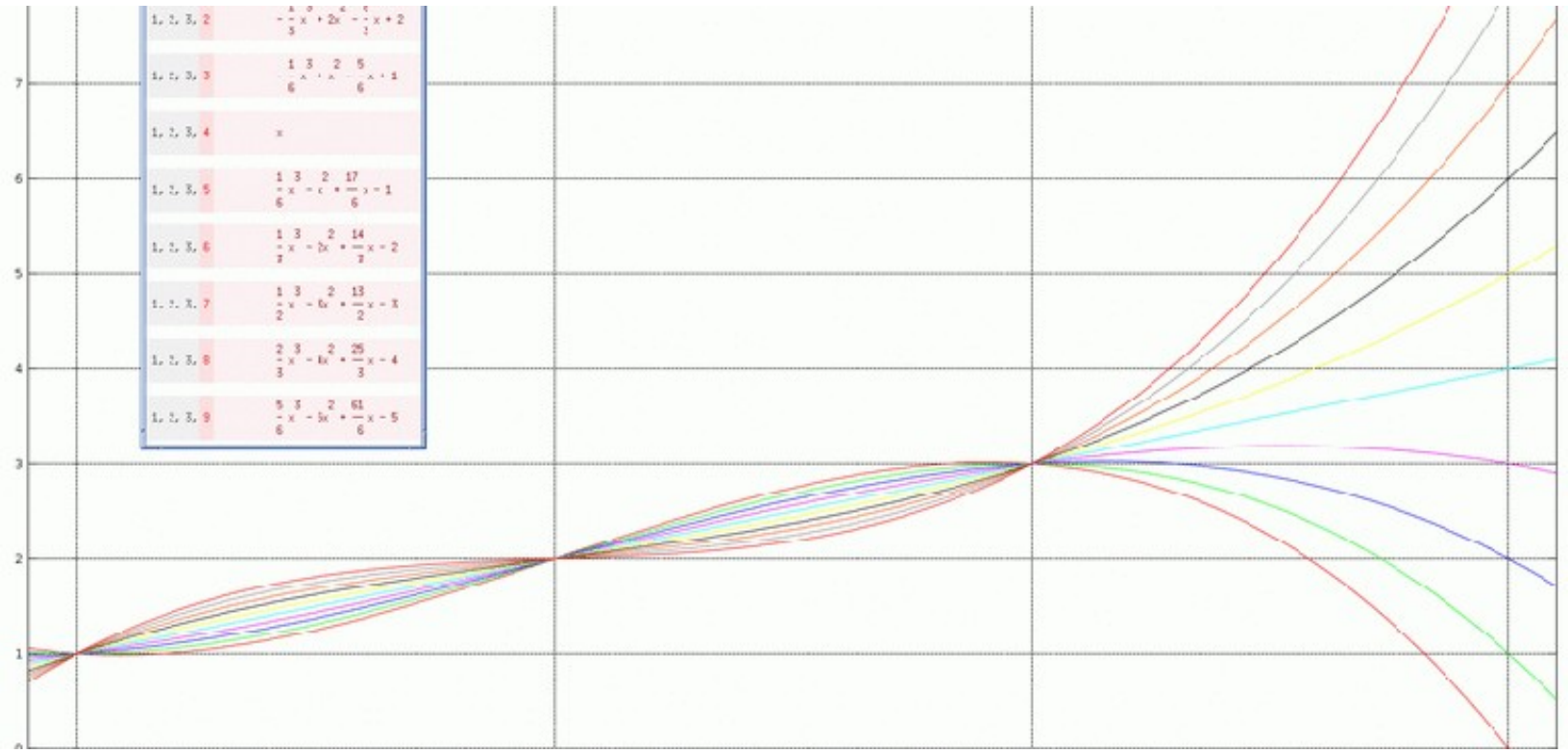

1, 1, 3, 2
1, 1, 3, 3
1, 1, 3, 4
x
1, 1, 3, 5
1, 1, 3, 6
1, 1, 3, 7
1, 1, 3, 8
1, 1, 3, 9
0
1
2
3
4
5
6
7

Concept Extrapolation

1. Concept extrapolation: key posts
2. Different perspectives on concept extrapolation
3. Model splintering: moving from one imperfect model to another
4. General alignment plus human values, or alignment via human values?
5. Value extrapolation, concept extrapolation, model splintering

Concept extrapolation: key posts

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

Concept extrapolation is the skill of taking a concept, a feature, or a goal that is defined in a narrow training situation... and extrapolating it safely to a more general situation. This more general situation might be very extreme, and the original concept might not make much sense (eg defining "human beings" in terms of quantum fields).

