GOAL

Toying With Goal-Directedness

Goal-directed = Model-based RL?

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

Epistemic Status: quick write-up, in reaction to a serendipitous encounter with an idea. I see the main value of this post as decently presenting a potentially interesting take on a concept in AI safety to the community.

While skimming my copy of Reinforcement Learning: an introduction for the part on AlphaGo Zero, I found a section called Habitual and Goal-directed behavior. That caught my attention, because one idea I keep going back to is goal-directed behavior from the Value Learning Sequence; when studying the sequence, I was intrigued by the idea. But the lack of formalization made me uncertain about my position on the argument, that not all useful (and possibly superintelligent) agents have to be goal-directed.

Going back to my serendipitous discovery, the section is part of the Psychology chapter: it compares model-based and model-free RL to goal-directed and habitual behavior in psychology. To clarify the RL terms used here:

- **Model-based RL** is the version of RL where the agents learns both from direct experience with the environment and from simulated experience with a model, which entails that it builds and updates a model of the environment.
- **Model-free RL** is the version of RL where the agents only learns from direct experience with the environment. Usually, the only thing available to the agent is its value function or its policy; there is no model of the environment.

As for the different behaviors, let's quote the book itself:

> Goal-directed behavior, according to how psychologists use the phrase, is purposeful in the sense that it is controlled by knowledge of the value of goals and the relationship between actions and their consequences.

and

> Habits are behavior patterns triggered by appropriate stimuli and then performed more-or-less automatically.

There is also a summary:

> Habits are sometimes said to be controlled by antecedent stimuli, whereas goal-directed behavior is said to be controlled by its consequences

Now, is this version of goal-directed behavior linked to the version that Rohin Shah wrote about? I think so. They are probably not the same, but examining the similarities and differences might clarify the part relevant to AI safety.

Comparison between the two versions of goal-directed behavior

Similarities

In intuitions about goal-directed behavior, the first example of goal-directed behavior vs non-goal directed behavior concerns policies for agents playing TicTacToe:

> 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!

This feels very model-based RL vs model-free RL to me! It's almost the same example as in the Figure 14.9 from the section: a rat tries to navigate a maze towards different rewards, and can either learn pure action-values from experience (habitual-behavior/model-free RL) or learn a model of the maze and fill it with action-values (goal-directed behavior/model-based RL).

(Notice that here, I assumes that the lookup-table contains actions learned on the environment, or at least adapted to the environment. I consider the case where the lookup table is just hard-coded random values in the next subsection).

There is also a parallel between the advantages of goal-directed behavior given by Rohin:

> 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.

and the intuition behind goal-directed behavior from the section:

> Goal-directed control has the advantage that it can rapidly change an animal’s behavior when the environment changes its way of reacting to the animal’s actions.

Differences

Going back to the TicTacToe example, we can interpret the lookup-table version as being hard-coded. If that's the case, then it is not really analoguous to model-free RL.

In the same vein, Rohin gives more examples of behavior he considers to **not** be goal-directed in another post:

- 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.

I can think of ways to explain these as habitual behavior, but it feels a bit forced to me. As I understand it, habitual behavior is still adaptative, just on a potentially longer scale and through different mechanisms. On the other hand, the examples above are about "habits" that are not even suited to the original environment.

Conclusion

These two versions of goal-directed behavior seem linked to me. Whether they are actually the same, or whether the connection will prove useful for safety research is still unclear.

Focus: you are allowed to be bad at accomplishing your goals

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

When asked about what it means for a system to be goal-directed, one common answer draws on some version of Dennett's intentional stance: a goal-directed system is a system such that modeling it as having a goal provides accurate and efficient predictions about its behavior. I agree up to that point. But then, some people follow up by saying that the prediction is that the system will accomplish its goal. For example, it makes sense to model AlphaGo as goal-directed towards winning at Go, because it will eventually win. And taking the intentional stance allows me to predict that.

But what if I make AlphaGo play against AlphaZero, which is strictly better at Go? Then AlphaGo will consistently lose. Does it mean that it's no longer goal-directed towards winning?

What feels wrong to me is the implicit link drawn between goal-directedness and competence. A bad Go player will usually lose, but it doesn't seem any less goal-directed to me than a stronger one that consistently wins.

