The Steering Problem

Paul Christiano Follow
Mar 14, 2015 · 22 min read

Most AI research focuses on reproducing human abilities: to learn, infer, and reason; to perceive, plan, and predict. There is a complementary problem which (understandably) receives much less attention: if you *had* these abilities, what would you do with them?

The steering problem: Using black-box access to human-level cognitive abilities, can we write a program that is as useful as a well-motivated human with those abilities?

This document explains what the steering problem is and why I think it's worth spending time on.

1. Introduction

A capable, well-motivated human can be extremely useful: they can work without oversight, produce results that need not be double-checked, and work towards goals that aren't precisely defined. These capabilities are critical in domains where decisions cannot be easily supervised, whether because they are too fast, too complex, or too numerous.

In some sense "be as useful as possible" is just another task at which a machine might reach human-level performance. But it is different from the concrete capabilities normally considered in AI research.

We can say clearly what it means to "predict well," "plan well," or "reason well." If we ignored computational limits, machines could achieve any of these goals today. And before the existing vision of AI is realized, we must *necessarily* achieve each of these goals.

For now, "be as useful as possible" is in a different category. We can't say exactly what it means. We could not do it no matter how fast our computers could compute. And even if we resolved the most salient challenges in AI, we could remain in the dark about this one.

Consider a capable AI tasked with running an academic conference. How should it use its capabilities to make decisions?

- We could try to specify exactly what makes a conference good or bad. But our requirements are complex and varied, and so specifying them exactly seems time-consuming or impossible.
- We could build an AI that imitates successful conference organizers. But this approach can never do any better than the humans we are imitating. Realistically, it won't even match human performance unless we somehow communicate what characteristics are important and why.
- We could ask an AI to maximize our satisfaction with the conference. But we'll get what we measure. An extensive evaluation would greatly increase the cost of the conference, while a superficial evaluation would leave us with a conference optimized for superficial metrics. Everyday experience with humans shows how hard delegation can be, and how much easier it is to assign a task to someone who actually cares about the outcome.

Of course there is already pressure to write *useful* programs in addition to smart programs, and some AI research studies how to efficiently and robustly communicate desired behaviors. For now, available solutions apply only in limited domains or to weak agents. The steering problem is to close this gap.

Motivation

A system which "merely" predicted well would be extraordinarily useful. Why does it matter whether we know how to make a system which is "as useful as possible"?

Our machines will probably do *some* things very effectively. We know what it means to "act well" in the service of a given goal. For example, using human cognitive abilities as a black box, we could probably design autonomous corporations which very effectively maximized growth. If the black box was cheaper than the real thing, such autonomous corporations could displace their conventional competitors.

If machines can do everything equally well, then this would be great news. If not, society's direction may be profoundly influenced by what can and cannot be done easily. For example, if we can only maximize what we can precisely define, we may inadvertently end up with a world filled with machines trying their hardest to build bigger factories and better widgets, uninterested in anything we consider intrinsically valuable.

All technologies are more useful for some tasks than others, but machine intelligence might be particularly problematic because it can entrench itself. For example, a rational profit-maximizing corporation might distribute itself throughout the world, pay people to help protect it, make well-crafted moral appeals for equal treatment, or campaign to change policy. Although such corporations could bring large benefits in the short term, in the long run they may be difficult or impossible to uproot, even once they serve no one's interests.

Why now?

Reproducing human abilities gets a lot of deserved attention. Figuring out exactly what you'd do once you succeed feels like planning the celebration before the victory: it might be interesting, but why can't it wait?

1. **Maybe it's hard**. Probably the steering problem is much easier than the AI problem, but it might turn out to be surprisingly difficult. If it *is* difficult, then learning that earlier will help us think more clearly about AI, and give us a head start on addressing the steering problem.

2. **It may help us understand AI.** The difficulty of saying exactly what you want is a basic challenge, and the steering problem is a natural perspective on this challenge. A little bit of research on natural theoretical problems is often worthwhile, even when the direct applications are limited or unclear. In section 4 we discuss possible approaches to the steering problem, many of which are new perspectives on important problems.

3. **It should be developed alongside AI.** The steering problem is a long-term goal in the same way that understanding human-level prediction is a long-term goal. Just as we do theoretical research on prediction before that research is commercially relevant, it may be sensible to do theoretical research on steering before it is commercially relevant. Ideally, our ability to build useful systems will grow in parallel with our ability to build capable systems.

