument is important and forms a piece of the picture, but that there are other arguments that push in other directions that might be more important. That’s not the case here: I am claiming that the argument that “superintelligent agents must be expected utility maximizers by virtue of coherence arguments” provides *no* useful information, with almost the force of a theorem. My uncertainty here is almost entirely caused by the fact that other smart people believe that this argument is important and relevant.

I am *not* claiming that we don’t need to worry about AI safety since AIs won’t be expected utility maximizers. First of all, you *can* model them as expected utility maximizers, it’s just not useful. Second, if we build an AI system whose internal reasoning consisted of maximizing the expectation of some simple utility function, I think all of the classic concerns apply. Third, it does seem likely that humans will build AI systems that are “trying to pursue a goal”, and that can have all of the standard convergent instrumental subgoals. I propose that we describe these systems as goal-directed rather than expected utility maximizers, since the latter is vacuous and implies a level of formalization that we have not yet reached. However, this risk is significantly different. If you believed that superintelligent AI *must* be goal-directed because of math, then your only recourse for safety would be to make sure that the goal is good, which is what motivated us to study ambitious value learning. But if the argument is actually that AI will be goal-directed because humans will make it that way, you could try to build AI that is not goal-directed that can do the things that goal-directed AI can do, and have humans build that instead.

Alternative solutions

Now that we aren't forced to influence just a utility function, we can consider alternative designs for AI systems. For example, we can aim for corrigible behavior, where the agent is *trying* to do what we want. Or we could try to learn human norms, and create AI systems that follow these norms while trying to accomplish some task. Or we could try to create an AI ecosystem akin to Comprehensive AI Services, and set up the services such that they are keeping each other in check. We could create systems that learn how to do what we want in particular domains, by learning our instrumental goals and values, and use these as subsystems in AI systems that accelerate progress, enable better decision-making, and are generally corrigible. If we want to take such an approach, we have another source of influence: the human policy. We can train our human overseers to provide supervision in a particular way that leads to good behavior on the AI's part. This is analogous to training operators of computer systems, and can benefit from insights from Human-Computer Interaction (HCI).

Not just value learning

This sequence is somewhat misnamed: while it is organized around value learning, there are many ideas that should be of interest to researchers working on other agendas as well. Many of the key ideas can be used to analyze *any* proposed solution for alignment (though the resulting analysis may not be very interesting).

The necessity of feedback. The main argument of Human-AI Interaction is that any proposed solution that aims to have an AI system (or a CAIS glob of services) produce good outcomes over the long term needs to continually use data about humans as feedback in order to "stay on target". Here, "human" is shorthand for "something that we know shares our values", eg. idealized humans, uploads, or sufficiently good imitation learning would all probably count.

(If this point seems obvious to you, note that ambitious value learning does not clearly satisfy this criterion, and approaches like impact measures, mild optimization, and boxing are punting on this problem and aiming for not-catastrophic outcomes rather than good outcomes.)

Mistake models. We saw that ambitious value learning has the problem that even if we assume perfect information about the human, we cannot infer their values without making an assumption about how their preferences relate to their behavior. This is an example of a much broader pattern: given that our AI systems necessarily get feedback from us, they must be making some assumption about how to interpret that feedback. For any proposed solution to alignment, we should ask what assumptions the AI system is making about the feedback it gets from us.