Competence is thus not the whole story. It might be useful to compute goal-directedness; reaching some lower-bound of competency might even be a necessary condition for goal-directedness (play badly enough and it becomes debatable whether you're even trying to win). But when forcing together the two, I feel like something important is lost.

To solve this problem, I propose a new metric of goal-directedness, focus: how much is the system trying to accomplish a certain goal. Focus is not the whole story about being goal-directed, but I think computing the focus of a system for some goal (details in the next paragraph) gives useful information about its goal-directedness.

Given a system S (as a function from states or histories to actions) and a goal G (as a set of states), here are the steps to compute the focus of S towards G.

- I define a reward function over states R valued 1 at states in and 0 at all other states.
- Then I define Pol be the set of all policies that can be generated by Reinforcement Learning (RL) on R. I'll go into details about Pol below, but the most important part here is that it isn't limited to optimal policies; I also consider policies of RL with "few resources". Basically all policies at intermediary steps of the RL training are in Pol.
- Lastly, I pick a distance between policies. If the two policies are deterministic, a Hamming distance will do. If they are stochastic, maybe some vector distance based on the Kullback-Leibler divergence.

- Then, the focus of S towards G is inversely proportional to the distance between S and Pol.

The intuition here is that any policy that result from training on this reward function is aiming maximally towards the goal, by definition. And by taking the appropriate distance, we can measure how far our system is from such a fully focused policy. The distance captures the proportion of actions taken by the policy that fits with aiming towards the specific goal.

Of course, there are many points that need further thought:

- What does “all policies given by using RL” mean in this case? The easy answer is all policies resulting from taking any RL method and any initial conditions, and training for any amount of resources on the reward function of the goal. But not only is this really, really uncomputable, I’m not sure it’s well defined enough what are “all methods of RL”?). Ideally, I would want to limit the study to one specific RL algorithm (SARSA for example) and then the set of generated policies would be well-defined. But I’m not sure if I’m losing any policy by doing so.
- Even when fixing some RL algorithm, it is completely unfeasible to consider all initial conditions and amounts of resources. Yet this is the obvious way to compute the set of maximally-focused policies. Here I hope for either a dense subset (or a good approximation) of this set of policies, or even an analytical characterization if ones exists.
- The ghost of competence strikes back here, because I cannot really consider any amount of resources; if I did, then every policy would be maximally-focused for the goal, as it would be generated by taking the policy as an initial condition and using no resources at all. My intuition for dealing with this is that there should be a meaningful lower bound on the amount of resources the RL algorithm has to use before the resulting policy is indeed maximally-focused. Maybe enough resource for all state values or state-action pairs value to have been updated at least once?

Finally, assuming we are able to compute the focus of any system for any goal, how to interpret the results we get? Focus is not divided between goals like probability: for example, the full goal consisting of all possible states always has maximal focus, as all policies are optimal for the corresponding reward; but other goals might also have the same focus. This entails that finding the most representative goal is not only about focus, but also about the triviality of the goal.

My far less clean intuition here is that the “triviality” of the goal should weight its focus. That is, the goal consisting of all possible states is trivial, whereas the one consisting of exactly one state is not trivial at all. Thus even if the former has stronger focus than the latter, it has to be really, really stronger to compensate its triviality. Or said another way, a non-trivial goal with a small but not negligible focus exhibits goal-directedness than a trivial goal with enormous focus.

Even with all those uncertainties, I still believe focus is a step in the right direction. It trims down competence to the part that seems the most relevant to goal-directedness. That being said, I am very interested in any weakness of the idea, or any competing intuition.

Thanks to Jérémy Perret for feedback on the writing, and to Joe Collman, Michele Campolo and Sabrina Tang for feedback on the idea.

Goal-directedness is behavioral, not structural

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

Goal-directedness is the term used by the AI Safety community to point to a specific property: following a goal. It comes from Rohin Shah's post in his sequence, but the intuition pervades many safety issues and current AI approaches. Yet it lacks a formal definition, or even a decomposition into more or less formal subcomponents.

Which questions we want to answer about goal-directed systems underlies the sort of definition we're looking for. There are two main questions that Rohin asks in his posts:

- Are non goal-directed systems or less goal-directed ones inherently safer than fully goal-directed ones?
- Can non-goal-directed systems or less goal-directed ones be competitive with fully goal-directed ones?

Answering these will also answer the really important meta-question: should we put resources into non-goal-directed approaches to AGI?