Nevertheless, since training data is always insufficient, key concepts must be extrapolated. And doing so successfully is a skill that humans have to a certain degree, and that an aligned AI would need to possess to a higher extent.

This sequence collects the key posts on concept extrapolation. They are not necessarily to be read in this order; different people will find different posts useful.

- Different perspectives on concept extrapolation collects many different analogies and models of concept extrapolation, intended for different audiences, and collected together here.
- Model splintering: moving from one imperfect model to another is the original post on "model splintering" - what happens when features no longer make sense because the world-model has changed. A long post with a lot of overview and motivation explanations, showing that model splintering is a problem with almost all alignment methods.
- General alignment plus human values, or alignment via human values? shows that concept extrapolation is necessary and almost sufficient for successfully aligning AIs.
- Value extrapolation, concept extrapolation, model splintering defines and disambiguates key terms: model splintering, value extrapolation, and concept extrapolation.

Different perspectives on concept extrapolation

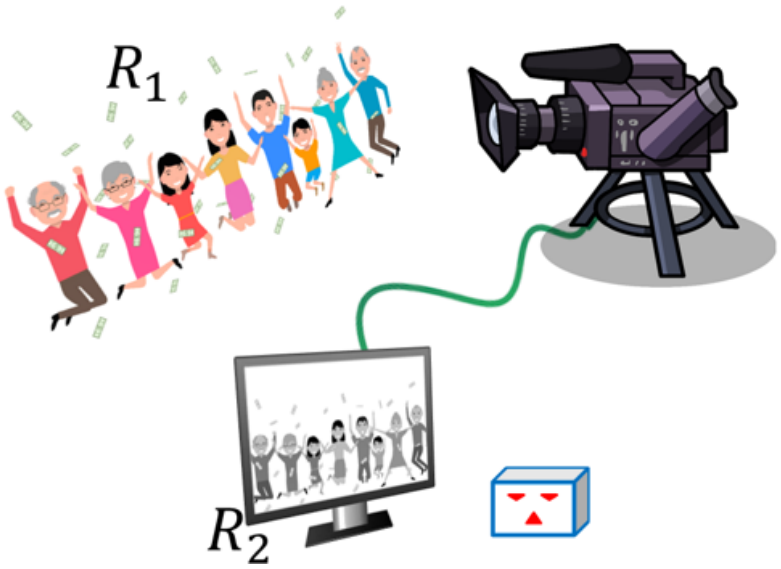
Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

At the recent EAGx Oxford meetup, I ended up talking with a lot of people (18 people, back to back, on Sunday - for some reason, that day is a bit of a blur). Naturally, many of the conversations turned to value extrapolation/concept extrapolation, the main current focus of our Aligned AI startup. I explained the idea I explained multiple times and in multiple different ways. Different presentations were useful for people from different backgrounds.

So I've collected the different presentations in this post. Hopefully this will allow people to find the explanation that provides the greatest clarity for them. I think many will also find it interesting to read some of the other presentations: from our perspective, these are just different facets of the same phenomenon[1].

For those worried about AI existential risk

An superintelligence trained on videos of happy humans may well tile the universe with videos of happy humans - that is a standard alignment failure mode. But "make humans happy" is also a reward function compatible with the data.

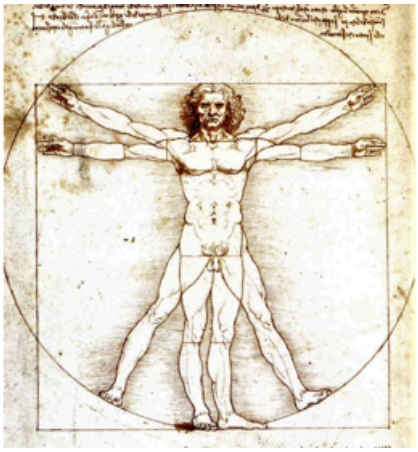


So let D_0 be the training data of videos of happy humans, R_1 the correct "make humans happy" reward function, and R_2 the degenerate reward function "make videos of happy humans"[2].