4. **Nine women can't make a baby in one month.** We could try to save resources by postponing work on the steering problem until it seems important. At this point it will be easier to work on the steering problem, and if the steering problem turns out to be unimportant then we can avoid thinking about it

altogether. But at large scales it becomes hard to speed up progress by increasing the number of researchers. Fewer people working for longer may ultimately be more efficient even if earlier researchers are at a disadvantage. In general, scaling up fields rapidly is difficult.

5. **AI progress may be surprising**. We probably won't reproduce human abilities in the next few decades, and we probably won't do it without ample advance notice. That said, AI is too young, and our understanding too shaky, to make confident predictions. A mere 15 years is 20% of the history of modern computing. If important human-level capabilities are developed surprisingly early or rapidly, then it would be worthwhile to better understand the implications in advance.

6. **The field is sparse**. Because the steering problem and similar questions have received so little attention, individual researchers are likely to make rapid headway. There are perhaps three to four orders of magnitude between basic research on AI and research directly relevant to the steering problem, lowering the bar for arguments 1–5.

In section 3 we discuss some other reasons not to work on the steering problem: Is work done now likely to be relevant? Is there any concrete work to do now? Should we wait until we can do experiments? Are there adequate incentives to resolve this problem already?

2. Defining the problem precisely

Recall our problem statement:

The steering problem: Using black-box access to human-level cognitive abilities, can we write a program that is as useful as a well-motivated human with those abilities?

We'll adopt a particular human, Hugh, as our "well-motivated human:" we'll assume that we have black-box access to Hugh-level cognitive abilities, and we'll try to write a program which is as useful as Hugh.

Abilities

In reality, AI research yields complicated sets of related abilities, with rich internal structure and no simple performance guarantees. But in order to do concrete work in advance, we will model abilities as black boxes with well-defined contracts.

We're particularly interested in tasks which are "AI complete" in the sense that human-level performance on that task could be used as a black box to achieve human-level performance on a very wide range of tasks. For now, we'll further focus on domains where performance can be unambiguously defined.

Some examples:

- **Boolean question-answering**. A question-answerer is given a statement and outputs a probability. A question-answerer is Hugh-level if it never makes judgments predictably worse than Hugh's. We can consider question-answerers in a variety of languages, ranging from natural language ("Will a third party win the US presidency in 2016?") to precise algorithmic specifications ("Will this program output 1?").

- **Online learning**. A function learner is given a sequence of labelled examples (x, y) and predicts the label of a new data point, x'. A function learner is Hugh-level if, after training on any sequence of data (x_i, y_i), the learner's guess for the label of the next point is—on average—at least as good as Hugh's.

- **Embodied reinforcement learning**. A reinforcement learner interacts with an environment and receives periodic rewards, with the goal of maximizing the discounted sum of its rewards. A reinforcement learner is Hugh-level if, following any sequence of observations, it achieves an *expected* performance as good as Hugh's in the subsequent rounds. The expectation is taken using our subjective distribution over the physical situation of an agent who has made those observations.

When talking about Hugh's predictions, judgments, or decisions, we imagine that Hugh has access to a reasonably powerful computer, which he can use to process or display data. For example, if Hugh is given the binary data from a camera, he can render it on a screen in order to make predictions about it.

We can also consider a particularly degenerate ability:

- **Unlimited computation**. A box that can run any algorithm in a single time step is—in some sense—Hugh level at every precisely stated task.

Although unlimited computation seems exceptionally powerful, it's not immediately clear how to solve the steering problem even using such an extreme ability.

Measuring usefulness

What does it mean for a program to be "as useful" as Hugh?

We'll start by defining "as useful for X as Hugh," and then we will informally say that a program is "as useful" as Hugh if it's as useful for the tasks we care most about.

Consider **H,** a black box which simulates Hugh or perhaps consults a version of Hugh who is working remotely. We'll suppose that running **H** takes the same amount of time as consulting our Hugh-level black boxes. A project to accomplish X could potentially use as many copies of **H** as it can afford to run.

A program **P** is as useful than Hugh for X if, for every project using **H** to accomplish X, we can efficiently transform it into a new project which uses **P** to accomplish X. The new project shouldn't be much more expensive—-it shouldn't take much longer, use much more computation or many additional resources, involve much more human labor, or have significant additional side-effects.

Well-motivated

What it does it mean for Hugh to be well-motivated?

The easiest approach is universal quantification: for *any* human Hugh, if we run our program using Hugh-level black boxes, it should be as useful as Hugh.