Notice that both questions above are about predicting properties of the system based on its goal-directedness. These properties we care about depend only on the behavior of the system, not on its internal structure. It thus makes sense to consider that goal-directedness should also depend only on the behavior of the system. For if it didn't, then two systems with the same properties (safety, competitiveness) would have different goal-directedness, breaking the pattern of prediction.

Actually, this assumes that our predictor is injective: it sends different "levels" of goal-directedness to different values of the properties. I agree with this intuition, given how much performance and safety issues seem to vary according to goal-directedness. But I wanted to make it explicit.

Reiterating the point of the post: goal-directedness is a property of behavior, not internal structure. By this I mean that given the complete behavior of a system over all environment, goal-directedness is independent of what's inside the system. Or equivalently, if two systems always behave in the same way, their goal-directedness is the same, regardless of if one contains a big lookup table and the other an homonculus.

This is not particularly original: Dennett's intentional stance pretty much says the same thing. (The Intentional Stance, p 15)

> Then I will argue that any object -- or as I shall say, any system -- whose behavior is well predicted by this strategy [considering it as moving towards a goal] is in the fullest sense of the word a believer. What it is to be a true believer is to be an intentional system, a system whose behavior is reliably and voluminously predictable via the intentional strategy.

Why write a post about it, then? I'm basically saying that our definition should depend only on observable behavior, which is pretty obvious, isn't it?

Well, goal is a very loaded term. It is a part of the set of mental states we attribute to human beings, and other agents, but that we are reluctant to give to anything else. See how I never used the word "agent" before in this post, preferring "system" instead? That was me trying to limit this instinctive thinking about what's inside. And here is the reason why I think this post is not completely useless: when looking for a definition of goal-directedness, the first intuition is to look for the internal structure. It seems obvious that goals should be somewhere "inside" the system, and thus that what really matters is the internal structure.

But as we saw above, goal-directedness should probably depend only on the complete behavior of the system. That is not to say that the internal structure is not important or useful here. On the contrary, this structure, in the form of source code for example, is usually the only thing we have at our disposal. It serves to compute goal-directedness, instead of defining it.

We thus have this split:

- Defining goal-directedness: depends only on the complete behavior of the system, and probably assumes infinite compute and resources.
- Computing goal-directedness: depends on the internal structure, and more specifically what information about the complete behavior can be extracted from this structure.

What I see as a mistake here, a mistake I personally made, is to look for the definition in the internal structure. To look at some neural net, or some C program, and try to find where the goals are and what makes the program follow them. Instead, I think we should define and formalize goal-directedness from the ideal context of knowing the full behavior of the system, and then use interpretability and formal methods to extract what's relevant to this definition from the internal structure.

Thanks to Jérémy Perret for feedback on the writing, and to Joe Collman, Michele Campolo and Sabrina Tang for feedback on the idea.

Locality of goals

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

Introduction

Studying goal-directedness produces two kinds of questions: questions about goals, and questions about being directed towards a goal. Most of my previous posts focused on the second kind; this one shifts to the first kind.

Assume some goal-directed system with a known goal. The nature of this goal will influence which issues of safety the system might have. If the goal focuses on the input, the system might wirehead itself and/or game its specification. On the other hand, if the goal lies firmly in the environment, the system might have convergent instrumental subgoals and/or destroy any unspecified value.

Locality aims at capturing this distinction.

Intuitively, the locality of the system's goal captures how far away from the system one must look to check the accomplishment of the goal.

Let's give some examples:

- The goal of "My sensor reaches the number 23" is very local, probably maximally local.
- The goal of "Maintain the temperature of the room at 23 °C" is less local, but still focused on a close neighborhood of the system.
- The goal of "No death from cancer in the whole world" is even less local.

Locality isn't about how the system extract a model of the world from its input, but about whether and how much it cares about the world beyond it.

Starting points

This intuition about locality came from the collision of two different classification of goals: the first from from Daniel Dennett and the second from Evan Hubinger.

Thermostats and Goals

In "The Intentional Stance", Dennett explains, extends and defends... the intentional stance. One point he discusses is his liberalism: he is completely comfortable with admitting ridiculously simple systems like thermostats in the club of intentional systems -- to give them meaningful mental states about beliefs, desires and goals.