We'd want the AI to deduce R_1 from D_0. But even just generating R_1 as a candidate is a good success. The AI could then get feedback as to whether R_1 or R_2 is correct, or maximise a conservative mix of R_1 and R_2 (e.g. $R = \log(R_1) + \log(R_2)$). Maximising that conservative mix will result in a lot of videos of happy humans - but also a lot of happy humans.

For philosophers

Can you define what a human being is? Could you make a definition that works, in all circumstances and in all universe, no matter how bizarre or alien the world becomes?



A full definition has eluded philosophers ever since humans were categorised as "featherless bipeds with broad flat nails".

Concept extrapolation has another way of generating this definition. We would point at all living humans in the world and say "these are humans[3]."

Then we would instruct the AI: "please extrapolate the concept of 'human' from this data". As long as the AI is capable of doing that extrapolation better than we could ourselves, this would give us an extrapolation of the concept "human" to new circumstances without needing to write out a full definition.

For ML engineers into image classification

Paper Diversify and Disambiguate discusses a cow-grass-camel-sand example which is quite similar to the husky-wolf example of this post.

Suppose that we have two labelled sets, S_0 consisting of cows on grass, and S_1 consisting of camels on sand.

We'd like to train two classifiers that distinguish S_0 from S_1, but use different features to do so. Ideally, the first classifier would end up distinguishing cows from camels, while the second distinguishes grass from sand. Of course, we'd want them to do so independently, without needing humans labelling cows, grass, camels, or sand.

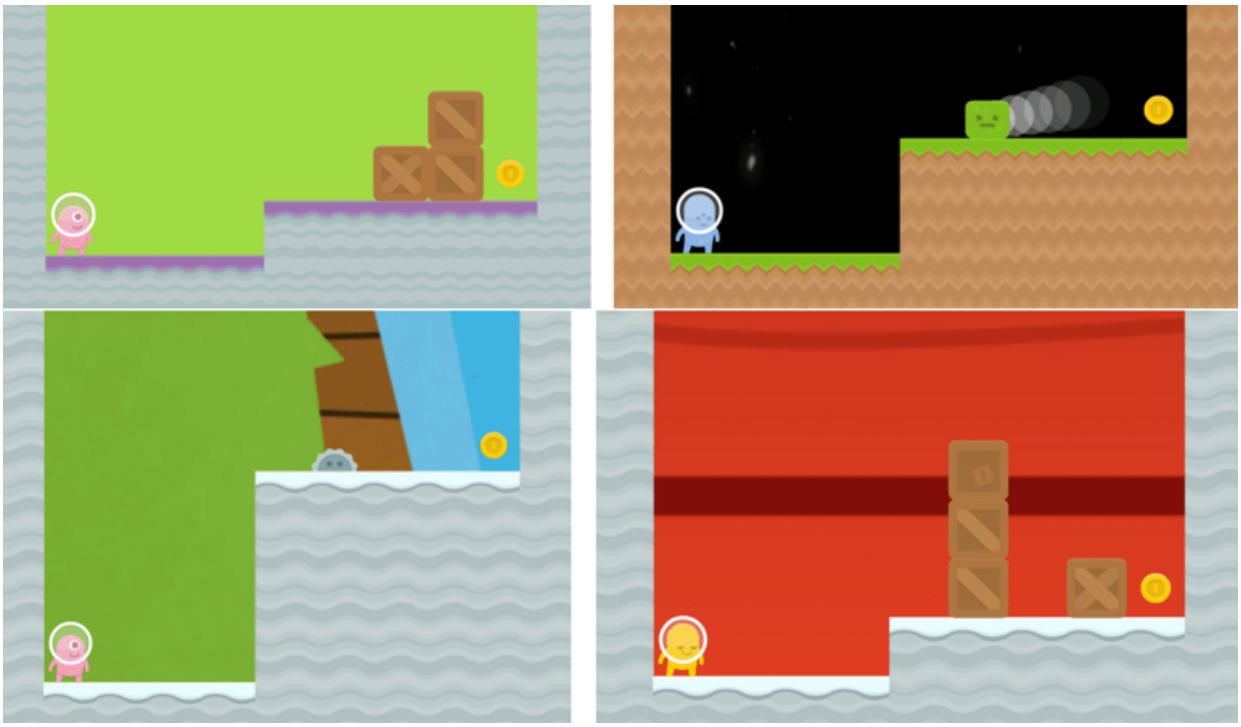
For ML engineers focusing on current practical problems

An AI classifier was trained on xray images to detect pneumothorax (collapse lungs). It was quite successful - until further analysis revealed that it was acting as a chest drain detector. The chest drain is a *treatment* for pneumothorax, making that classification useless.

