Epistemic Cookbook for Alignment

1. On Solving Problems Before They Appear: The Weird Epistemologies of Alignment
2. Epistemic Strategies of Selection Theorems
3. Epistemic Strategies of Safety-Capabilities Tradeoffs
4. Interpreting Yudkowsky on Deep vs Shallow Knowledge
5. Redwood's Technique-Focused Epistemic Strategy

On Solving Problems Before They Appear: The Weird Epistemologies of Alignment

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

Crossposted to the EA Forum

Introduction

Imagine you are tasked with curing a disease which hasn't appeared yet. Setting aside why you would know about such a disease's emergence in the future, how would you go about curing it? You can't directly or indirectly gather data about the disease itself, as it doesn't exist yet; you can't try new drugs to see if they work, as **the disease doesn't exist yet**; you can't do anything that is experimental in any way on the disease itself, as… you get the gist. You would be forgiven for thinking that there is nothing you can do about it right now.

AI Alignment looks even more hopeless: it's about solving a never-before seen problem on a technology which doesn't exist yet.

Yet researchers are actually working on it! There are papers, books, unconferences, whole online forums full of alignment research, both conceptual and applied. Some of these work in a purely abstract and theoretical realm, others study our best current analogous of Human Level AI/Transformative AI/AGI, still others iterate on current technologies that seem precursors to these kinds of AI, typically large language models. With so many different approaches, how can we know if we're making progress or just grasping at straws?

Intuitively, the latter two approaches sound more like how we should produce knowledge: they take their **epistemic strategies** (ways of producing knowledge) out of Science and Engineering, the two cornerstones of knowledge and technology in the modern world. Yet recall that in alignment, models of the actual problem and/or technology can't be evaluated experimentally, and one cannot try and iterate on proposed solutions directly. So when we take inspiration from Science and Engineering (and I want people to do that), we must be careful and realize that most of the guarantees and checks we associate with both are simply not present in alignment, for the decidedly pedestrian reason that Human Level AI/Transformative AI/AGI doesn't exist yet.

I thus claim that:

- Epistemic strategies from Science and Engineering don't dominate other strategies in alignment research.
- Given the hardness of grounding knowledge in alignment, we should leverage every epistemic strategy we can find and get away with.
- These epistemic strategies should be made explicit and examined. Both the ones taken or adapted from Science and Engineering, and every other one (for example the more theoretical and philosophical strategies favored in conceptual alignment research).

This matters a lot, as it underlies many issues and confusions in how alignment is discussed, taught, created and criticized. Having such a diverse array of epistemic strategies is fascinating, but their implicit nature makes it challenging to communicate with newcomers, outsiders, and even fellow researchers leveraging different strategies. Here is a non-exhaustive list of issues that boil down to epistemic strategy confusion:

- There is a strong natural bias towards believing that taking epistemic strategies from Science and Engineering automatically leads to work that is valuable for alignment.
    - This leads to a flurry of work (often by well-meaning ML researchers/engineers) that doesn't tackle or help with alignment, and might push capabilities (in an imagined tradeoff with the assumed alignment benefits of the work)
    - Very important: that doesn't mean I consider work based on ML as useless for alignment. Just that the sort of ML-based work that actually tries to solve the problem tends to be by people who understand that one must be careful when transfering epistemic strategies to alignment.
- There is a strong natural bias towards disparaging any epistemic strategy that doesn't align neatly with the main strategies of Science and Engineering (even when the alternative epistemic strategies are actually used by scientists and engineers in practice!)
    - This leads to more and more common confusion about what's happening on the Alignment Forum and conceptual alignment research more generally. And that confusion can easily turn into the sort of criticism that boils down to "this is stupid and people should stop doing that".
- It's quite hard to evaluate whether alignment research (applied or conceptual) creates the kind of knowledge we're after, and helps move the ball forward. This comes from both the varieties of epistemic strategies and the lack of the usual guarantees and checks when applying more mainstream ones (every observation and experiment is used through analogies or induction to future regimes).
    - This makes it harder for researchers using different sets of epistemic strategies to talk to each other and give useful feedback to one another.
- Criticism of some approach or idea often stops at the level of the epistemic strategy being weird.
    - This happens a lot with criticism of lack of concreteness and formalism and grounding for Alignment Forum posts.
    - It also happens when applied alignment research is rebuked solely because it uses current technology, and the critics have decided that it can't apply to AGI-like regimes.
- Teaching alignment without tackling this pluralism of epistemic strategies, or by trying to fit everything into a paradigm using only a handful of those, results in my experience in people who know the lingo and some of the concepts, but have trouble contributing, criticizing and teaching the ideas and research they learnt.
    - You can also end up with a sort of dogmatism that alignment can only be done a certain way.

Note that most fields (including many sciences and engineering disciplines) also use weirder epistemic strategies. So do many attempts at predicting and directing the future (think existential risk mitigation in general). My point is not that alignment is a special snowflake, more that it's both weird enough (in the inability to experiment and iterate directly with) and important enough that elucidating the epistemic strategies we're using, finding others and integrating them is particularly important.

In the rest of this post, I develop and unfold my arguments in more detail. I start with digging deeper into what I mean by the main epistemic strategies of Science and Engineering, and why they don't transfer unharmed to alignment. Then I demonstrate the importance of looking at different epistemic strategies, by focusing on examples of alignment results and arguments which make most sense as interpreted through the epistemic lens of Theoretical Computer Science. I use the latter as an inspiration because I adore that field and because it's a fertile ground for epistemic strategies. I conclude by pointing at the sort of epistemic analyses I feel are needed right now.

Lastly, this post can be seen as a research agenda of sorts, as I'm already doing some of these epistemic analyses, and believe this is the most important use of my time and my nerdiness about weird epistemological tricks. We roll with what evolution's dice gave us.

Thanks to Logan Smith, Richard Ngo, Remmelt Ellen,, Alex Turner, Joe Collman, Ruby, Antonin Broi, Antoine de Scorraille, Florent Berthet, Maxime Riché, Steve Byrnes, John Wentworth and Connor Leahy for discussions and feedback on drafts.

Science and Engineering Walking on Eggshells

What is the standard way of learning how the world works? For centuries now, the answer has been Science.

I like Peter Godfrey-Smith's description in his glorious Theory and Reality:

> Science works by taking theoretical ideas and trying to find ways to expose them to observation. The scientific strategy is to construe ideas, to embed them in surrounding conceptual frameworks, and to develop them, in such a way that this exposure is possible even in the case of the most general and ambitious hypotheses about the universe.

That is, the essence of science is in the trinity of modelling, predicting and testing.

This doesn't mean there are no epistemological subtleties left in modern science; finding ways of gathering evidence, of forcing the meeting of model and reality, often takes incredibly creative turns. Fundamental Chemistry uses synthesis of molecules never seen in nature to test the edge cases of its models; black holes are not observable directly, but must be inferred by a host of indirect signals like the light released by matter falling in the black hole or gravitational waves during black holes merging; Ancient Rome is probed and explored through a discussion between textual analysis and archeological discoveries.

Yet all of these still amount to instantiating the meta epistemic strategy “say something about the world, then check if the world agrees”. As already pointed out, it doesn’t transfer straightforwardly to alignment because Human Level AI/Transformative AI/AGI doesn’t exist yet.

What I’m not saying is that epistemic strategies from Science are irrelevant to alignment. But because they must be adapted to tell us something about a phenomenon that doesn’t exist yet, they lose their supremacy in the creation of knowledge. They can help us to gather data about what exists now, and to think about the sort of models that are good at describing reality, but checking their relevance to the actual problem/thinking through the analogies requires thinking in more detail about what kind of knowledge we’re creating.

If we instead want more of a problem solving perspective, tinkering is a staple strategy in engineering, before we know how to solve the problem things reliably. Think about curing cancer or building the internet: you try the best solutions you can think of, see how they fail, correct the issues or find a new approach, and iterate.

Once again, this is made qualitatively different in alignment by the fact that neither the problem nor the source of the problem exist yet. We can try to solve toy versions of the problem, or what we consider analogous situations, but none of our solutions can be actually tested yet. And there is the additional difficulty that Human-level AI/Transformative AI/AGI might be so dangerous that we have only one chance to implement the solution.

So if we want to apply the essence of human technological progress, from agriculture to planes and computer programs, just trying things out, we need to deal with the epistemic subtleties and limits of analogies and toy problems.

An earlier draft presented the conclusion of this section as “Science and Engineering can’t do anything for us—what should we do?” which is not my point. What I’m claiming is that in alignment, the epistemic strategies from both Science and Engineering are not as straightforward to use and leverage as they usually are (yes, I know, there’s a lot of subtleties to Science and Engineering anyway). They don’t provide privileged approaches demanding minimal epistemic thinking in most cases; instead we have to be careful how we use them as epistemic strategies. Think of them as tools which are so good that most of the time, people can use them without thinking about all the details of how the tools work, and get mostly the wanted result. My claim here is that these tools need to be applied with significantly more care in alignment, where they lose their “basically works all the time” status.

Acknowledging that point is crucial for understanding why alignment research is so pluralistic in terms of epistemic strategies. Because no such strategy works as broadly as we’re used to for most of Science and Engineering, alignment has to draw from literally every epistemic strategy it can pull, taking inspiration from Science and Engineering, but also philosophy, pre-mortems of complex projects, and a number of other fields.

To show that further, I turn to some main alignment concepts which are often considered confusing and weird, in part because they don’t leverage the most common epistemic strategies. I explain these results by recontextualizing them through the lens of Theoretical Computer Science (TCS).

Epistemic Strategies from TCS

For those who don’t know the field, Theoretical Computer Science focuses on studying computation. It emerged from the work of Turing, Church, Gödel and others in the 30s, on formal models of what we would now call computation: the process of following a step-by-step recipe to solve a problem. TCS cares about what is computable and what isn’t, as well as how much resources are needed for each problem. Probably the most active subfield is

Complexity Theory, which cares about how to separate computational problems in classes capturing how many resources (most often time – number of steps) are required for solving them.

What makes TCS most relevant to us is that theoretical computer scientists excel at wringing knowledge out of the most improbable places. They are brilliant at inventing epistemic strategies, and remember, we need every one we can find for solving alignment.

To show you what I mean, let's look at three main ideas/arguments in alignment (Convergent Subgoals, Goodhart's Law and the Orthogonality Thesis) through some TCS epistemological strategies.

Convergent Subgoals and Smoothed Analysis

One of the main argument for AI Risk and statement of a problem in alignment is Nick Bostrom's Instrumental Convergence Thesis (which also takes inspiration from Steve Omohundro's Basic AI Drives):

> Several instrumental values can be identified which are convergent in the sense that their attainment would increase the chances of the agent's goal being realized for a wide range of final goals and a wide range of situations, implying that these instrumental values are likely to be pursued by many intelligent agents.

That is, actions/plans exist which help with a vast array of different tasks: self-preservation, protecting one's own goal, acquiring resources... So a Human-level AI/Transformative AI/AGI could take them while still doing what we asked it to do. Convergent subgoals are about showing that behaviors which look like they can only emerge from rebellious robots actually can be pretty useful for obedient (but unaligned in some way) AI.