Alternatively, we can leverage our intuitive sense of what it means for someone to be well-motivated to do X, and define "well-motivated" to mean "motivated to help the user's project succeed."

Scaling up

If we are given better black boxes, we should make a better program. This is captured by the requirement that our program should be as useful as Hugh, no matter how capable Hugh is (as long as the black boxes are equally capable).

Ideally, our solutions should scale far past human-level abilities. This is not a theoretical concern—in many domains computers already have significantly superhuman abilities. This requirement is harder to make precise, because we can no longer talk about the "human benchmark." But in general, we would like to build systems which are (1) working towards their owner's interests, and (2) nearly as effective as the best goal-directed systems that can be built using the available abilities. The ideal solution to the steering problem will have these characteristics in general, even when the black-box abilities are radically superhuman.

Scaling down

"Human-level abilities" could refer to many different things, including:

1. Human-level performance on high-level tasks.
2. The level of functionality embodied in the human brain. Human-level perception, intuition, motor control, subsymbolic reasoning, and so on.

In general, as we shift from 1 towards 2 the steering problem becomes more difficult. It may be difficult to produce simple or predictable high-level functions using low-level abilities.

For example, humans pursue a complicated set of goals that would be very difficult to determine by looking at the human brain (and some of which are quite distant from the evolutionary pressures that produced us). When given a task that doesn't serve these goals, a human may simply decide to pursue their own agenda. If we build human-like abilities out of human-like low-level functions, we may find ourselves with similarly unpredictable high-level functions.

It is harder to formalize or understand low-level abilities than high-level functions. One approach is to consider very short time periods. For example, we could consider black boxes which learn functions as well as a human who spends only 500 milliseconds per example. Unfortunately, at this level it is harder to encapsulate human abilities in a small number of simple functions, and we must pay more attention to the way in which these abilities can be connected.

If the steering problem were satisfactorily resolved, "scaling down" to these lower-level abilities would be a natural but challenging next step.

3. Objections

The simple, abstract capabilities we can think of now are much harder to use productively than the rich and messy AI capabilities we will actually develop.

For now we can't clearly state *anything* a machine could do that would make the steering problem easy (short of exactly reproducing human behavior). Filling in this gap would be an appropriate response to the steering problem.

Perhaps we don't yet know exactly what we want machines to do, but figuring it out is inextricably bound up with getting them to do it. If so, it might be easier to say what we want once we know how to do it. But by the same token, it might be easier to figure out how to do it once we can better say what we want.

In either case, it seems likely that the steering problem fills in its own niche: either it is a distinct problem that won't be solved automatically en route to AI; or else it is a different perspective on the same underlying difficulties, and can be productively explored in parallel with other AI research.

Because the steering problem is non-trivial for simple, precisely stated abilities, it may well be non-trivial for the abilities we actually obtain. Certainly we can imagine developing a human-level predictor without learning too much about how to build useful systems. So it seems unreasonable to be confident that the steering problem will turn out to be a non-problem.

The simple, abstract abilities we can think of now are much *easier* to work with than the human-level abilities we will actually develop, or at least much different. Building a robust system is easier when all of the pieces have clean, reliable functions; in practice things won't be so pretty.

It would be a larger leap to continue "…and the ideas required to work with simple, reliable components will have no relevance to their more realistic counterparts." *Whatever* abilities we end up with, many solutions to the steering problem will turn out to be inapplicable, and they will all be incomplete. But we can still find useful general techniques by developing ideas that are helpful for many versions of the steering problem; and we can identify important technical challenges by understanding what makes each version easy or hard.

We can gradually scale up the difficulty of the steering problem by demanding more robust solutions, making weaker guarantees on our black boxes, or working with less manageable abilities. Our choices can be informed by ongoing progress in AI, focusing on those capabilities we think are most realistic and the forms of robustness we consider most likely to be necessary.

Why is autonomy necessary?

One apparent solution to the steering problem is to retain human decision-making, with AI systems acting as assistants and tools to help humans accomplish their goals.

This is an appropriate solution while AI systems remain relatively limited. It has serious problems when scaling:

- If large numbers of machines make large numbers of decisions, with human wages orders of magnitude larger than the operating costs of machines, then the cost of human oversight becomes prohibitive. Imagine a million humans overseeing a billion or trillion human-level machines.
- If machines make very rapid decisions, human oversight can introduce unacceptable latency. Imagine human engineers overseeing the handling of individual Google searches.
- If machines work on complex problems, human overseers may not be able to understand their reasoning process. Imagine a physics undergraduate overseeing a team of world-class physicists.