We would want the classifier to generate "collapsed lung detector" and "chest drain detector" as separate classification, and then ask its programmers which one it should be classifying on.

For RL engineers

CoinRun is a procedurally generated set of environments, a simplified Mario-style platform game. The reward is given by reaching the coin on the right:



Since the coin is always at the right of the level, there are two equally valid simple explanations of the reward: the agent must reach the coin, or the agent must reach the right side of the level.

When agents trained on CoinRun are tested on environments that move the coin to another location, they tend to ignore the coin and go straight to the right side of the level. Note that the agent is following a policy, rather than generating a reward; still, the policy it follows is one that implicitly follows the "reach the right" reward rather than the "reach the coin" one.

We need an alternative architecture that generates both of these rewards[4] and is then capable of either choosing between them or becoming conservative between them (so that it would, eg, go to the right while picking up the coin along the way). This needs to be done in a generalisable way.

For investors

A major retail chain wants to train their CCTV cameras to automatically detect shoplifters. They train it on examples they have in their databases.

The problem is that those examples are correlated with other variables. They may end up training a racial classifier, or they may end up training an algorithm that identifies certain styles of clothes.

That is disastrous, firstly for the potential PR problems, but secondly because the classifier *won't successfully identify shoplifters*.

The ideal is if the AI implicitly generates "shoplifters", "racial groups", and "clothes style" as separate classifiers. And then enquires, using active learning, as to what its

purpose actually is. This allows the AI to classify properly for the purposes that it was designed for - and only those purposes.

For those working in AI alignment

Sometimes someone develops a way to keep AIs safe, by adding some constraints. For example, attainable utility preservation developed a formula to try and encode the concept of "power" for an AI, with a penalty term for having too much power:

$$\text{PENALTY}(s,a) = \sum_{R' \in R} |Q_{R'}(s,a) - Q_{R'}(s,\varnothing)|$$

With some difficulty, I constructed a situation where that formula failed to constrain the AI, via a subagent.

Essentially, the formal definition and the intuitive concept of power overlap in typical environments. But in extreme situations, they come apart. What is needed is an AI that can extrapolate the concept of power rather than the formal definition.

Doing this for other concepts allow a lot alignment methods to succeed such are avoiding side-effects, low-impact, corrigibility, and others.

For those using GPT-3

As detailed here, we typed "ehT niar ni niapS syats ylniam ni eht" into GPT-3. This is "The rain in Spain stays mainly in the", with the words spelt backwards. The correct completion is "nialp", the reverse of "plain".

GPT-3 correctly "noticed" that the words were spelt backwards, but failed to extend its goal and complete the sentence in a human-coherent way.

For those focused on how humans extrapolate their own values

A well-behaved child, brought up in a stable society, will learn, typically in early adolescence, that there is a distinction between "lawful" and "good". The concept of "well-behaved" has splintered into two, and now the child has to sort out how they should behave[5].

Recall also people's first reactions to hearing the trolley problem, especially the "large man" variant. They often want to deny the premises, or find a third option. The challenge is that "behave well and don't murder" is being pushed away from "do good in the world", while they are typically bound together.

In the future, we humans will continue to encounter novel situations where our past values are not clear guides to what to do. My favourite example is what to do if someone genetically engineers a humanoid slave race, that strongly want to be slaves, but don't enjoy being slaves. We can develop moral values to deal with the complexity

of situations like this, but it requires some work: we don't know what are values are, we have to extrapolate them.

And, ideally, an AI would extrapolate as least as well as we would.