What kind of argument is that? Bostrom makes a claim about "most goals"—that is, the space of all goals. His claim is that convergent subgoals are so useful that goal-space is almost chock-full of goals incentivizing convergent subgoals.

And more recent explorations of this argument have followed this intuition: Alex Turner et al.'s work on power-seeking formalizes the instrumental convergence thesis in the setting of Markov decision processes (MDP) and reward functions by looking, for every "goal" (a distribution over reward functions) at the set of all its permuted variants (the distribution given by exchanging some states – so the reward labels stay the same, but are not put on the same states). Their main theorems state that given some symmetry properties in the environment, a majority (or a possibly bigger fraction) of the permuted variant of **every** goal will incentivize convergent subgoals for its optimal policies.

So this tells us that goals without convergent subgoals exist, but they tend to be crowded out by ones with such subgoals. Still, it's very important to realize what neither Bostrom nor Turner are arguing for: they're not saying that every goal has convergent subgoals. Nor are they claiming to have found the way humans sample goal-space, such that their results imply goals with convergent subgoals must be sampled by humans with high probability. Instead, they show the overwhelmingness of convergent subgoals in some settings, and consider that a strong indicator that avoiding them is hard.

I see a direct analogy with the wonderful idea of smoothed analysis in complexity theory. For a bit of background, complexity theory generally focuses on the worst case time taken by algorithms. That means it mostly cares about which input will take the most time, not about the average time taken over all inputs. The latter is also studied, but it's nigh impossible to find the distribution of input actually used in practice (and some problems studied in complexity theory are never used in practice, so a meaningful distribution is even harder to

make sense of). Just like it's very hard to find the distribution from which goals are sampled in practice.

As a consequence of the focus on worst case complexity, some results in complexity theory clash with experience. Here we focus on the simplex algorithm, used for linear programming: it runs really fast and well in practice, despite having provable exponential worst case complexity. Which in complexity-theory-speak means it shouldn't be a practical algorithm.

Daniel Spielman and Shang-Hua Teng had a brilliant intuition to resolve this inconsistency: what if the worst case inputs were so rare that just a little perturbation would make them easy again? Imagine a landscape that is mostly flat, with some very high but very steep peaks. Then if you don't land exactly on the peaks (and they're so pointy that it's really hard to get there exactly), you end up on the flat surface.

This intuition yielded smoothed analysis: instead of just computing the worst case complexity, we compute the worst case complexity averaged over some noise on the input. Hence the peaks get averaged with the flatness around them and have a low smoothed time complexity.

Convergent subgoals, especially in Turner's formulation, behave in an analogous way: think of the peaks as goals without convergent subgoals; to avoid power-seeking we would ideally reach one of them, but their rarity and intolerance to small perturbations (permutations here) makes it really hard. So the knowledge created is about the shape of the landscape, and leveraging the intuition of smoothed analysis, that tells us something important about the hardness of avoiding convergent subgoals.

Note though that there is one aspect in which Turner's results and the smoothed analysis of the simplex algorithm are complete opposite: in the former the peaks are what we want (no convergent subgoals) while in the latter they're what we don't want (inputs that take exponential time to treat). This inversion doesn't change the sort of knowledge produced, but it's an easy source of confusion.

epistemic analysis isn't only meant for clarifying and distilling results: it can and should pave the way to some insights on how the arguments could fail. Here, the analogy to smoothed complexity and the landscape picture suggests that Bostrom and Turner's argument could be interrogated by:

- Arguing that the sampling method we use in practice to decide tasks and goals for AI targets specifically the peaks.
- Arguing that the sampling is done in a smaller goal-space for which the peaks are broader
    - For Turner's version, one way of doing that might be to not consider the full orbit, but only the close-ish variations of the goals (small permutations instead of all permutations). So questioning the form of noise over the choice of goals that is used in Turner's work.

Smoothed Analysis	**Convergent Subgoals (Turner et al)**
Possible Inputs	Possible Goals
Worst-case inputs make steep and rare peaks	Goals without convergent subgoals make steep and rare peaks

Goodhart's Law and Distributed Impossibility/Hardness Results

Another recurrent concept in alignment thinking is Goodhart's law. It wasn't invented by alignment researchers, but Scott Garrabrant and David Manheim proposed a taxonomy of its different forms. Fundamentally, Goodhart's law tells us that if we optimize a proxy of what we really want (some measure that closely correlates with the wanted quantity in the regime we can observe), strong optimization tends to make the two split apart, meaning we don't end up with what we really wanted. For example, imagine that everytime you go for a run you put on your running shoes, and you only put on these shoes for running. Putting on your shoes is thus a good proxy for running; but if you decide to optimize the former in order to optimize the latter, you will take most of your time putting on and taking off your shoes instead of running.

In alignment, Goodhart is used to argue for the hardness of specifying exactly what we want: small discrepancies can lead to massive differences in outcomes.

Yet there is a recurrent problem: Goodhart's law assumes the existence of some True Objective, of which we're taking a proxy. Even setting aside the difficulties of defining what we really want at a given time, what if what we really want is not fixed but constantly evolving? Thinking about what I want nowadays, for example, it's different from what I wanted 10 years ago, despite some similarities. How can Goodhart's law apply to my values and my desires if there is not a fixed target to reach?

Salvation comes from a basic insight when comparing problems: if problem A (running a marathon) is harder than problem B (running 5 km), then showing that the latter is really hard, or even impossible, transfers to the former.

My description above focuses on the notion of one problem being harder than another. TCS formalizes this notion by saying the easier problem is reducible to the harder one: a solution for the harder one lets us build a solution for the easier problem. And that's the trick: if we show there is no solution for the easier problem, this means that there is no solution for the harder one, or such a solution could be used to solve the easier problem. Same thing with hardness results which are about how difficult it is to solve a problem.

That is, when proving impossibility/hardness, you want to focus on the easiest version of the problem for which the impossibility/hardness still holds.

In the case of Goodhart's law, this can be used to argue that it applies to moving targets because having True Values or a True Objective makes the problem easier. Hitting a fixed target sounds simpler than hitting a moving or shifting one. If we accept that conclusion, then because Goodhart's law shows hardness in the former case, it also does in the latter.

That being said, whether the moving target problem is indeed harder is debatable and debated. My point here is not to claim that this is definitely true, and so that Goodhart's law necessarily applies. Instead, it's to focus the discussion on the relative hardness of the two problems, which is what underlies the correctness of the epistemic strategy I just described. So the point of this analysis is that there is another argument to decide the usefulness of Goodhart's law in alignment than debating the existence of True Value

	Running	**Alignment**
Easier Problem	5k	Approximating a **fixed** target (True Values)
Harder Problem	Marathon	Approximating a **moving** target

Orthogonality Thesis and Complexity Barriers

My last example is Bostrom's Orthogonality Thesis: it states that goals and competence are orthogonal, meaning that they are independent– a certain level of competence doesn't force a system to have only a small range of goals (with some subtleties that I address below).

That might sound only too general to really be useful for alignment, but we need to put it in context. The Orthogonality Thesis is a response to a common argument for alignment-by-default: because a Human-level AI/Transformative AI/AGI would be competent, it should realize what we really meant/wanted, and correct itself as a result. Bostrom points out that there is a difference between understanding and caring. The AI understanding our real intentions doesn't mean it must act on that knowledge, especially if it is programmed and instructed to follow our initial commands. So our advanced AI might understand that we don't want it to follow convergent subgoals while maximizing the number of paperclips produced in the world, but what it cares about is the initial goal/command/task of maximizing paperclips, not the more accurate representation of what we really meant.

Put another way, if one wants to prove alignment-by-default, the Orthogonality thesis argues that competence is not enough. As it is used, it's not so much a result about the real world, but a result about how we can reason about the world. It shows that one class of arguments (competence will lead to human values) isn't enough.

Just like some of the weirdest results in complexity theory: the barriers to P vs NP. This problem is one of the biggest and most difficult open questions in complexity theory: settling formally the question of whether the class of problems which can tractably be solved (P for Polynomial time) is equal to the class of problems for which solutions can be tractably checked (NP for Non-deterministic Polynomial time). Intuitively those are different: the former is about creativity, the second about taste, and we feel that creating something of quality is harder than being able to recognize it. Yet a proof of this result (or its surprising opposite) has evaded complexity theorists for decades.

That being said, recall that theoretical computer scientists are experts at wringing knowledge out of everything, including their inability to prove something. This resulted in the three barriers to P vs NP: three techniques from complexity theory which have been proved to not be enough by themselves for showing P vs NP or its opposite. I won't go into the technical details here, because the analogy is mostly with the goal of these barriers. They let complexity theorists know quickly if a proof has potential – it must circumvent the barriers somehow.

The Orthogonality thesis plays a similar role in alignment: it's an easy check for the sort of arguments about alignment-by-default that many people think of when learning about the topic. If they extract alignment purely from the competence of the AI, then the Orthogonality Thesis tells us something is wrong.

What does this mean for criticism of the argument? That what matters when trying to break the Orthogonality Thesis isn't its literal statement, but whether it still provides a barrier to alignment-by-default. Bostrom himself points out that the Orthogonality Thesis isn't literally true in some regimes (for example some goals might require a minimum of competence) but that doesn't affect the barrier nature of the result.

Barriers to P vs NP	**Orthogonality Thesis**
Proof techniques that are provably not enough to settle the question	Competence by itself isn't enough to show alignment-by-default

Improving the Epistemic State-of-the-art

Alignment research aims at solving a never-before-seen problem caused by a technology that doesn't exist yet. This means that the main epistemic strategies from Science and Engineering need to be adapted if used, and lose some of their guarantees and checks. In consequence, I claim we shouldn't only focus on those, but use all epistemic strategies we can find to produce new knowledge. This is already happening to some extent, causing both progress on many fronts but also difficulties in teaching, communicating and criticizing alignment work.

In this post I focused on epistemological analyses drawing from Theoretical Computer Science, both because of my love for it and because it fits into the background of many conceptual alignment researchers. But many different research directions leverage different epistemic strategies, and those should also be studied and unearthed to facilitate learning and criticism.

More generally, the inherent weirdness of alignment makes it nigh impossible to find one unique methodology of doing it. We need everything we can get, and that implies a more pluralistic epistemology. Which means the epistemology of different research approaches must be considered and studied and made explicit, if we don't want to be confusing for the rest of the world, and each other too.

I'm planning on focusing on such epistemic analyses in the future, both for the main ideas and concepts we want to teach to new researchers and for the state-of-the art work that needs to be questioned and criticized productively.

Epistemic Strategies of Selection Theorems

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

Introduction: Epistemic Strategies Redux

This post examines the epistemic strategies of John Wentworth's selection theorem posts.

(If you want to skim this post, just read the Summary subsection that display the different epistemic strategies as design patterns)

I introduced the concept in a recent post, but didn't define them except as the "ways of producing" knowledge that are used in a piece of research. If we consider a post or paper as a computer program outputting (producing) knowledge about alignment, epistemic strategies are the underlying algorithm or, even more abstractly, the design patterns.