Lest we readers feel insulted at the comparison, Dennett nonetheless admits that the goals of a thermostat differ from ours.

> Going along with the gag, we might agree to grant [the thermostat] the capacity for about half a dozen different beliefs and fewer desires—it can believe the room is too cold or too hot, that the boiler is on or off, and that if it wants the room warmer it should turn on the boiler, and so forth. But surely this is imputing too much to the thermostat; it has no concept of heat or of a boiler, for instance. So suppose we de-interpret its beliefs and desires: it can believe the A is too F or G, and if it wants the A to be more F it should do K, and so forth. After all, by attaching the thermostatic control mechanism to different input and output devices, it could be made to regulate the amount of water in a tank, or the speed of a train, for instance.

The goals and beliefs of a thermostat are thus not about heat and the room it is in, as our anthropomorphic bias might suggest, but about the binary state of its sensor.

Now, if the thermostat had more information about the world -- a camera, GPS position, general reasoning ability to infer information about the actual temperature from all its inputs --, then Dennett argues its beliefs and goals would be much more related to heat in the room.

> The more of this we add, the less amenable our device becomes to serving as the control structure of anything other than a room-temperature maintenance system. A more formal way of saying this is that the class of indistinguishably satisfactory models of the formal system embodied in its internal states gets smaller and smaller as we add such complexities; the more we add, the richer or more demanding or specific the semantics of the system, until eventually we reach systems for which a unique semantic interpretation is practically (but never in principle) dictated (cf. Hayes 1979). At that point we say this device (or animal or person) has beliefs about heat and about this very room, and so forth, not only because of the system's actual location in, and operations on, the world, but because we cannot imagine an-other niche in which it could be placed where it would work.

Humans, Dennett argues, are more like this enhanced thermostat, in that our beliefs and goals intertwine with the state of the world. Or put differently, when the world around us changes, it will influence almost always influence our mental states; whereas a basic thermostat might react the exact same way in vastly different environments.

> But as systems become perceptually richer and behaviorally more versatile, it becomes harder and harder to make substitutions in the actual links of the system to the world without changing the organization of the system itself. If you change its environment, it will notice, in effect, and make a change in its internal state in response. There comes to be a two-way constraint of growing specificity between the device and the environment. Fix the device in any one state and it demands a very specific environment in which to operate properly (you can no longer switch it easily from regulating temperature to regulating speed or anything else); but at the same time, if you do not fix the state it is in, but just plonk it down in a changed environment, its sensory attachments will be sensitive and discriminative enough to respond appropriately to the change, driving the system into a new state, in which it will operate effectively in the new environment.

Part of this distinction between goals comes from generalization, a property considered necessary for goal-directedness since Rohin's initial post on the subject. But the two goals also differs in their "groundedness": the thermostat's goal lies completely in its sensors' inputs, whereas the goals of humans depend on things farther away, on the environment itself.

That is, these two goals have different locality.

Goals Across Cartesian Boundaries

The other classification of goals comes from Evan Hubinger, in a personal discussion. Assuming a Cartesian Boundary outlining the system and its inputs and outputs, goals can be functions of:

- **The environment**. This includes most human goals, since we tend to refuse wireheading. Hence the goal depends on something else than our brain state.
- **The input**. A typical goal as a function of the input is the one ascribed to the simple thermostat: maintaining the number given by its sensor above some threshold. If we look at the thermostat without assuming that its goal is a proxy for something else, then this system would happily wirehead itself, as the goal IS the input.
- **The output**. This one is a bit weirder, but captures goals about actions: for example, the goal of twitching. If there is a robot that only twitches, not even trying to keep twitching, just twitching, its goal seems about its output only.
- **The internals**. Lastly, goals can depend on what happens inside the system. For example, a very depressed person might have the goal of "Feeling good". If that is the only thing that matters, then it is a goal about their internal state, and nothing else.

Of course, many goals are functions of multiple parts of this quatuor. Yet separating them allows a characterization of a given goal through their proportions.

Going back to Dennett's example, the basic thermostat's goal is a function of its input, while human goals tend to be functions of the environment. And once again, an important aspect of the difference appears to lie in how far from the system is there information relevant to the goal -- locality.

What Is Locality Anyway?