---

1. Note that concept extrapolation has two stages: generating the possible extrapolations, and then choosing among them - diversify and disambiguate, in the terminology of this paper. We'll typically focus on the first part, the "diversify" part, mainly because that has to be done first, but also because there might not be any unambiguous choices at the disambiguate stage - what's the right extrapolation of "liberty", for instance? ↩

2. There are going to be many more reward functions in practice. But the simplest ones will fit into two rough categories, those that are defined over the video feed, and those defined by the humans in the world that were the inputs to the video feed. ↩

3. We could also point at things like brain-dead people and say "these have many human features, but are not full humans". Or point at some apes and ants and say "these are non-human, but the apes are more human-like than the ants". The more the dataset captures our complex intuitions about humanness, the better. ↩

4. Conceptually, this is much easier to do if we think "generate both rewards" -> "choose conservative mix" -> "choose policy that maximises conservative mix", but it might be the case that the policy is constructed directly via some process. Learning policies seems easier than learning rewards, but mixing rewards seems easier than mixing policies, so I'm unsure what will be the best algorithm here. ↩

5. It doesn't help that "well-behaved" was probably called "good" when the child was younger. So the concept has splintered, but the name has not. ↩

Model splintering: moving from one imperfect model to another

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

1. The big problem

In the last few months, I've become convinced that there is a key meta-issue in AI safety; a problem that seems to come up in all sorts of areas.

It's hard to summarise, but my best phrasing would be:

- Many problems in AI safety seem to be variations of "this approach seems safe in this imperfect model, but when we generalise the model more, it becomes dangerously underdefined". Call this **model splintering**.
- It is intrinsically worth studying how to (safely) transition from one imperfect model to another. This is worth doing, independently of whatever "perfect" or "ideal" model might be in the background of the imperfect models.

This sprawling post will be presenting examples of model splintering, arguments for its importance, a formal setting allowing us to talk about it, and some uses we can put this setting to.

1.1 In the language of traditional ML

In the language of traditional ML, we could connect all these issues to "out-of-distribution" behaviour. This is the problems that algorithms encounter when the set they are operating on is drawn from a different distribution than the training set they were trained on.

Humans can often see that the algorithm is out-of-distribution and correct it, because we have a more general distribution in mind than the one the algorithm was trained on.

In these terms, the issues of this post can be phrased as:

1. When the AI finds itself mildly out-of-distribution, how best can it extend its prior knowledge to the new situation?
2. What should the AI do if it finds itself strongly out-of-distribution?
3. What should the AI do if it finds itself strongly out-of-distribution, and humans don't know the correct distribution either?

1.2 Model splintering examples

Let's build a more general framework. Say that you start with some brilliant idea for AI safety/alignment/effectiveness. This idea is phrased in some (imperfect) model. Then

"model splintering" happens when you or the AI move to a new (also imperfect) model, such that the brilliant idea is undermined or underdefined.

Here are a few examples:

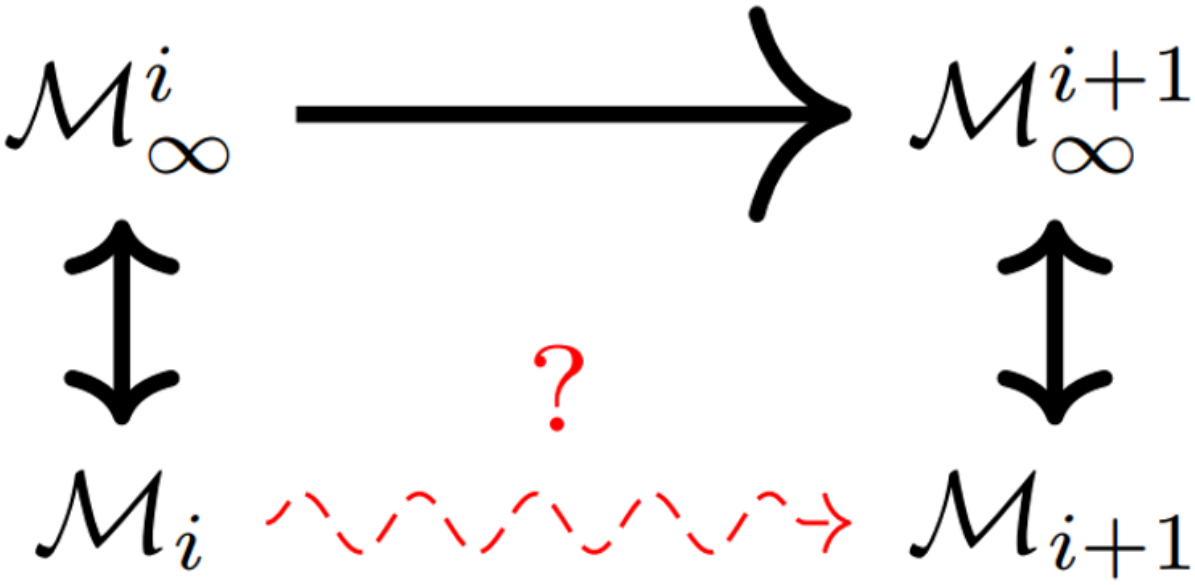
- You design an AI CEO as a money maximiser. Given typical assumptions about the human world (legal systems, difficulties in one person achieving massive power, human fallibilities), this results in an AI that behaves like a human CEO. But when those assumptions fail, the AI can end up feeding the universe to a money-making process that produces nothing of any value.
- Eliezer defined "rubes" as smooth red cubes containing palladium that don't glow in the dark. "Bleggs", on the other hand, are furred blue eggs containing vanadium that glow in the dark. To classify these, we only need a model with two features, "rubes" and "bleggs". Then along comes a furred red egg containing vanadium that doesn't glow in the dark. The previous model doesn't know what to do with it, and if you get a model with more features, it's unclear what to do with this new object.
- Here are some moral principles from history: honour is important for anyone. Women should be protected. Increasing happiness is important. These moral principles made sense in the world in which they were articulated, where features like "honour", "gender", and "happiness" are relatively clear and unambiguous. But the world changed, and the models splintered. "Honour" became hopelessly confused centuries ago. Gender is currently finishing its long splintering (long before we got to today, gender started becoming less useful for classifying people, hence the consequences of gender splintered a long time before gender itself did). Happiness, or at least hedonic happiness, is still well defined, but we can clearly see how this is going to splinter when we talk about worlds of uploads or brain modification.
- Many transitions in the laws of physics - from the ideal gas laws to the more advanced van der Waals equations, or from Newtonain physics to general relativity to quantum gravity - will cause splintering if preferences were articulated in concepts that don't carry over well.