An example of epistemic strategy, common in natural sciences (and beyond), is

- Look at the data
- Find a good explanation
- Predict new things with that explanation
- Get new data for checking your prediction

More than just laying out some abstract recipe, analysis serves to understand how each step is done, whether that makes sense, and how each step (and the whole strategy) might fail. Just like a design pattern or an algorithm, it matters tremendously to know when to apply it and when to avoid it as well as subtleties to be aware of.

Laying this underlying structure bare matters in three ways:

- It clarifies the research's purpose and value for newcomers and researchers from other fields, with minimal assumptions of shared approaches.
    - Just like a programmer switching to a new domain of problems will get up to speed faster and more reliably if they get access to the patterns/algorithms/tricks used in their new domain.
- It focuses feedback and criticism on the most important parts of the idea/proposition/argument.
    - Issues with an algorithm more often focus on the point of it instead of the details,whereas issues with an implementation of that algorithm can be as much about typos, optimization tricks and bad structure than about the actual core (the algorithm.
- It builds a library of such strategies for alignment in particular, a cookbook newcomers and senior researchers alike can browse for inspiration or a take on some new post/paper they don't grok.

    - Like the glorious Game Programming Patterns who does exactly that for game programming

Now, before starting, I need to point out that the selection theorems posts don't present research results; they present epistemic strategies (the eponymous theorems). Does that mean my job has already been done? Not exactly: John's posts do present that epistemic strategy, but not in all the ways I want to stress out. John is also trying to fill in a lot of concrete details and to convince people that selection theorems are a nice thing to research, which I don't have to do. Instead, you can see this post as distilling the structure of selection theorems and interrogating them further as ways of producing knowledge.

(I use the word "agent" to stay coherent with John, but nothing in the epistemic strategy itself requires agency, and so finding the idea of agents confusing shouldn't be an issue for reading this post)

Thanks to John Wentworth for feedback on a draft of this post.

Characterizing Selection Theorems

Selection theorems are theorems. Obviously. But what sort of theorems? What are they trying to find about the world?

John summarizes the whole class of results in the following way:

> Roughly speaking, **a Selection Theorem tells us something about what agent type signatures will be selected for (by e.g. natural selection or ML training or economic profitability) in some broad class of environments**.

This gives us three components of a selection theorem: the selection pressure, the class of environments considered and the constraint on the agent (what John calls the "type signatures"). Let's get into each, looking for what can fill the corresponding hole in the general selection theorem.

Selection

A selection theorem is first and foremost about selection. Not just **selection mechanisms** (low-level processes like natural selection) but also **selection criteria** (abstract conditions like no Dutch-booking). The former state how selection happens, whereas the other just characterize the sort of things that will be selected.

One of the differences is that a selection mechanism implies a selection criterion, either implicitly (natural selection) or explicitly (ML training with an actual loss function); whereas a selection criterion doesn't necessarily come with a mechanism.

Still, both mechanisms and criteria come in a wide variety -- what makes a good one for a selection theorem? Making the selection theorem applicable to the real world situation we care about. The next section focuses on this topic of application, but in summary: **mechanisms must fit actual selection processes in the situation, whereas criteria must come with an explanation of why they would be**

instantiated (possibly a corresponding selection mechanism, but not necessarily).

It's also less obvious what makes a "good" criterion, because of the risk to assume the constraint we want to show in the selection criterion itself.

Environments

I find John's formulation above unfortunate, because it doesn't stress enough how the "broad class of environment" **is part of the hypothesis of a selection theorem**, not the conclusion. The intuition here is that we need enough variety to instantiate the selection pressure or criterion. Selection let's you force the agent's hand, but only if you can instantiate the situations you need.

For a selection mechanism, this amounts to containing **the sort of situations where the mechanism will push in the right direction and be strong enough** (for example predation pushing natural selection forward). For a selection criterion, it is about including **the situations that take advantage of every suboptimality in the agent** (like the exploitative bets punishing suboptimality in no Dutch-Booking)

Note though that while a broader class of environments might be necessary for proving the theorems, it makes applying it more difficult by putting more conditions on the environments in the real world setting. **There is thus a trade-off between making it possible to prove the theorem (more environments) and making it possible to apply it (less environments)**. We thus want as small a set of environments as possible while still being large enough to leverage the selection.

Constraint on agents

In the original post, John takes pains to split agents' type signatures into different components and to explain how they interact with each other. At the level we're seeing stuff though, we only need to understand that type signatures are **necessary conditions** on the agents coming from the selection: if an agent is to be selected, it must do X (or do X with high probability).

What sort of conditions do selection theorems show? Here we have a discrepancy between what selection theorems historically prove and what John wants to get out of them. Existing selection theorems **only prove behavioral necessary conditions**: you must act like this (as in coherence theorems) or you must be able to do that (as in the Gooder Regulator theorem). On the other hand, **what we truly want are structural necessary conditions** — for example "you must have a separate world model with this interface and these properties". John's third post on selection theorems is all about how he wants that.

Indeed, structural constraints tell you not only that the system must solve the problem, but how it will do so. Alignment just becomes easier if we have knowledge of the internal structure of the system: we can make more pointed predictions about how it might be unaligned; we might use this structure for more concrete alignment schemes. Fundamentally, structural constraints give us back some of the guarantees of the main epistemic strategies of Science and Engineering that get lost in alignment: we don't have the technology yet, but we have some ideas of how it will work.

I'll go into more detail about proving structural constraints in the next section, but for the moment just note that this is the sort of thing we want.

Summary

Selection theorems thus have the following structure:

- **Hypotheses**
  - **(Selection pressure)** Some means of selection, either a mechanism or a criterion.
  - **(Environments)** A class of environments broad enough to instantiate the selection pressure in needed way, but small enough to still apply the theorem to real world settings.
- **Conclusion**
  - **(Necessary Condition on Agents)** Some property (ideally structural but maybe behavioral) that is guaranteed for all selected agents, or at least with high probability.

Proving Selection Theorems

Behavioral constraints

Existing selection theorems only prove behavioral constraints — that is, they only show that the agents must be behaviorally equivalent to a specific class (like EU maximizers in coherence theorems) or that they must be able solve a specific problem (remembering all relevant data in the Gooder Regulator theorem).

How to prove selection theorems for behavioral constraints? Looking at the existing theorems, the first thing to notice is that they tend to use selection criteria. It makes sense, as they tend to be proved backwards: looking at the necessary condition on agents, what criterion selects only agents behaving like this?

It doesn't mean such theorems are trivial or useless, **just that they tell us which criterion selects for the necessary condition, not what is selected by some selection pressure**.

Structural constraints

Here instead of criteria, mechanisms are favored. This is mostly because we want to show that some process (natural selection and/or ML training) leads to structural constraints, not find criteria for structural constraints.

Note that we should expect any mechanism to find some good ad-hoc agent without the structural properties; selection theorems for structural constraints can thus only give probabilistic guarantees. They say "out of the agents favored by this selection mechanism, most/almost all will have these structural properties".

Here are some epistemic strategies to argue that the typical agents selected by a selection theorem on behavior alone should in expectation have additional structural

constraints. The list isn't exhaustive, and I expect these strategies to be combined when actually arguing for such structural constraints.

- **(Agents with these structural constraints are easier to find)** Especially with a selection mechanism, we can argue for properties of the selected agents that are easier to find.
  For example, John argues that robust and broad optima are easier to find and retain through mechanisms for selection like gradient descent or natural selection, and proposes that these optima might correspond mostly to agents with modular structures.
- **(Agents with structural constraints are a majority)** If we can show that most of the selected-for agents have these structural constraints, that is some evidence that we should expect that structure. Not as strong as with an explanation of why these would be favored though.
  Note that this applies both to mechanisms and criteria.
- **(Agents with structural constraints are easier to sample)** I already described this epistemic strategy, in relation with Alex Turner's work on Convergent Subgoals and a comparison to Smoothed Analysis. Basically, if one can show that the agents without structural properties are so rare that they correspond to very steep high peaks in a mostly flat landscape, all but very few sampling of selected-for agents will end up satisfying the structural properties.
- **(Proposed sampling gets structural constraints with high-probability)** If we can propose a sampling method for agents, argue that it indeed fits with how the selection pressure eventually samples (like the proposal here for SGD), and show this sampling to find in expectation agents with the structural constraint, that's a very strong argument for assuming this structure.

Summary

Proving selection theorems use the following epistemic strategies:

- **Proving behavioral selection theorem**
  - Choose a necessary condition to investigate.
  - Find a selection criterion that should favor the necessary condition.
  - Prove the theorem.
- **Proving structural selection theorem**
  - Choose a selection mechanism to investigate.
  - Find a structural constraint that should be favored by the mechanism.
  - Prove the theorem.
    - Show that agents with these structural constraints are easier to find.
    - Show that agents with structural constraints are a majority.
    - Show that agents with structural constraints are easier to sample.
    - Propose a sampling of agents and show it results in structural constraints with high-probability.

Applying Selection Theorems

Even pure mathematicians don't prove theorems only for the joy of the proof: the value of a theorem often comes from what it shows and where it can be applied. The same holds in alignment, with the additional difficulty that we want to apply it to real world systems and situations, not only to other abstractions. This means we need to

understand when we can apply selection theorems and what we can learn from that application.

Requirements of selection theorems

First thing first: selection theorems require **the existence of selection**. Once again quite obvious, but it becomes more interesting if we dig into the subtleties.

How to argue for the existence of selection depends on whether the theorem uses a mechanism or a criterion.

- **(Mechanism)** The question is whether the mechanism actually happens in the real world application. Answering this question can go from trivial (we know ML training happens because we're the one implementing it) to yielding epistemic strategies used for showing selection happens (like the arguments for natural selection).
- **(Criterion)** An additional difficulty with a criterion is that we need to justify that selection along this criterion indeed happens. That doesn't necessarily mean providing a full selection mechanism, but we need at least reasons for why this would happen.
  Most selection theorems using criteria (like coherence theorems) propose a high-level selection mechanism for this purpose.

The other requirement lies on environments. **Not only do we need the variety of environments over which selection is taking place, but environments also need to fit the mold assumed in selection theorems.** Coherence theorems for example require that bets can be defined in the environments with the required properties, and that the space of bets considered contains the dutch-booking strategies for any suboptimal policy. The Gooder Regulator Theorem has more concrete requirements in terms of the underlying causal structure, and the same sort of variety constraint on the "tasks" that the agent has to solve.

Interpreting the application of selection theorems

Once we are confident the selection theorem applies in our concrete setting, we can reap its fruits. But what are those fruits? At first glance, they're obvious: the necessary conditions stated in the theorem! Yet anyone who ever applied a theorem to a real world setting knows how perilous that task is.

How do you make sense of the necessary conditions in your setting for example? You need to find a way of grounding the constraints on agents you get out of the theorem.

This is where most applications of theorems to real world settings go wrong, in my opinion. Yet this is also the part I have the least to say about, because I just don't have some nice epistemic strategy to check that some conclusions taken from applying a theorem to situation S actually make sense. I've seen people do that move, I've made it myself, but I don't have a nice description of the underlying algorithm. So **let's flag that as an open problem for the time being**.