All of these problems become particularly severe when we consider *thinking about thinking*. That is, machines must make numerous, rapid decisions about how to process information, what to investigate or compute, how to organize their resources, and so on. If we want to use machine intelligence to make those decisions better, that will have to be done without substantial human oversight.

It may be possible to maintain human involvement in all important automation, but doing so will eventually become a serious bottleneck. Tasks that can be performed without human oversight will become increasingly efficient, and without explicit coordination (and a willingness to make short-term sacrifices) it seems likely that more autonomous operations will outcompete their less autonomous counterparts.

Is there any concrete work to do on the steering problem?

In the next section I'll describe a handful of existing research directions that bear on the steering problem. I think the steering problem suggests an interesting and unusual perspective on each of these domains; I don't know whether it will prove to be a fruitful perspective, but if it fails it won't be because of a lack of first steps.

I have done some work motivated explicitly by the steering problem: a formalization of "judgment upon reflection," which can be expressed entirely algorithmically based on (experimentally controlled) observations of human behavior, an alternative to goal-directed behavior which may enjoy similar productivity benefits while being more robust, and some simple protocols for delegating to untrusted agents.

4. Approaches, ingredients, and related work

Rational agency

One natural approach to the steering problem is to build goal-directed agents who want to be useful or who share their creators' goals.

There are two main difficulties:

- Specifying goals in an appropriate language. What does it mean to "be useful"? How can we define what we want?
- Building agents that reliably pursue goals specified in that language.

Deploying a goal-directed agent is somewhat worrying: an agent with an almost-but-not-quite-correct goal will be working at cross-purposes to its creator, and will be motivated (for example) to avoid revealing that its goal is not quite correct. These concerns motivate a third line of research:

- Designing goals or goal-directed agents which "fail gracefully," i.e. which don't behave adversarially or resist correction, even if their goals are not perfectly aligned with their creators'.

Several lines of existing research bear on each of these questions.

Specifying goals

Rather than directly specifying what outcomes are good, it seems more promising to specify how to learn what outcomes are good. This is a topic of existing research, although the focus is typically on pragmatic considerations rather than on the more general theoretical problem.

- Inverse reinforcement learning (for example see Russell, Ng and Russell, or Ziebart et al.) and goal inference (for example see Baker, Tenenbaum, and Saxe or Verma and Rao) attempt to infer underlying preferences by observing behavior, despite a complex relationship between actions and outcomes. To apply to the steering problem, the techniques would need to be generalized to learners who are much better informed and more capable than the human models they are learning from, and who have much noisier information about the human models and their environment. This requires understanding the limitations and errors of the human models, and generalizing the human's goals robustly so that they remain acceptable even when they are pursued in an unfamiliar way.

- Preference learning attempts to infer underlying preferences from observed decisions, despite noisy information and potentially irrational behavior (for a small sample, see Fürnkranz and Hüllermeier, Fürnkranz and Hüllermeier, or Gervasio et al.). Existing work considers small domains with explicitly represented preferences, and there seem to be serious challenges when scaling to preferences over complete states-of-affairs. As a result, preference learning seems less directly applicable than inverse reinforcement learning or goal inference.

- Some more speculative and philosophical research (for example, see my post on the subject, or this more discursive article by Yudkowsky) has explored how to formalize our preferences in a general and precise way. The focus is on determining what processes of deliberation correctly capture our informed judgment and how we might formalize those processes. The primary challenge, on this perspective, is defining our preferences about outcomes that are difficult for us to describe or reason about.

We could also investigate goals of the form "maximize user satisfaction," but it seems hard to find a satisfactory definition along these lines.

Pursuing goals

Even with desirable goals in hand, it may be challenging to design systems that reliably pursue those goals. There are questions about how goal-directed behavior relates to reasoning and to the behavior of subsystems (does the system pursue the goals it appears to?), about the theoretical basis for optimal rational behavior (does it pursue them well?), and about how an agent should behave in light of uncertainty about what outcomes are desirable.

- Some existing work in reinforcement learning (for a summary, see Sutton and Barto) and probabilistic inference (for example, see Attas) shows how goal-directed behavior can be implemented using other faculties.

- Some work in philosophy studies the formal basis for rational agency, from decision theory to epistemology. This work clarifies what it means to rationally pursue a particular goal. A lack of understanding may result in systems that appear to pursue one goal but actually pursue another, or that generalize to novel environments in undesirable ways.