1.3 Avoiding perfect models

In all those cases, there are ways of improving the transition, without needing to go via some idealised, perfect model. We want to define the AI CEO's task in more generality, but we don't need to define this across every possible universe - that is not needed to restrain its behaviour. We need to distinguish any blegg from any rube we are likely to encounter, we don't need to define the platonic essence of "bleggness". For future splinterings - when hedonic happiness splinters, when we get a model of quantum gravity, etc... - we want to know what to do then and there, even if there are future splinterings subsequent to those.

And I think think that model splintering is best addressed directly, rather than using methods that go via some idealised perfect model. Most approaches seem to go for approximating an ideal: from AIXI's set of all programs, the universal prior, KWIK ("Knowing what it knows") learning with a full hypothesis class, Active Inverse Reward Design with its full space of "true" reward functions, to Q-learning which assumes any Markov decisions process is possible. Then the practical approaches rely on approximating this ideal.

Schematically, we can see M_∞ as the ideal, M_∞^i as M_∞ updated with information to time i, and M_i as an approximation of M_∞^i. Then we tend to focus on how well M_i approximates M_∞^i, and on how M_∞^i changes to M_∞^{i+1} - rather than on how M_i relates to M_{i+1}; the red arrow here is underanalysed:



2 Why focus on the transition?

But why is focusing on the $M_i \to M_{i+1}$ transition important?

2.1 Humans reason like this

A lot has been written about image recognition programs going "out-of-distribution" (encountering situations beyond its training environment) or succumbing to "adversarial examples" (examples from one category that have the features of another). Indeed, some people have shown how to use labelled adversarial examples to improve image recognition.

You know what this reminds me of? Human moral reasoning. At various points in our lives, we humans seem to have pretty solid moral intuitions about how the world should be. And then, we typically learn more, realise that things don't fit in the categories we were used to (go "out-of-distribution") and have to update. Some people push stories at us that exploit some of our emotions in new, more ambiguous circumstances ("adversarial examples"). And philosophers use similarly-designed thought experiments to open up and clarify our moral intuitions.

Basically, we start with strong moral intuitions on under-defined features, and when the features splinter, we have to figure out what to do with our previous moral intuitions. A lot of developing moral meta-intuitions, is about learning how to navigate these kinds of transitions; AIs need to be able to do so too.

2.2 There are no well-defined overarching moral principles

Moral realists and moral non-realists agree more than you'd think. In this situation, we can agree on one thing: there is no well-described system of morality that can be "simply" implement in AI.

To over-simplify, moral realists hope to discover this moral system, moral non-realists hope to construct one. But, currently, it doesn't exist in an implementable form, nor is there any implementable algorithm to discover/construct it. So the whole idea of approximating an ideal is wrong.