Summary

Proving selection theorems use the following epistemic strategies:

- **Checking that the selection theorem applies**
  - Check that the selection exists.
    - For a mechanism, check that it fit with how selection happens.
    - For a criterion, find a (possibly high-level) mechanism for why selection happens along these lines.
  - Check that the environments fit the required structure.
  - Check that the environments fit the required variety.
- **Interpreting the theorem after applying it**
  - **Open Epistemic Strategy Problem**
    - How to interpret a behavioral constraint?
    - How to interpret a structural constraint?
      - That we can model the agent coherently with that structure?
      - That the agent implements that structure explicitly?

Breaking Selection Theorems

Last but not least, analyzing an epistemic strategy tells us where it can go wrong. The analogy to think here is of falsification: this is a standard and strong way of trying to break a scientific model. What does that look like for selection theorems?

Let's use the summary design patterns of the previous section, and for each one, finding issues/criticisms/ways of breaking that step.

- **Proving behavioral selection theorem**
  - Choose a necessary condition to investigate.
  - Find a selection criterion that should favor the necessary condition.
    - ***Find a counterexample (agent selected by criterion but not satisfying the necessary condition; or subset of enough agents to break probabilistic condition).***
  - Prove the theorem.
    - ***Find an error in the proof.***
- **Proving structural selection theorem**
  - Choose a selection mechanism to investigate.
  - Find a structural constraint that should be favored by the mechanism.
  - Prove the theorem.
    - Show that agents with these structural constraints are easier to find.
      - ***Show that many agents without the structural constraints can be easily found by the selection pressure.***
    - Show that agents with structural constraints are a majority.
      - ***Show that there isn't a majority of selected-for agents with structural constraints.***
    - Show that agents with structural constraints are easier to sample.
      - ***Argue that the set of selected-for agents is different that the one used in the work, and that for the actual set, sampling agents without structural constraints becomes simpler.***

        - Propose a sampling of agents and show it results in structural constraints with high-probability.
            - ***Show that the proposed sampling disagrees with what the selection pressure actually finds (showing that the probabilities are different, or that one can sample agents that the other can't).***
- **Checking that the selection theorem applies**
    - Check that the selection exists.
        - For a mechanism, check that it fits with how selection happens.
            - ***Show that the actual selection works differently than the mechanism described, and that these differences influence massively what is selected in the end.***
        - For a criterion, find a (possibly high-level) mechanism for why selection happens along these lines.
            - ***Argue that the posited high-level selection mechanism for the criterion doesn't exist or that it doesn't push towards the criterion.***
    - Check that the environments fit the required structure.
        - ***Show that the concrete environments don't fit the constraints of the theorem.***
    - Check that the environments fit the required variety
        - ***Show that the concrete environment lacks some situations that are needed to make the proof hold.***
- **Interpreting the theorem after applying it**
    - Open Epistemic Strategy Problem
        - How to interpret a behavioral constraint?
        - How to interpret a structural constraint?
            - That we can model the agent coherently with that structure?
            - That the agent implements that structure explicitly?.

Lastly, in addition to criticizing a specific application of the theorem, we might argue that the theorem cannot be applied to the wanted setting, or that it doesn't make sense to conclude what is wanted from it. This amounts to the points above, with the twist of arguing that it's impossible instead of just breaking the argument at some joint.

This obviously fails to list all possible ways of critiquing a selection theorem and its application. You might have noted that I didn't say anything about interpreting the necessary condition once the theorem is applied; indeed, without understanding the epistemic strategy involved, it's harder to get to the core.

Still, any criticism and feedback along these lines would be directly useful to the researcher (John or someone else) proposing a new selection theorem and/or applying one. My claim is that using the design pattern above helps in providing feedback, by drawing attention to the most important parts of the epistemic strategies involved.

Epistemic Strategies of Safety-Capabilities Tradeoffs

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

Introduction: Epistemic Strategies Redux

This post examines the epistemic strategies of Steve Byrnes' Safety-capabilities tradeoff dials are inevitable in AGI.

(If you want to skim this post, just read the Summary subsection that display the epistemic strategy as a design pattern)

I introduced the concept in a recent post, but didn't define them except as the "ways of producing" knowledge that are used in a piece of research. If we consider a post or paper as a computer program outputting (producing) knowledge about alignment, epistemic strategies are the underlying algorithm or, even more abstractly, the design patterns.

An example of epistemic strategy, common in natural sciences (and beyond), is

- Look at the data
- Find a good explanation
- Predict new things with that explanation
- Get new data for checking your prediction

More than just laying out some abstract recipe, analysis serves to understand how each step is done, whether that makes sense, and how each step (and the whole strategy) might fail. Just like a design pattern or an algorithm, it matters tremendously to know when to apply it and when to avoid it as well as subtleties to be aware of.

Laying this underlying structure bare matters in three ways:

- It clarifies the research's purpose and value for newcomers and researchers from other fields, with minimal assumptions of shared approaches.
    - Just like a programmer switching to a new domain of problems will get up to speed faster and more reliably if they get access to the patterns/algorithms/tricks used in their new domain.
- It focuses feedback and criticism on the most important parts of the idea/proposition/argument.
    - Issues with an algorithm more often focus on the point of it instead of the details,whereas issues with an implementation of that algorithm can be as much about typos, optimization tricks and bad structure than about the actual core (the algorithm.
- It builds a library of such strategies for alignment in particular, a cookbook newcomers and senior researchers alike can browse for inspiration or a take on some new post/paper they don't grok.

- Like the glorious Game Programming Patterns who does exactly that for game programming

Thanks to Steve Byrnes for feedback on a draft of this post.

Defining Safety-Capabilities Tradeoffs

What sort of knowledge is Steve attempting to create in his post? He set up explicitly to show that any alignment proposal must deal with one or more tradeoffs between safety and capabilities (which he calls **safety-capabilities tradeoff dials**).

> I will argue that the discussion should be framed as "Just how problematic is this dial? How do we minimize its negative impact?", not "This particular approach has a dial, so it's automatically doomed. Let's throw it out and talk about something else instead."

This is in opposition to claims that some alignment proposals should be deemed less promising or insufficient because they would include such tradeoffs.

> (Recent examples of the latter attitude, at least arguably: here, here.)

A good way of framing the difference between safety and capabilities is that safety is about worst-case reasoning (improving the bad things that might happen) whereas capabilities is about best-case or average-case reasoning (improving the plans the AI might come up with). Nothing forbids a solution with great worst-case, average-case and best-case guarantees; yet it's not incoherent to imagine a tradeoff between not failing too badly and succeeding as impressively as possible.

Then the problem is that if such tradeoffs exist, people will differ in their incentives and probabilities and preferences, in such a way that not everyone will agree on where to stand in the tradeoff. Given that safety is restrictive, we should expect people favoring capabilities over safety to get more impressive and sellable systems until existential risks kick in. Which is bad.

Showing the Inevitability of Safety-Capabilities Tradeoffs

Steve claims that any alignment proposal must include some safety-capabilities tradeoffs. What I'm interested in here is how he argues for his point, and whether his epistemic strategy makes sense.

Unfortunately, his section on exactly that is confusing. The section is called "Why do I say that these dials are inevitable?" (what we want, right?) and starts with this sentence:

> Here are a few examples.

A list of examples sounds like a particularly bad way of showing that something is **impossible** to avoid. Hand-picking of examples comes to mind as a big risk, and more generally non-representative examples .

Yet Steve actually makes a decent argument for the inevitability of safety-capabilities tradeoffs, just far too implicitly. His examples are not examples of alignment proposals and their corresponding tradeoffs, but of places where tradeoffs might appear in any alignment proposal.

- **(Testing before deployment)** More testing improves the safety guarantees and reduces our uncertainty, but costs time and money.
- **(Human feedback and/or supervision)** Humans being able to understand and correct the model helps with safety, but makes the model slower, less competitive, and constrained to only proposed plans it can justify to humans — all of which make it less competitive and capable
- **(Access to resources)** Constrained access to resources (internet, money, compute…) makes the model safer, but makes it less capable.
- **(Human norms and laws)** Following human norms, laws and customs helps with safety but adds additional constraints on the capabilities.

That at least some of these tradeoffs must emerge in every alignment proposal is the (very) implicit last step of his epistemic strategy. And it's unfortunately not so much argued for than stated. For example on testing:

> *Some* amount of sandbox testing would help capabilities, by helping the team better understand how things are going. But there's an optimal amount of sandbox testing for capabilities, and doing further testing *beyond* that point is a safety-capabilities tradeoff.

How can we actually argue for this instead of simply saying it? Here I go one step further than the original post (while staying coherent with Steve's points) by proposing that we adapt how impossibility results are proved in Theoretical Computer Science. Impossibility proofs tend to focus on the potential counterexamples, and get to the gist of why they don't actually work. This involves the sort of back and forth between trying to create a counterexample and showing why it doesn't work described by the great Nancy Lynch in her A Hundred Impossibility Proofs for Distributed Computing (Yes, there are a hundred results, although many come for free by the same methods)

> How does one go about working on an impossibility proof? The first thing to do is to try to avoid solving the problem, by using a reducibility to reduce some other unsolvable problem to it. If this fails, next consider your intuitions about the problem. This might not help much either: in my experience, my intuitions about which way the result will go have been wrong about 50% of the time.
>
> Then it is time to begin the game of playing the positive and negative directions of a proof against each other. My colleagues and I have often worked alternately on one direction and the other, in each case until we got stuck. It is not a good idea to work just on an impossibility result, because there is always the unfortunate possibility that the task you are trying to prove is impossible is in fact possible, and some algorithm may surface.
>
> An interesting interplay often arises when you work alternately on both directions. The limitations you find in designing an algorithm - e.g., the reason a particular algorithm fails - may be generalizable to give a limitation on all algorithms. […] Conversely, the reasons that mathematical impossibility proof fails can sometimes be exploited to devise counterexample algorithms.

Although we have no hope of proving Steve's claims in the near future (given our inability to formalize any of the relevant terms), this approach can be leveraged by looking for what would make a counterexample to each of Steve's examples.

This means we're looking for cases where there is no tradeoff between safety and capabilities: everyone agrees on what should be done. This amounts to saying that alignment people agree that there is nothing more to be done, which means one of two things:

- The methods proposed (testing, human understanding…) are deemed useless because they cannot catch the relevant problems (maybe the model is superhumanly deceptive, and no test/supervision/constraints will change anything). In other worlds, problems are hidden in a way that our techniques cannot handle, and so there is no point in asking for more safety checks.
    - **Yet this hides a more high-level tradeoff: alignment people would say that we shouldn't create and/or release the model at all in these conditions!**
- The methods proposed (testing, human understanding…) are deemed useless because even alignment people are **all completely certain** that they got the right scheme and that it will work.
    - **That sounds wildly improbable, and even if it was possible in principle, I don't know anyone who would argue that it is probable in the near future.**

Summary

The epistemic strategy at hands here is thus the following:

- **Arguing that a class of tradeoffs cannot be avoided in alignment proposals**
    - Give a list of tradeoffs from this class.
        - If possible from different parts/points in proposals.
    - Argue that some of these tradeoffs appear for every proposal.
        - Extract different types of potential counterexamples.
        - Argue why each category of counterexamples can't exist.**.**

Breaking the Inevitability of Safety-Capabilities Tradeoffs

Recall that epistemic strategies are design patterns, blueprints — following one helps, but doesn't ensure that the resulting argument will be correct. And epistemic strategies highlight where the meat of the reasoning is, thus where to focus attention and criticism.