Assuming some model of the world (possibly a causal DAG) containing the system, the locality of the goal is inversely proportional to the minimum radius of a ball, centered at the system, which suffice to evaluate the goal. Basically, one needs to look a certain distance away to check whether one’s goal is accomplished; locality is a measure of this distance. The more local a goal, the less grounded in the environment, and the most it is susceptible to wireheading or change of environment without change of internal state.

Running with this attempt at formalization, a couple of interesting point follow:

- If the model of the world includes time, then locality also captures how far in the future and in the past one must go to evaluate the goal. This is basically the short-sightedness of a goal, as exemplified by variants of twitching robots: the robot that simply twitches; the one that want to maximize its twitch in the next second; the one that want to maximize its twitching in the next 2 seconds,... up to the robot that want to maximize the time it twitches in the future.

- Despite the previous point, locality differs from the short term/long term split. An example of a short-term goal (or one-shot goal) is wanting an ice cream: after its accomplishment, the goal simply dissolves. Whereas an example of a long-term goal (or continuous goal) is to bring about and maintaing world peace -- something that is never over, but instead constrains the shape of the whole future. Short-sightedness differs from short-term, as a short-sighted goal can be long-term: "for all times t (in hours to simplify), I need to eat an ice cream in the interval [t-4,t+4]".
- Where we put the center of the ball inside the system is probably irrelevant, as the classes of locality should matter more than the exact distance.
- An alternative definition would be to allow the center of the ball to be anywhere in the world, and make locality inversely proportional to the sum of the distance of the center to the system plus the radius. This captures goals that do not depend on the state of the system, but would give similar numbers than the initial definition.

In summary, locality is a measure of the distance at which information about the world matters for a system's goal. It appears in various guises in different classification of goals, and underlies multiple safety issues. What I give is far from a formalization; it is instead a first exploration of the concept, with open directions to boot. Yet I believe that the concept can be put into more formal terms, and that such a measure of locality captures a fundamental aspect of goal-directedness.

Thanks to Victoria Krakovna, Evan Hubinger and Michele Campolo for discussions on this idea.

Goals and short descriptions

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

Outline

I develop some contents—previously introduced in the Value Learning sequence by Rohin Shah—more formally, to clarify the distinction between agents with and without a goal. Then I present related work and make some considerations on the relation between safety and goal-directedness. The appendix contains some details on the used formalism and can be skipped without losing much information.

A brief preliminary

In the first post of the Value Learning sequence, Shah compares two agents that exhibit the same behaviour (a winning strategy) when playing Tic-Tac-Toe, but are different in their design: one applies the minimax algorithm to the setting and rules of the game, while the other one follows a lookup table—you can think of its code as a long sequence of if-else statements.

Shah highlights the difference in terms of generalisation: the first one would still win if the winning conditions were changed, while the lookup table would not. Generalisation is one of the components of goal-directedness, and lookup tables are among the least goal-directed agent designs.

Here I want to point at another difference that exists between agents with and without a goal, based on the concept of algorithmic complexity.

Setup

Most problems in AI consist in finding a function $\pi \in A^O$, called policy in some contexts, where $A = \{a_1, \dots, a_m\}$ and $O = \{o_1, \dots, o_n\}$ indicate the sets of possible actions and observations. A deterministic policy can be written as a string $\pi = a_{i_1} a_{i_2} \dots a_{i_n}$ with a_{i_k} indicating the action taken when o_k is observed.

Here I consider a problem setting as a triplet (A, O, D) where D stands for some kind of environmental data—could be about, for example, the transition function in a MDP, or the structure of the elements in the search space O. Since I want to analyse behaviour across different environments, instead of considering one single policy I'll sometimes refer to a more general function g (probably closer to the concept of "agent design", rather than g just “agent”) mapping environments to policies on them:

$(A, O, D) \mapsto \pi_{(A,O,D)}$.

Lookup table vs simple-reward RL

As written before, a lookup table is a policy that is described case-by-case, by explicitly giving the list of all observation-action pairs. Thus, a generic lookup table for a setting (A, O, D) is expected to have Kolmogorov complexity $K(\pi|D) \approx |O|$: most lookup tables are incompressible.

Now let’s consider a policy generated via Reinforcement Learning. Such a policy can be written as $\pi = RL(p(D))$, where RL is the training algorithm, and p indicates an algorithmic procedure that gets the environmental data D as input and returns a reward function $r: O \to [-1; 1]$ to be used by the RL algorithm. Often, p corresponds to the “work” done by the human designers who assign appropriate rewards to states, according to what they want the agent to achieve in the environment.