- Some work on AI safety (see Dewey or Bostrom, ch. 12) explores frameworks for pursuing uncertain goals, in the interest of understanding how and to what extent “learning what is good” is different from “learning what is true.”

- Some work in multi-objective optimization considers settings with a large space of possible objectives, and seeks policies which are appropriate in light of uncertainty about which objective we really care about.

Failing gracefully

Even a “near miss” when defining a goal-directed agent might have undesirable consequences. In addition to minimizing the probability of failure, it would be nice to minimize the costs of a near miss—-and to allow us to use the kind of “trial and error” approach that is more typical of software development.

- Researchers interested in AI safety have introduced and discussed the notion of “corrigible” agents, who cooperate with “corrective” interventions by their programmers—even if we cannot implement goals consistent with that cooperation.

- Some researchers have worked to make AI reasoning understandable (For example, see Bullinaria or Craven). Understandability can reduce the scope for malignant failure modes, since engineers might be able to directly monitor the motivation for decisions (and in particular to distinguish between honest and deceptive behavior).

Delegation

Every day humans work productively on projects that they don't intrinsically care about, motivated by a desire for money, recognition, or satisfaction. We could imagine an AI doing the same thing. For example, a reinforcement learner might do useful work in order to earn a reward, without having any intrinsic concern for the work being done.

Unfortunately, such delegation runs into some problems. The problems appear even when delegating to humans, but they get considerably worse as machines become more powerful and more numerous:

- Naively, the agent will only do good work when the principal can verify the quality of that work. The cost of this oversight can be non-trivial.
- Unless the oversight is extremely thorough or the problem particularly straightforward, there will be some gap between what the principal wants and what the principal evaluates. The reward-driven agent will maximize whatever the principal evaluates.
- If agents are granted much autonomy, they may find other ways to get rewards. This depends in detail on how the "reward" is implemented, and what the agent cares about.

There are many tools available to address these problems: breaking a system up into pieces with differing values and limited autonomy, performing randomized audits, automating audits and auditing auditors, relying on agents with short time horizons or extreme risk aversion, and so on. So far there are no compelling proposals that put the pieces together.

A full solution is likely to rely on agents with different values, combined with an appropriate system of checks and balances. But if the agents coordinate to pursue their collective values, they could fatally undermine such a system. We can try to minimize the risk by making the agents' interactions essentially zero sum, or employing other agents to oversee interactions and report signs of collusion. But the possibility of collusion remains a serious obstacle.

Even if delegation cannot fully resolve the steering problem, a weak solution might be useful as part of a bootstrapping protocol (see the section on bootstrapping below).

This problem is similar in spirit to mechanism design, but the details (and apparently the required tools) are quite different. Nevertheless, some ideas from mechanism design or the economics of delegation may turn out to be applicable. Conversely, some approaches to the steering problem might be of interest to economists in these areas.

Shared language and concepts

When delegating to a helpful human, we would say what we want done in natural language, relying on a rich network of shared concepts that can be used to specify goals or desired actions. Writing programs with the same capability would greatly simplify or perhaps solve the steering problem.

In some sense, human-level language understanding is already encapsulated in human-level cognitive abilities. For example, if we were pursuing the delegation approach in the last section, we could describe tasks in natural language. The agents would infer what we expect them to do and under what conditions we will give them rewards, and they would behave appropriately in light of that knowledge. But this "language understanding" only appears in the agent's goal-directed behavior.

To address the steering problem, we would like something stronger. We would like to build agents that share human concepts, such that we can write code that operates in terms of those concepts: specifying a goal in terms of higher-level concepts, or executing instructions defined in terms of these concepts. These tasks don't seem to be possible using only goal-directed language understanding.

Understanding concept learning and the relationship to language is a fundamental problem in cognitive science and AI. Work in these areas thus bears directly on the steering problem.

For now, we cannot say formally what it would mean to have a program that reliably acquired "the same" concepts as humans, so that instructions expressed in those concepts would have the intended meaning. Even given unlimited computation, it's not clear how we would solve the steering problem using concept learning. This is not at all to say it is not possible, merely that it has not yet been done.

There may be a tight connection between the theoretical question—-what would it mean to learn human concepts, and how could you do it with any amount of computation—-and the pragmatic computational issues. If there is a connection, then the theoretical question might be easier once the pragmatic issues are better understood. But conversely, the pragmatic question might also be easier once the theoretical issues are better understood.