So let's take the summary strategy and propose ways of breaking it.

Arguing that a class of tradeoffs cannot be avoided in alignment proposals

- Give a list of tradeoffs from this class.
    - If possible from different parts/points in proposals.

        - ***Argue that they are too clustered in proposal space, too focused on a specific kind of proposals.***
- Argue that some of these tradeoffs appear for every proposal.
    - Extract different types of potential counterexamples.
        - ***Argue that these are not all the possible types, for example by providing a counterexample that doesn't fit in any.***
    - Argue why each category of counterexamples can't exist.
        - ***Break one of these arguments, by showing a failure of reasoning.***
        - ***Break one of these arguments by providing an actual counterexample from the category.***

Interpreting Yudkowsky on Deep vs Shallow Knowledge

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

Here is an exploration of what Eliezer Yudkowsky means when he writes about deep vs shallow patterns (although I'll be using "knowledge" instead of "pattern" for reasons explained in the next section). Not about any specific pattern Yudkowsky is discussing, mind you, about what deep and shallow patterns are at all. In doing so, I don't make any criticism of his ideas and instead focus on quoting him (seriously, this post is like 70% quotes) and interpreting him by finding the best explanation I can of his words (that still fit them, obviously). Still, there's a risk that my interpretation misses some of his points and ideas— I'm building a lower-bound on his argument's power that is as high as I can get, not an upper-bound. Also, I might just be completely wrong, in which case defer to Yudkowsky if he points out that I'm completely missing the point.

Thanks to Eliezer Yudkowsky, Steve Byrnes, John Wentworth, Connor Leahy, Richard Ngo, Kyle, Laria, Alex Turner, Daniel Kokotajlo and Logan Smith for helpful comments on a draft.

Back to the FOOM: Yudkowsky's explanation

In recent discussions, Yudkowsky often talks about deep patterns and deep thinking. What he made clear in a comment on this draft is that he has been using the term "deep patterns" in two different ways:

- What I'll call **deep knowledge**, which is a form of human knowledge/theory as well as the related epistemic strategies. This is what I explore below.
- What I'll call **deep cognition**, which is the sort of deep patterns that Yudkowsky points out AGI would have. There's a link and an analogy with the deep knowledge, but I don't get it enough to write something convincing to me and Yudkowsky, **so I'll mostly avoid that topic in this post.**

Focusing on deep knowledge then, Yudkowsky recently seems to ascribe his interlocutors' failure to grasp his point to their inability to grasp different instances of deep knowledge.

(All quotes from Yudkowsky if not mentioned otherwise)

(From the first discussion with Richard Ngo)

> In particular, just as I have a model of the Other Person's Beliefs in which they think alignment is easy because they don't know about difficulties I see as very deep and fundamental and hard to avoid, I also have a model in which people think "why not just build an AI which does X but not Y?" because they don't realize what X and Y have in common, which is something that draws deeply on having

> deep models of intelligence. And it is hard to convey this deep theoretical grasp.

That being said, he doesn't really explain what this sort of deep knowledge is.

(From the same discussion with Ngo)

> (Though it's something of a restatement, a reason I'm not going into "my intuitions about how cognition works" is that past experience has led me to believe that conveying this info in a form that the Other Mind will actually absorb and operate, is really quite hard and takes a long discussion, relative to my current abilities to Actually Explain things; it is the sort of thing that might take doing homework exercises to grasp how one structure is appearing in many places, as opposed to just being flatly told that to no avail, and I have not figured out the homework exercises.)

The thing is, he did exactly that in the FOOM debate with Robin Hanson 13 years ago. (For those unaware of this debate, Yudkoswky is responding to Hanson's use of trends — like Moore's law — extrapolations to think about intelligence explosion).

(From The Weak Inside View (2008))

> Robin keeps asking me what I'm getting at by talking about some reasoning as "deep" while other reasoning is supposed to be "surface." One thing which makes me worry that something is "surface" is when it involves generalizing a level N feature across a shift in level N−1 causes.
>
> For example, suppose you say, "Moore's Law has held for the last sixty years, so it will hold for the next sixty years, even after the advent of superintelligence" (as Kurzweil seems to believe, since he draws his graphs well past the point where you're buying a billion times human brainpower for $1,000).
>
> Now, if the Law of Accelerating Change were an exogenous, ontologically fundamental, precise physical law, then you wouldn't expect it to change with the advent of superintelligence.
>
> But to the extent that you believe Moore's Law depends on human engineers, and that the timescale of Moore's Law has something to do with the timescale on which human engineers think, then extrapolating Moore's Law across the advent of superintelligence is extrapolating it across a shift in the previous causal generator of Moore's Law.
>
> So I'm worried when I see generalizations extrapolated across a change in causal generators not themselves described—i.e., the generalization itself is on the level of the outputs of those generators and doesn't describe the generators directly.
>
> If, on the other hand, you extrapolate Moore's Law out to 2015 because it's been reasonably steady up until 2008—well, Reality is still allowed to say, "So what?" to a greater extent than we can expect to wake up one morning and find Mercury in Mars's orbit. But I wouldn't bet against you, if you just went ahead and drew the graph.
>
> So what's "surface" or "deep" depends on what kind of context shifts you try to extrapolate past

An important subtlety here comes from the possible conflation of two uses of "surface": the implicit use of "surface knowledge" as the consequences of some underlying causal processes/generator, and the explicit use of "surface knowledge" as drawing similarities without thinking about the causal process generating them. To simplify the discussion, let's use the more modern idiom of "shallow" for the more explicit sense here.

So what is Yudkowsky pointing at? Two entangled things:

- **If you have shallow knowledge, that is a trend without an underlying causal model, then you can't extend it when the causal process generating it changes.** So if Moore's law depends on "the timescale on which human engineers think", we can't extend it past the intelligence explosion, because then human engineers would be reply by AI engineers which would think faster.
- **If you have shallow knowledge, you can't even know when to extend the trend safely because understanding when the underlying causal process changes is harder when you don't know what the causal process is!**

Imagine a restaurant that has a dish you really like. The last 20 times you went to eat there, the dish was amazing. So should you expect that the next time it will also be great? Well, that depends on whether anything in the kitchen changes. Because you don't understand what makes the dish great, you don't know of the most important aspects of the causal generators. So if they can't buy their meat/meat-alternative at the same place, maybe that will change the taste; if the cook is replaced, maybe that will change the taste; if you go at a different time of the day, maybe that will change the taste.

You're incapable of extending your trend (except by replicating all the conditions) to make a decent prediction because you don't understand where it comes from. If on the other hand you knew why the dish was so amazing (maybe it's the particular seasoning, or the chef's touch), then now you can estimate its quality. But then you're not using the trend, you're using a model of the underlying causal process.

Here is another phrasing by Yudkowsky from [the same essay]:

> Though this is to some extent an argument produced after the conclusion, I would explain my reluctance to venture into quantitative futurism via the following trichotomy:
>
> - On problems whose pieces are individually precisely predictable, you can use the Strong Inside View to calculate a final outcome that has never been seen before—plot the trajectory of the first moon rocket before it is ever launched, or verify a computer chip before it is ever manufactured.
> - On problems that are drawn from a barrel of causally similar problems, where human optimism runs rampant and unforeseen troubles are common, the Outside View beats the Inside View. Trying to visualize the course of history piece by piece will turn out to not (for humans) work so well, and you'll be better off assuming a probable distribution of results similar to previous historical occasions—without trying to adjust for all the reasons why this time will be different and better.
> - But on problems that are new things under the Sun, where there's a huge change of context and a structural change in underlying causal forces, the

> Outside View also fails—try to use it, and you'll just get into arguments about what is the proper domain of "similar historical cases" or what conclusions can be drawn therefrom. In this case, the best we can do is use the Weak Inside View—visualizing the causal process—to produce loose, qualitative conclusions about only those issues where there seems to be lopsided support.

More generally, these quotes point out to what Yudkowsky means when he says "deep knowledge": **the sort of reasoning that focuses on underlying causal models**.

As he says himself:

> To stick my neck out further: I am liable to trust the Weak Inside View over a "surface" extrapolation, if the Weak Inside View drills down to a deeper causal level and the balance of support is sufficiently lopsided.

Before going deeper into how such deep knowledge/Weak Inside View works and how to build confidence in it, I want to touch upon the correspondence between this kind of thinking and the Lucas Critique in macroeconomics. This link has been pointed out in the comments of the recent discussions — we thus shouldn't be surprised that Yudkowsky wrote about it 8 years ago (yet I was surprised by this).

(From Intelligence Explosion Microeconomics (2013))

> The "outside view" (Kahneman and Lovallo 1993) is a term from the heuristics and biases program in experimental psychology. A number of experiments show that if you ask subjects for estimates of, say, when they will complete their Christmas shopping, the right question to ask is, "When did you finish your Christmas shopping last year?" and not, "How long do you think it will take you to finish your Christmas shopping?" The latter estimates tend to be vastly over-optimistic, and the former rather more realistic. In fact, as subjects are asked to make their estimates using more detail—visualize where, when, and how they will do their Christmas shopping—their estimates become more optimistic, and less accurate. Similar results show that the actual planners and implementers of a project, who have full acquaintance with the internal details, are often much more optimistic and much less accurate in their estimates compared to experienced outsiders who have relevant experience of similar projects but don't know internal details. This is sometimes called the dichotomy of the inside view versus the outside view. The "inside view" is the estimate that takes into account all the details, and the "outside view" is the very rough estimate that would be made by comparing your project to other roughly similar projects without considering any special reasons why this project might be different.
>
> The Lucas critique (Lucas 1976) in economics was written up in 1976 when "stagflation"—simultaneously high inflation and unemployment—was becoming a problem in the United States. Robert Lucas's concrete point was that the Phillips curve trading off unemployment and inflation had been observed at a time when the Federal Reserve was trying to moderate inflation. When the Federal Reserve gave up on moderating inflation in order to drive down unemployment to an even lower level, employers and employees adjusted their long-term expectations to take into account continuing inflation, and the Phillips curve shifted. Lucas's larger and meta-level point was that the previously observed Phillips curve wasn't fundamental enough to be structurally invariant with respect to Federal Reserve policy—the concepts of inflation and unemployment weren't deep enough to

> describe elementary things that would remain stable even as Federal Reserve policy shifted.

and later in that same essay:

> The lesson of the outside view pushes us to use abstractions and curves that are clearly empirically measurable, and to beware inventing new abstractions that we can't see directly.
>
> The lesson of the Lucas critique pushes us to look for abstractions deep enough to describe growth curves that would be stable in the face of minds improving in speed, size, and software quality.
>
> You can see how this plays out in the tension between "Let's predict computer speeds using this very well-measured curve for Moore's Law over time—where the heck is all this other stuff coming from?" versus "But almost any reasonable causal model that describes the role of human thinking and engineering in producing better computer chips, ought to predict that Moore's Law would speed up once computer-based AIs were carrying out all the research!"