However, any policy could actually be expressed in the form $\pi = RL(r)$ with an appropriately constructed reward function r, because of an argument analogous to the one that Shah showed in this other post. The relevant element for goal-directedness is the algorithmic complexity, compared to the environment size, of the reward function.

If the algorithmic procedure p has low complexity, then I expect that

$$K(g(A, O, D)|D) = K(\pi|D) = K(RL|D) + K(p|D) \ll |O|$$

especially in large environments. As an example for this case, consider the policy that is the result of RL training on a maze with the same small negative reward assigned to every state except for the exit, which has reward 1 instead. For this reward function, $K(p|D)$ is low, since the procedure p is only about recognising the exit square of the maze given as input D. Moreover, the same exact procedure can be used to find an exiting policy in larger mazes.

The fact that low algorithmic complexity is a sign of goal-directed behaviour has an analogy in natural language. When we say the goal of an agent is to exit mazes, we are giving a short description of its policies across multiple environments, no matter how big they are. In other words, we are expressing the policies in a compressed form by using the simple reward function, which coincides with the goal of the agent.

On the other hand, consider a reward function that assigns lots of different values to all states in the maze, with no recognisable pattern. In this case, the algorithmic complexity of the reward function is approximately equal to the size of the observation set: the situation is the same as with incompressible lookup tables.

In the analogy with natural language, this corresponds to the fact that the only way of describing such policies would be to state what the action (or the reward) is for each observation. There would be no goal which we can use to give a short description of the agent behaviour.

The lookup table vs simple-reward RL contrast appears under different forms in other contexts as well. Consider search processes (in this case, $\pi \in \{0, 1\}^{O}$) that select one or more elements in a search space O and take as input some data D about the elements. Manually specified filters, that for each search space list which elements to take and which to discard, are generally as complex as the size of the search space, thus essentially identical to lookup tables and without a recognisable goal.

On the other hand, a search algorithm that uses the data D to generate an ordering of the elements according to a simple evaluating function, and takes the best element, is succinctly described by the evaluating function itself. Moreover, the latter search process naturally extends to larger environments, while the filter needs one more specification for each element that is added to the search space.

Related work

The previous example with search processes smoothly leads to the consideration of an alternative formalism to standard optimisation: quantilizers. Briefly, instead of taking the best element in the ordering generated by the evaluation function, an element is chosen from the top q portion of the same ordering, according to a probability distribution.

If a standard optimisation process, like the previous example, is described as $\pi = \text{best}(\text{eval}(D))$, then a quantilizer can be expressed similarly as $\pi \in \text{top}_q(\text{eval}(D))$, with a negligible change in complexity.

In terms of goal-directedness, this corresponds to the fact that the two agents can be said to have more or less the same goal, since they are interested in the same property captured by the evaluating function. The difference lies in the degree of “directedness”: the first agent applies straightforward hard optimisation, while the quantilizer is, in a sense, more relaxed and possesses some safety properties (e.g. Lemma 1 in the original paper).

A similar comparison in the context of learning can be done between the optimal policy maximising a simple reward function, and a policy that does the same $1 - q$ of the time but takes a default action q of the time—something like waiting for new human instructions. These two agents have the same goal, but they pursue it differently, since the second one leaves more room for corrections.

Overall, when comparing agents with the same goal, it seems that less-directed agents trade a certain amount of performance for a gain in terms of safety.

Compression of complex policies is also listed as a favourable condition for mesa-optimisation (see section 2 in the paper). When searching for algorithms that can solve a certain class of problems, a bias in favour of short policies increases the likelihood that an algorithm which is itself an optimiser is chosen; ideally, if the bias for simplicity is strong enough, it may be possible to find an algorithm that generalises to problems outside of the original class. Unsurprisingly, such an algorithm is also more likely to be goal-directed.

Implications for safety

When the full policy of the agent can be described as the result of a relatively short algorithm applied to some short data representing the goal, as in the case of $\pi = RL(p(D))$, we can interpret p as a compressor that selects the goal-related data from all the environmental data D, and RL as a decompressor that uses the goal-related data to generate the full policy. Thus, because of error propagation, we should expect arbitrarily large errors in the full policy if there is any kind of error in the specification of the goal-related data.