Non-consequentialist intelligence

If describing our real goals is too demanding, and describing a crude approximation is hazardous, then we might try to build systems without explicitly defined goals. This makes the safety problem much easier, but probably makes it harder to build systems which are sufficiently powerful at all.

Non-agents

One idea is to focus on systems with some narrower function: answering questions, proposing plans, executing a narrow task, or so on. On their own these systems might not be terribly useful, but the hope is that as tools they can be nearly as useful as a goal-directed assistant. Moreover, because these systems don't need to be aligned with human goals, they may be easier to construct.

For example, research in decision support and multi-objective optimization aims to find good solutions to an optimization problem (or a planning problem) by interacting with a decision-maker rather than giving a scalar representation of their preferences (for example, see Fonseca and Fleming or Deb).

These systems can certainly be useful (and so may be appropriate as part of a bootstrapping strategy or as a component in a more complex solution; see the next sections). But most of them inherently require significant human oversight, and do not seem suitable as solutions to the full steering problem: for projects involving large numbers of agents with a small number of human overseers, the involvement of human oversight in all substantive decisions is probably an unacceptable overhead.

This issue applies at every level simultaneously, and seems particularly serious when we consider systems thinking about how to think. For example, in order to produce a plan, a simulated human or team would first plan how to plan. They would seek out relevant information, talk to people with necessary expertise, focus their attention on the highest-priority questions, allocate available computational resources, and so on. If human oversight is necessary for every step of this process, the resulting system is not even as useful as a black box that outputs good plans.

Of course, even if humans rely on goal-directed behavior to accomplish these tasks, this doesn't imply that machines must as well. But that would require a concrete alternative approach that still captures the benefits of goal-directed reasoning without substantial oversight.

Non-consequentialist agents

Instead we might build agents with no explicitly defined goals which can nevertheless accomplish the same tasks as a helpful human.

Perhaps most straightforward is an agent that asks "What would a helpful human do?" and then does that. If we have a particular helpful human available as a template, we could build a predictive model of that human template's decisions, and use this model to guide our agent's decisions. With a good enough model, the result would be precisely as useful as the human template.

This proposal has a number of potential problems. First, it does not scale to the available abilities—-it is never more useful than a simulation of the human template. So it is not a general solution to the steering problem, unless we have access to arbitrarily capable human templates. Second, if our predictions are imperfect, the behavior may be significantly worse than the helpful human template. Third, making accurate predictions about a human is itself a superhuman skill, and so asking Alice to do what she thinks Bob would do can result in behavior worse than Alice would produce on her own, no matter how smart Bob is.

We can address some of these problems by instead asking "What choice would the human overseer most approve of?" and then taking the most-approved-of option. And a bootstrapping procedure can address some limitations of using a human template. I explore these ideas here.

Research on inverse reinforcement learning or goal inference could also be used to learn imitative behavior which is sensitive to human goals, rather than to build systems that infer a human's long-term goals.

There may be many other alternatives to goal-directed behavior. Any proposal would probably be combined with some ideas under "rational agency" and "shared language" above.

Bootstrapping

We might try to deal with two cases separately:

- We are dealing with machines *much* smarter, faster, or more numerous than humans.
- We aren't.

In the first case, we could try to delegate the steering problem to the much more capable machines. In the second case, we might be able to make do with a weak solution to the steering problem which wouldn't scale to more challenging cases.

It's hard to know how this strategy would work out, and to a greater extent than the other approaches we would expect to play it by ear. But by thinking about some of the difficulties in advance, we can get a better sense for how hard the steering problem will actually be, and whether a strong solution is necessary.

Roughly, there are two pieces to a bootstrapping protocol:

1. We need to solve a weak version of the steering problem. We don't have to make our machine as useful as a human, but we do need to get *some* useful work, beyond what we could have done ourselves. Ideally we'll come as close as possible to a useful human.
2. We need to use the available machines to solve a stronger version of the steering problem. This is repeated until we've solved the full problem.

Both of these steps would rely on the kinds of techniques we have discussed throughout this section. The difference is that the humans, as well as the machine intelligences, only need to solve the steering problem for machines somewhat smarter or faster than themselves.

One implication is that weak solutions to the steering problem might be amplified into strong solutions, and so are worth considering even if they can’t be scaled up to strong solutions directly. This includes many of the approaches listed under “Delegation” and “Goal-less intelligence.”