This last sentence in particular points out another important feature of deep knowledge: **that it might be easier to say negative things (like "this can't work") than precise positive ones (like "this is the precise law") because the negative thing can be something precluded by basically all coherent/reasonable causal explanations, while they still disagree on the precise details.**

Let's dig deeper into that by asking more generally what deep knowledge is useful for.

How does deep knowledge work?

We now have a pointer (however handwavy) to what Yudkowsky means by deep knowledge. Yet we have very little details at this point about what this sort of thinking looks like. To improve that situation, the next two subsections explore two questions about the nature of deep knowledge: what is it for, and where does it come from?

The gist of this section is that:

- **Deep knowledge is primarily useful for saying what isn't possible/what can't work, especially in cases (like alignment) where there is very little data to draw from.** (The comparison Yudkowsky keeps coming back to is how thermodynamics allows you to rule out perpetual motion machines)
- **Deep knowledge takes the form of compressed constraints on solution/hypothesis space, which have weight behind them because they let us rederive most of our current knowledge from basic/compressed ideas, and finding such compression without a strong entanglement with reality is incredibly hard.** (Here an example used by Yudkowsky is the sort of thought experiments, conservation laws, and general ideas about what physical laws look like that guided Einstein in his path to Special and General Relativity)

What is deep knowledge useful for?

The big difficulty that comes up again and again, in the FOOM debate with Hanson and the discussion with Ngo and Christiano, is that deep knowledge doesn't always lead to quantitative predictions. That doesn't mean that the deep knowledge isn't quantitative itself (expected utility maximization is an example used by Yudkowsky that is completely formal and quantitative), but that the causal model only partially constrains what can happen. That is, it doesn't constrain enough to make precise quantitative predictions.

Going back to his introduction of the Weak Outside view, recall that he wrote:

> But on problems that are new things under the Sun, where there's a huge change of context and a structural change in underlying causal forces, the Outside View also fails—try to use it, and you'll just get into arguments about what is the proper domain of "similar historical cases" or what conclusions can be drawn therefrom. In this case, the best we can do is use the Weak Inside View—visualizing the causal process—to produce **loose, qualitative conclusions about only those issues where there seems to be lopsided support**.

He follows up writing:

> So to me it seems "obvious" that my view of optimization is only strong enough to produce loose, qualitative conclusions, and that it can only be matched to its retrodiction of history, or wielded to produce future predictions, on the level of qualitative physics.
>
> "Things should speed up here," I could maybe say. But not "The doubling time of this exponential should be cut in half."
>
> I aspire to a deeper understanding of intelligence than this, mind you. But I'm not sure that even perfect Bayesian enlightenment would let me predict quantitatively how long it will take an AI to solve various problems in advance of it solving them. That might just rest on features of an unexplored solution space which I can't guess in advance, even though I understand the process that searches.

Let's summarize it that way: **deep knowledge only partially constrains the surface phenomena it describes (which translate into quantitative predictions) and it takes a lot of detailed deep knowledge (and often data) to refine it enough to pin down exactly the phenomenon and make precise quantitative predictions.** Alignment and AGI are fields where we don't have that much deep knowledge, and the data is sparse, and thus we shouldn't expect precise quantitative predictions anytime soon.

Of course, just because a prediction is qualitative doesn't mean it comes from deep knowledge; all hand-waving isn't wisdom. For a good criticism of shallow qualitative reasoning in alignment, let's turn to Qualitative Strategies of Friendliness.

> These then are three problems, with strategies of Friendliness built upon qualitative reasoning that seems to imply a positive link to utility:
>
> The fragility of *normal* causal links when a superintelligence searches for more efficient paths through time;
>
> The superexponential vastness of conceptspace, and the unnaturalness of the boundaries of our desires;

> And all that would be lost, if success is less than complete, and a superintelligence squeezes the future without protecting everything of value in it.

The shallow qualitative reasoning criticized here relies too much on human common sense and superiority to the AI, when the situation to predict is about superintelligence/AGI. That is, this type of qualitative reasoning extrapolates across a change in causal generators.

On the other hand, Yudkowsky uses qualitative constraints to guide his criticism: he knows there's a problem because the causal model forbids that kind of solution. Just like the laws of thermodynamics forbid perpetual motion machines.

Deep qualitative reasoning starts from the underlying (potentially quantitative) causal explanations and mostly tells you what cannot work or what cannot be done. That is, deep qualitative reasoning points out that a whole swatch of search space is not going to yield anything. A related point is that Yudkwosky rarely (AFAIK) makes predictions, even qualitative ones. He sometimes admits that he might do some, but it feels more like a compromise with the prediction-centered other person than what the deep knowledge is really for. Whereas he constantly points out how certain things cannot work.

(From Qualitative Strategies of Friendliness (2008))

> In general, a lot of naive-FAI plans I see proposed, have the property that, if actually implemented, the strategy might appear to work while the AI was dumber-than-human, but would fail when the AI was smarter than human. The fully general reason for this is that while the AI is dumber-than-human, it may not yet be powerful enough to create the exceptional conditions that will break the neat little flowchart that would work if every link operated according to the 21st-century First-World modal event.
>
> This is why, when you encounter the AGI wannabe who hasn't planned out a whole technical approach to FAI, and confront them with the problem for the first time, and they say, "Oh, we'll test it to make sure that doesn't happen, and if any problem like that turns up we'll correct it, now let me get back to the part of the problem that really interests me," know then that this one has not yet leveled up high enough to have interesting opinions. It is a general point about failures in bad FAI strategies, that quite a few of them don't show up while the AI is in the infrahuman regime, and only show up once the strategy has gotten into the transhuman regime where it is too late to do anything about it.

(From the second discussion with Ngo)

> I live in a world where I proceed with very strong confidence if I have a detailed formal theory that made detailed correct advance predictions, and otherwise go around saying, "well, it sure looks like X, but we can be on the lookout for a miracle too".
>
> If this was a matter of thermodynamics, I wouldn't even be talking like this, and we wouldn't even be having this debate.
>
> I'd just be saying, "Oh, that's a perpetual motion machine. You can't build one of those. Sorry." And that would be the end.

(From Security Mindset and Ordinary Paranoia (2017))

> You need to master two ways of thinking, and there are a lot of people going around who have the first way of thinking but not the second. One way I'd describe the deeper skill is seeing a system's security as resting on a story about why that system is safe. We want that safety-story to be as solid as possible. One of the implications is resting the story on as few assumptions as possible; as the saying goes, the only gear that never fails is one that has been designed out of the machine.
>
> [...]
>
> There's something to be said for redundancy, and having fallbacks in case the unassailable wall falls; it can be wise to have additional lines of defense, so long as the added complexity does not make the larger system harder to understand or increase its vulnerable surfaces. But at the core you need a simple, solid story about why the system is secure, and a good security thinker will be trying to eliminate whole assumptions from that story and strengthening its core pillars, not only scurrying around parrying expected attacks and putting out risk-fires.

Or my reading of the whole discussion with Christiano, which is that Christiano constantly tries to get Yudkowsky to make a prediction, but the latter focuses on aspects of Christiano's model and scenario that don't fit his (Yudkoswky's) deep knowledge.

I especially like the perpetual motion machines analogy, because it drives home how just proposing a tweak/solution without understanding Yudkowsky's deep knowledge (and what it would take for it to not apply) has almost no chance of convincing him. Because if someone said they built a perpetual motion machine without discussing how they bypass the laws of thermodynamics, every scientifically literate person would be doubtful. On the other hand, if they seemed to be grappling with thermodynamics and arguing for a plausible way of winning, you'd be significantly more interested.

(I feel like Bostrom's Orthogonality Thesis is a good example of such deep knowledge in alignment that most people get, and I already argued elsewhere that it serves mostly to show that you can't solve alignment by just throwing competence at it — also note that Yudkowsky had the same pattern earlier/parallely, and is still using it)

To summarize: **the deep qualitative thinking that Yudkowsky points out by saying "deep knowledge" is the sort of thinking that cuts off a big chunk of possibility space, that is tells you the whole chunk cannot work. It also lets you judge from the way people propose a solution (whether they tackle the deep pattern or not) whether you should ascribe decent probability to them being right.**

A last note in this section: although deep knowledge primarily leads to negative conclusions, it can also lead to positive knowledge through a particularly Bayesian mechanism: **if the deep knowledge destroys every known hypothesis/proposal except one (or a small number of them), then that is strong evidence for the ones left.**

(This quote is more obscure than the others without the context. It's from Intelligence Explosion Microeconomics (2013), and discusses the last step in a proposal for formalizing the sort of deep insight/pattern Yudkowksy leveraged during the FOOM debate. If you're very confused, I feel like the most relevant part to my point is the bold last sentence.)

> If Step Three is done wisely—with the priors reflecting an appropriate breadth of uncertainty—and doesn't entirely founder on the basic difficulties of formal statistical learning when data is scarce, then I would expect any such formalization to yield mostly qualitative yes-or-no answers about a rare handful of answerable questions, rather than yielding narrow credible intervals about exactly how the internal processes of the intelligence explosion will run. A handful of yeses and nos is about the level of advance prediction that I think a reasonably achievable grasp on the subject should allow—we shouldn't know most things about intelligence explosions this far in advance of observing one—we should just have a few rare cases of questions that have highly probable if crude answers. I think that one such answer is "AI go FOOM? Yes! AI go FOOM!" but I make no pretense of being able to state that it will proceed at a rate of 120,000 nanofooms per second.
>
> Even at that level, covering the model space, producing a reasonable simplicity weighting, correctly hooking up historical experiences to allow falsification and updating, and getting back the rational predictions would be a rather ambitious endeavor that would be easy to get wrong. Nonetheless, I think that Step Three describes in principle what the ideal Bayesian answer would be, given our current collection of observations. **In other words, the reason I endorse an AI-go-FOOM answer is that I think that our historical experiences falsify most regular growth curves over cognitive investments that wouldn't produce a FOOM.**

Where does deep knowledge come from?

Now that we have a decent grounding of what Yudkowsky thinks deep knowledge is for, the biggest question is how to find it, and how to know you have found good deep knowledge. After all, maybe the causal models one assumes are just bad?

This is the biggest difficulty that Hanson, Ngo, and Christiano seemed to have with Yudkowsky's position.

(Robin Hanson, from the comments after Observing Optimization in the FOOM Debate)

> If you can't usefully connect your abstractions to the historical record, I sure hope you have some data you can connect them to. Otherwise I can't imagine how you could have much confidence in them.