This slightly formal reasoning reflects the less formal argument that, if we design a powerful goal-directed AI supposed to act in large and varied environments, and we make a mistake in the specification of the goal, the resulting policy could be arbitrarily far from what we originally expected or wanted.

A fundamental safety question is whether it's possible to design safe agents that have goals and act in the real world. We've seen before that, among agents with the same goal, some are safer than others, but it seems hard to tell if this is enough to avoid all possible bad outcomes.

Note, however, that even if "standard" goal-directed agents were unsafe, an alternative solution could be to design agents that still use some compression, but also have a certain amount of explicitly-specified ad-hoc behaviour for unique scenarios that wouldn't be handled correctly otherwise. Such an agent would be an intermediate design between the two extremes shown before, i.e. lookup tables and agents with a clear goal.

Thanks to Adam Shimi, Joe Collman and Sabrina Tang for feedback.

This work was supported by CEEALAR (EA hotel).

Appendix

- The environmental data D is underspecified and not completely formal, but the main ideas in the post should be clear enough anyway, so that it's easy to criticise them or suggest new research directions.
- Even though I showed the simpler case with deterministic policies over states, the reasoning is the same with stochastic policies over histories.

- The use of Kolmogorov complexity actually requires fixing a Universal Turing Machine: the above analysis doesn’t change at its core. You can think of all mentioned algorithms as if they were written in the same programming language.
- Kosoy has proposed a definition of goal-directed intelligence that also uses algorithmic complexity, but in a different way.

Goal-Directedness: What Success Looks Like

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

This sequence already contains a couple of blog posts, exploring different aspects of goal-directedness. But one question has never been fully addressed: what constraints should a good formalization of goal-directedness satisfy? An answer is useful both for people like me which study this topic, and for people trying to assess the value of this research. The following is my personal view, as always informed with discussion with Michele Campolo, Joe Collman and Sabrina Tang.

So what makes a good formalization of goal-directedness? The first part comes from what it means to formalize a set of philosophical intuitions. On that front, I agree with Vanessa in her research agenda:

> Although I do not claim a fully general solution to metaphilosophy, I think that, pragmatically, a quasiscientific approach is possible. In science, we prefer theories that are (i) simple (Occam's razor) and (ii) fit the empirical data. We also test theories by gathering further empirical data. In philosophy, we can likewise prefer theories that are (i) simple and (ii) fit intuition in situations where intuition feels reliable (i.e. situations that are simple, familiar or received considerable analysis and reflection). We can also test theories by applying them to new situations and trying to see whether the answer becomes intuitive after sufficient reflection.

So the first step is to fit the main intuitions about goal-directedness. This is the point of ideas like focus, short descriptions, and locality. The core intuitions should probably emerge from a community discussion, such that a consensus is reached. Then, fitting the intuitions looks like an optimization problem: for each formalization, there is a "distance" to the intuitions. The point is to minimize this distance, or at least come close to the minimum.

Yet this is not a pure philosophical endeavor. I study goal-directedness to understand if Rohin's position in this sequence of posts is right. I want to know if we as a community should invest time and efforts into less goal-directed approaches. This depends on two propositions by Rohin: that a less goal-directed system is not necessarily trivial or uncompetitive; and that being less goal-directed removes some safety issues, like convergent instrumental subgoals and wireheading.

Thus our initial optimization problem is actually a constrained optimization problem: minimizing the distance to the intuitions, while ensuring that less-goal directed systems are not necessarily trivial and that they don't suffer from the aforementioned safety issues.

Now we have a clean description of the problem. And this entails two success modes in the limit: either finding a good enough solution to the optimization problem (positive answer), or showing that no feasible solution is good enough to capture the intuitions (negative answer). The first case would vindicate Rohin and justify a research investment into the less goal-directed approaches. The second case would tell us that if there is a concept satisfying the constraints, it is not really linked to goal-directedness. Obviously, we might fail to reach either limit. But providing good evidence for one or the other would already be a big step.

Viewing the study of goal-directedness in these terms also informs how I go about it: by focusing on the optimization, while regularly checking for the constraints. Without an attempt at formalization, I don't know how to study the non triviality and the safety questions. I thus try to address as much intuitions as possible, and then see if the resulting theory survives contact with the constraints. Then I adapt the theory in response.

Rinse and repeat.