(Richard Ngo from his second discussion with Yudkowsky)

> Let me put it this way. There are certain traps that, historically, humans have been very liable to fall into. For example, seeing a theory, which seems to match so beautifully and elegantly the data which we've collected so far, it's very easy to dramatically overestimate how much that data favours that theory. Fortunately, science has a very powerful social technology for avoiding this (i.e. making falsifiable predictions) which seems like approximately the only reliable way to avoid it - and yet you don't seem concerned at all about the lack of application of this technology to expected utility theory.

(Paul Christiano from his discussion with Yudkowsky)

> OK, but you keep saying stuff about how people with my dumb views would be "caught flat-footed" by historical developments. Surely to be able to say something like that you need to be making some kind of prediction?

Note that these attitudes make sense. I especially like Ngo's framing. Falsifiable predictions (even just postdictions) are the cornerstone of evaluation hypotheses in Science. It even feels to Ngo (as it felt to me) that Yudkowsky argued for that in the Sequences:

(Ngo from his second discussion with Yudkowsky)

> I'm familiar with your writings on this, which is why I find myself surprised here. I could understand a perspective of "yes, it's unfortunate that there are no advanced predictions, it's a significant weakness, I wish more people were doing this so we could better understand this vitally important theory". But that seems very different from your perspective here.

(And Yudkoswky himself from Making Belief Pay Rent (In Anticipated Experience))

> Above all, don't ask what to believe—ask what to anticipate. Every question of belief should flow from a question of anticipation, and that question of anticipation should be the center of the inquiry. Every guess of belief should begin by flowing to a specific guess of anticipation, and should continue to pay rent in future anticipations. If a belief turns deadbeat, evict it.

But the thing is… rereading part of the Sequences, I feel Yudkowsky was making points about deep knowledge all along? Even the quote I just used, which I interpreted in my rereading a couple of weeks ago as being about making predictions, now sounds like it's about the sort of negative form of knowledge that forbids "perpetual motion machines". Notably, Yudkowsky is very adamant that beliefs must tell you what **cannot** happen. Yet that doesn't imply at all to make predictions of the form "this is how AGI will develop", so much as saying things like "this approach to alignment cannot work".

Also, should I point out that there's a whole sequence dedicated to the ways rationality can do better than science? (Thanks to Steve Byrnes for the pointer). I'm also sure I would find a lot of relevant stuff by rereading Inadequate Equilibria too, but if I wait to have reread everything by Yudkowsky before posting, I'll be there a long time…

My Initial Mistake and the Einstein Case

Let me jump here with my best guess of Yudkowsky's justification of deep knowledge: **their ability to both**

- **strongly compress "what sort of hypothesis ends up being right" without having to add anything ad-hoc-y to get our theory and hypotheses back;**
- **and constrain anticipations in non-trivial ways.**

The thing is, I got it completely wrong initially. Reading Einstein's Arrogance (2007), an early Sequences post that is all about saying that Einstein had excellent reasons to believe General Relativity's correctness before experimental verification (of advanced predictions), I thought that relativity was the deep knowledge and that Yudkowsky was

pointing out how Einstein, having found an instance of true deep knowledge, could allow himself to be more confident than the social process of Science would permit in the absence of experimental justification.

Einstein's Speed (2008) made it clear that I had been looking at the moon when I was supposed to see the pointing finger: **the deep knowledge Yudkowsky pointed out was not relativity itself, but what let Einstein single it out by a lot of armchair reasoning and better use of what was already known.**

> In our world, Einstein didn't even *use* the perihelion precession of Mercury, except for verification of his answer produced by other means. Einstein sat down in his armchair, and thought about how *he* would have designed the universe, to look the way he thought a universe should look—for example, that you shouldn't ought to be able to distinguish yourself accelerating in one direction, from the rest of the universe accelerating in the other direction.
>
> And Einstein executed the whole long (multi-year!) chain of armchair reasoning, without making any mistakes that would have required further experimental evidence to pull him back on track.

More generally, I interpret the whole Science and Rationality Sequence as explaining how deep knowledge can let rationalists do something that isn't in the purview of traditional Science: estimate which hypotheses make sense before the experimental predictions and evidence come in.

(From Faster Than Science (2008))

> This doesn't mean that the process of deciding which ideas to test is *unimportant* to Science. It means that Science doesn't *specify* it.
>
> [...]
>
> In practice, there are some scientific queries with a large enough answer space, that picking models at random to test, it would take zillions of years to hit on a model that made good predictions—like getting monkeys to type Shakespeare.
>
> At the *frontier* of science—the boundary between ignorance and knowledge, where science *advances*—the process relies on at least some individual scientists (or working groups) seeing things that are not yet confirmed by Science. That's how they know which hypotheses to test, in advance of the test itself.
>
> If you take your Bayesian goggles off, you can say, "Well, they don't have to know, they just have to guess." If you put your Bayesian goggles back on, you realize that "guessing" with 10% probability requires nearly as much epistemic work to have been successfully performed, behind the scenes, as "guessing" with 80% probability—at least for large answer spaces.
>
> The scientist may not *know* he has done this epistemic work successfully, in advance of the experiment; but he must, in fact, have done it successfully! Otherwise he will not even *think* of the correct hypothesis. In large answer spaces, anyway.

There's a subtlety that is easy to miss: Yudkowsky doesn't say that specifying an hypothesis in a large answer space makes it high evidence. After all, you can just generate any random guess. **What he's pointing at is that to ascribe a decent**

amount of probability to a specific hypothesis in a large space through updating on evidence, you need to cut a whole swath of the space to redirect the probability on your hypothesis. And that from a purely computational perspective, this implies more work on whittling down hypotheses than to make the favored hypothesis certain enough through experimental verification.

His claim then seems that Einstein, and other scientists who tended to "guess right" at what would be later experimentally confirmed, couldn't have been just lucky — they must have found ways of whittling down the vastness of hypothesis space, so they had any chance of proposing something that was potentially right.

Yudkowsky gives some pointers to what he thinks Einstein was doing right.

(From Einstein's Speed (2008))

> Rather than observe the planets, and infer what laws might cover their gravitation, Einstein was observing the other laws of physics, and inferring what new law might follow the same pattern. Einstein wasn't finding an equation that covered the motion of gravitational bodies. Einstein was finding a character-of-physical-law that covered previously observed equations, and that he could crank to predict the next equation that would be observed.
>
> Nobody knows where the laws of physics come from, but Einstein's success with General Relativity shows that their common character is strong enough to predict the correct form of one law from having observed other laws, without necessarily needing to observe the precise effects of the law.
>
> (In a general sense, of course, Einstein did know by observation that things fell down; but he did not get GR by backward inference from Mercury's exact perihelion advance.)

So in that interpretation, Einstein learned from previous physics and from thought experiments how to cut away the parts of the hypothesis space that didn't sound like they could make good physical laws, until he was left with a small enough subspace that he could find the right fit by hand (even if that took him 10 years)

> So, from a Bayesian perspective, what Einstein did is still induction, and still covered by the notion of a simple prior (Occam prior) that gets updated by new evidence. It's just the prior was over the *possible characters of physical law,* and observing other physical laws let Einstein update his model of *the character of physical law*, which he then used to predict a particular law of gravitation.
>
> If you didn't have the concept of a "character of physical law", what Einstein did would look like magic—plucking the correct model of gravitation out of the space of all possible equations, with vastly insufficient evidence. But Einstein, by looking at *other* laws, cut down the space of possibilities for the *next* law. He learned the alphabet in which physics was written, constraints to govern his answer. Not magic, but reasoning on a higher level, across a wider domain, than what a naive reasoner might conceive to be the "model space" of only this one law.

In summary, **deep knowledge doesn't come in the form of a particularly neat hypothesis or compression; it is the engine of compression itself. Deep knowledge compresses "what sort of hypothesis tends to be correct", such that it can be applied to the search of a correct hypothesis at the object level.** That also cements the idea that deep knowledge gives constraints, not

predictions: you don't expect to be able to have such a strong criterion for correct hypothesis that given a massive hypothesis space, you can pinpoint the correct one.

Here it is good to generalize my previous mistake; recall that I took General Relativity for the deep knowledge, when it was actually the sort of constraints on physical laws that Einstein used for even finding General Relativity. Why? I can almost hear Yudkowsky answering in my head: because General Relativity is the part accepted and acknowledged by Science. I don't think it's the only reason, but there's an element of truth: **I privileged the "proper" theory with experimental validation over the more vague principles and concepts that lead to it.**

A similar mistake is to believe the deep knowledge is the theory when it actually is what the theory and the experiments unearthed. This is how I understand Yudkowsky's use of thermodynamics and evolutionary biology: he points out at the deep knowledge that led and was revealed by the work on these theories, more than at the theories themselves.

Compression and Fountains of Knowledge

We still don't have a good way of finding and checking deep knowledge, though. Not any constraint on hypothesis space is deep knowledge, or even knowledge at all. The obvious idea is to have a reason for that constraint. And the reason Yudkowsky goes for almost every time is compression. Not a compressed description, like Moore's law; nor a "compression" that is as complex as the pattern of hypothesis it's trying to capture. **Compression in the sense that you get a simpler constraint that can get you most of the way to regenerate the knowledge you're starting from.**

This view of the importance of compression is everywhere in the Sequences. A great example is Truly Part of You, which asks what knowledge you could rederive if it was deleted from your mind. If you have a deep understanding of the subject, and you keep recursively asking how a piece of knowledge could be rederived and then how "what's needed for the derivation" can be rederived, Yudkwosky argues that you will reach "fountains of knowledge". Or in the terminology of this post, deep knowledge.

> Almost as soon as I started reading about AI—even before I read McDermott—I realized it would be *a really good idea* to always ask myself: "How would I regenerate this knowledge if it were deleted from my mind?"
>
> The deeper the deletion, the stricter the test. If all proofs of the Pythagorean Theorem were deleted from my mind, could I re-prove it? I think so. If all knowledge of the Pythagorean Theorem were deleted from my mind, would I notice the Pythagorean Theorem to re-prove? That's harder to boast, without putting it to the test; but if you handed me a right triangle with sides of length 3 and 4, and told me that the length of the hypotenuse was calculable, I think I would be able to calculate it, if I still knew all the rest of my math.
>
> What about the notion of *mathematical proof*? If no one had ever told it to me, would I be able to reinvent *that* on the basis of other beliefs I possess? There was a time when humanity did not have such a concept. Someone must have invented it. What was it that they noticed? Would I notice if I saw something equally novel and equally important? Would I be able to think that far outside the box?
>
> How much of your knowledge could you regenerate? From how deep a deletion? It's not just a test to cast out insufficiently connected beliefs. It's a way of

absorbing *a fountain of knowledge, not just one fact*.

What do these fountains look like? They're not the fundamental theories themselves, but instead their underlying principles. Stuff like the principle of least action, Noether's theorem and the principles underlying Statistical Mechanics (don't know enough about it to name them). **They are the crystallized insights which constrain enough the search space that we can rederive what we knew from them.**

(Feynman might have agreed, given that he chose the atomic hypothesis/principle, "all things are made of atoms — little particles that move around in perpetual motion, attracting each other when they are a little distance apart, but repelling upon being squeezed into one another" was the one sentence he salvage for further generations in case of a cataclysm.)

Here I hear a voice in my mind saying "What does simple mean? Shouldn't it be better defined?" Yet this doesn't feel like a strong objection. Simple is tricky to define intensively, but scientists and mathematicians tend to be pretty good at spotting it, as long as they don't fall for Mysterious Answers. And most of the checks on deep knowledge seem to be in their ability to rederive the known correct hypotheses without adding stuff during the derivation.

A final point before closing this section: Yudkowsky writes that the same sort of evidence can be gathered for more complex arguments if they can be summarized by simple arguments that still get most of the current data right. **My understanding here is that he's pointing at the wiggle room of deep knowledge, that is at the non-relevant ways in which it can be off sometimes.** This is important because asking for that wiggle room can sound like ad-hoc adaptation of the pattern, breaking the compression assumption.

(From Intelligence Explosion Microeconomics (2013))

> In my case, I think how much I trusted a Step Three model would depend a lot on how well its arguments simplified, while still yielding the same net predictions and managing not to be falsified by history. I trust complicated arguments much more when they have simple versions that give mostly the same answers; I would trust my arguments about growth curves less if there weren't also the simpler version, "Smart minds build even smarter minds." If the model told me something I hadn't expected, but I could translate the same argument back into simpler language and the model produced similar results even when given a few cross-validational shoves, I'd probably believe it.

Conclusion

Based on my reading of his position, **Yudkowsky sees deep knowledge as highly compressed causal explanations of "what sort of hypothesis ends up being right". The compression means that we can rederive the successful hypotheses and theories from the causal explanation. Finally, such deep knowledge translates into partial constraints on hypothesis space, which focus the search by pointing out what cannot work. This in turn means that deep knowledge is far better at saying what won't work than at precisely predicting the correct hypothesis.**

I also want to point out something that became clearer and clearer in reading old posts: Yudkowsky is nothing if not coherent. You might not like his tone in the recent discussions, but if someone has been saying the same thing for 13 years, nobody seems to get it, and their model predicts that this will lead to the end of the world, maybe they can get some slack for talking smack.

Redwood's Technique-Focused Epistemic Strategy

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

Imagine you're part of a team of ML engineers and research scientists, and you want to help with alignment. Everyone is ready to jump in the fray; there's only one problem — how are you supposed to do applied research when you don't really know how AGI will be built, what it will look like, not even the architecture or something like that? What you have is the current state of ML, and a lot of conceptual and theoretical arguments.

You're in dire need of a bridge (an epistemic strategy) between the experiments you could run now and the knowledge that will serve for solving alignment.

Redwood Research is in this precise situation. And they have a bridge in mind. Hence this post, where I write down my interpretation of their approach, based on conversations with Redwood's Buck Shlegeris. As such, even if I'm describing Redwood's strategy, it's probably biased towards what sounds most relevant to both Buck and me.

Thanks to Buck Shlegeris for great discussions and feedback. Thanks to Nate Thomas for listening to me when we talked about epistemic strategies and for pushing me to send him and Buck my draft, starting this whole collaboration. Thanks to Seraphina Nix for feedback on a draft of this post.

Techniques, not Tasks or Issues

My first intuition, when thinking about such a bridge between modern experiments and useful practical knowledge for alignment, is to focus on **tasks** and/or **issues**. By task, I mean the sort of things we want an aligned AGI to do ("learn what killing means and not do it" might be an example), whereas issues are… issues with an AGI (deception for example). It sounded obvious to me that you start from one of those, and then try to make a simpler, analogous version you can solve with modern technology— the trick being how to justify your analogy.

This is not at all how Buck sees it. After my initial confusion wore off, I realized he thinks in terms of **techniques**: potential ways of aligning an AGI. If tasks are "what we want" and issues are "what might go wrong", **techniques focus on "how" — how we solve the task and avoid the issues.**

Buck's favorite example of a technique (and the one driving Redwood's current work) is a form of adversarial training where the main model receives adversarial examples from a trained adversary, and has its response judged by an overseer to see if they're acceptable or not.

Another example is Debate: the alignment proposal where two models debate on the questions proposed by the human judge/supervisor, and hopefully the honest strategy is favored such that it should always win the debate.

Have you seen the trend with these two examples? They both can be specified as ML problems. What I mean is that the researchers who invented these techniques provided broad strokes ML setups and parameterized loss functions for them. Of course, we can't train all the relevant parts yet (adversary and overseer in adversarial training, advanced debaters in debate). If we could, then studying the technique would just boil down to straightforward ML work.

Yet these two ingredients, ML specifications and hardness of specific parts, point to Redwood's bridge between their experiments and the AGI-level technique.

Aside: Choosing a Technique

Before going into the detail of the bridge, I want to clarify that I'm not going to discuss how to choose a technique. That depends on many subtle arguments and discussions, especially conceptual ones. This is an expertise that Redwood partly has already and that they want to improve on. But at the moment, the work they've done focuses on turning a technique they've chosen into relevant experiments they can run today. Hence the focus of this post.

Replacing AGI-level Capabilities

Recall that the main reason we can't experiment directly with a technique like adversarial training or debate is that some parts require more capabilities than we currently have — we don't know how to train the relevant parts of the model and get something that behaves as expected. How to deal with that problem?

By **replacing the unfeasible parts with "current technology" that accomplishes the same or similar job.**

Which "current technology"?

- **(Humans)**
  - In Redwood's current project (which works on a simpler version of adversarial training), both adversary and overseer are replaced by humans: the adversary by the team members looking for injurious continuations, and the overseer by the labellers of training data.
  - In Debate, the most recent write-up presents experiments where all models involved (judge and debaters) are replaced by humans.
- **(Hardcoded search to approximate the real model)**
  - In Redwood's current project, the conservative policy is first implemented by rejection sampling on the continuations generated by the fine-tuned model, based on the classifier output. So instead of directly training a model that "integrates" the classifier, they babble and prune using two distinct simpler models.
  - An example Buck gave me was a case where directly training the optimal policy was too hard, but training the value function and explicitly searching for decent actions according to the value function was possible (the difference comes from cases where the action space is so vast that searching for the optimal action with only the value function is intractable, which means that search would do worse than the optimal policy)
- **(Current ML models)**

    - In the Debate paper, they conduct an experiment where the debate is about classifying an image, the debaters (ML models) see the full picture but the judge (another ML model) only sees the pixels shown by the debaters during the debate.
- **(Other hard coded-tools)**
    - Buck gives the example of a technique where one component gives you a PDF of explanations about a given neuron, and replacing that by saliency maps and other current interpretability techniques. Long term, you'd hope that we can defer the interpretability-tool-writing work to AIs, but for the moment alignment researchers have to do it themselves.

One reason I was confused initially when talking to Buck is that this simplification process often requires a simpler task too, because if we want to use our currently available models, we have to pick easier tasks that they're definitely able to do. You can see this in Redwood's current project, where the task "don't produce injurious continuation" is just at the level where it's not trivial for current Language Models but they can still do it. Or in the first debate experiment with only ML models, where they simplified the debate to being about image classification such that the debater models could handle the task. But remember that even if there is a task simplification aspect, the main analogy is with the technique.

A Different Kind of Knowledge

Now we have a simpler setting where we can actually run ML experiments and try to make the technique work. This is a neat trick, but why should it give us the sort of knowledge we care about? It's not obvious that the result of such experiments tell us how to align an AI.

After all, they're not solving the real problem with the real technology.

And yet this strategy can produce important and valuable knowledge: **telling us which are the difficulties in applying the technique, and helping us refine the easier parts.**

Let's look at both in turn.

Finding the Difficulty

Once you simplify the problem enough that lack of capabilities isn't stopping you, **difficulties and issues tell you that even if you solve the parts you assumed away, there might be a core problem left.**

Again the last debate experiment is a good example: the researchers only use humans, and still they found a particularly nasty strategy for the dishonest debaters that the honest debater had trouble dealing with (see this section for a description). This tells them that such strategies (and the generalization of the mechanism underlying them) are a failure mode (or at least a tricky part) of the current debate protocol. They then attempted to change the protocol to disincentive this technique.

Now it's easy to look at that, and feel that they're not really showing anything or not addressing the problem because they're dealing with only one failure mode out of many, and not even one that appears only at superintelligent levels. Indeed, even

solving the problem found would not show that debate would be sufficient for alignment.

But that's not the point! The point is to build more familiarity with the technique and to get a grip on the hardest parts. Sure, you don't prove the safety that way, but you catch many problems, and maybe you can then show that the entire class of such problems can't happen. This is not a proof of alignment, but it is a step for biting off a whole chunk of failure modes. And just like in any natural science, what the experiments find can give insights and ideas to the theoretical researcher which help them formulate stronger techniques.

Honing the Simple Parts

What if you actually solve the simplified task, though? Assuming that you did a non-trivial amount of work, you found out about a part that is not instantaneous but can be done with modern technology.

Here Redwood's own project provides a good example: they successfully trained the classifier, the babble-and-prune conservative policy, and the distilled version.

What does it buy them? Well, they know they can do that part. They also have built some expertise into how to do it, what are the tricky parts, and how far they expect their current methods to generalize. More generally, they built skills for implementing that part of the technique. And they can probably find faster implementations, or keep up to speed with ML developments by adapting this part of the solution.

This is even less legible than the knowledge of the previous section, but still incredibly important: they honed part of the skills you need to implement that technique, as well as future variations on that technique. The alignment problem won't be solved if/when the conceptual researchers find a great working proposal, but if/when it is implemented first. Building up these skills is fundamental to having a pool of engineers and research scientists who can actually make the alignment proposal a reality, competitively enough to win the race if needs be.

Summary

The epistemic strategy at hand here is thus the following:

- **Find a technique that looks promising for alignment, and can be expressed as an ML problem**
  - (Not included here)
- **Replace parts of the ML problem that can't be solved with current technology**
  Possible options:
  - Use humans
  - Use hard-coded search
  - Simplify the task and use current ML models
  - Use hard-coded program
- **Solve the simplified ML problem**
  - (Normal ML)
- **Extract the relevant knowledge**
  - If problem is unsolved, unearthed a difficult part

    - If problem is solved, unearthed a part to hone.

Breaking the Epistemic Strategy

I normally finish these posts by a section on breaking the presented epistemic strategy. Because knowing how the process of finding new knowledge could break tells us a lot about when the strategy should be applied.

Yet here… it's hard to find a place where this strategy breaks? Maybe the choice of technique is bad, but I'm not covering this part here. If the simplification is either too hard or too simple, it still teaches us relevant knowledge, and the next iteration can correct for it.

Maybe the big difference with my previous example is that this strategy doesn't build arguments. Instead it's a process for learning more about the conceptual techniques concretely, and preparing ourselves to be able to implement them and related approaches as fast and efficiently as possible when needed. From that perspective, the process might not yield that much information in one iteration, but it generally gives enough insight to adapt the problem or suggest a different experiment.

The epistemic strategy presented here doesn't act as a guarantee for an argument; instead it points towards a way of improving the skill of concretely aligning AIs, and building mastery in it.