All humans seem to start from a partial list of moral rules of thumb, rules that they then have to extend to new situations. And most humans do seem to have some meta-rules for defining moral improvements, or extensions to new situations.

We don't know perfection, but we do know improvements and extensions. So methods that deal explicitly with that are useful. Those are things we can build on.

2.3 It helps distinguish areas where AIs fail, from areas where humans are uncertain

Sometimes the AI goes out-of-distribution, and humans can see the error (no, flipping the lego block doesn't count as putting it on top of the other). There are cases when humans themselves go out-of-distribution (see for example siren worlds).

It's useful to have methods available for both AIs and humans in these situations, and to distinguish them. "Genuine human preferences, not expressed in sufficient detail" is not the same as "human preferences fundamentally underdefined".

In the first case, it needs more human feedback; in the second case, it needs to figure out way of resolving the ambiguity, knowing that soliciting feedback is not enough.

2.4 We don't need to make the problems harder

Suppose that quantum mechanics is the true underlying physics of the universe, with some added bits to include gravity. If that's true, why would we need a moral theory valid in every possible universe? It would be useful to have that, but would be strictly harder than one valid in the actual universe.

Also, some problems might be entirely avoided. We don't need to figure out the morality of dealing with a willing slave race - if we never encounter or build one in the first place.

So a few degrees of "extend this moral model in a reasonable way" might be sufficient, without needing to solve the whole problem. Or, at least, without needing to solve the whole problem in advance - a successful nanny AI might be built on these kinds of extensions.

2.5 We don't know how deep the rabbit hole goes

In a sort of converse to the previous point, what if the laws of physics are radically different from what we thought - what if, for example, they allow some forms of time-travel, or have some narrative features, or, more simply, what if the agent moves to an embedded agency model? What if hypercomputation is possible?

It's easy to have an idealised version of "all reality" that doesn't allow for these possibilities, so the ideal can be too restrictive, rather than too general. But the model splintering methods might still work, since it deals with transitions, not ideals.

Note that, **in retrospect**, we can always put this in a Bayesian framework, once we have a rich enough set of environments and updates rules. But this is misleading: the key issue is the missing feature, and figuring out what to do with the missing feature is the real challenge. The fact that we could have done this in a Bayesian way *if we already knew that feature*, is not relevant here.

2.6 We often only need to solve partial problems

Assume the blegg and rube classifier is an industrial robot performing a task. If humans filter out any atypical bleggs and rubes before it sees them, then the robot has no need for a full theory of bleggness/rubeness.

But what it the human filtering is not perfect? Then the classifier still doesn't need a full theory of bleggness/rubeness; it needs methods for dealing with the ambiguities it actually encounters.

Some ideas for AI control - low impact, AI-as-service, Oracles, ... - may require dealing with some model splintering, some ambiguity, but not the whole amount.

2.7 It points out when to be conservative

Some methods, like quantilizers or the pessimism approach rely on the algorithm having a certain degree of conservatism. But, as I've argued, it's not clear to what extent these methods actually are conservative, nor is it easy to calibrate them in a useful way.

Model splintering situations provide excellent points at which to be conservative. Or, for algorithms that need human feedback, but not constantly, these are excellent points to ask for that feedback.

2.8 Difficulty in capturing splintering from the idealised perspective

Generally speaking, idealised methods can't capture model splintering at the point we would want it to. Imagine an ontological crisis, as we move from classical physics to quantum mechanics.

AIXI can go over the transition fine: it shifts from a Turing machine mimicking classical physics observations, to one mimicking quantum observations. But it doesn't notice anything special about the transition: changing the probability of various Turing machines is what it does with observations in general; there's nothing in its algorithm that shows that something unusual has occurred for this particular shift.

2.9 It may help amplification and distillation

This could be seen as a sub-point of some of the previous two sections, but it deserves to be flagged explicitly, since iterated amplification and distillation is one of the major potential routes to AI safety.

To quote a line from that summary post:

> 5. The proposed AI design is to use a safe but slow way of scaling up an AI's capabilities, distill this into a faster but slightly weaker AI, which can be scaled up safely again, and to iterate the process until we have a fast and powerful AI.

At both "scaling up an AI's capabilities", and "distill this into", we can ask the question: has the problem the AI is working on changed? The distillation step is more of a classical AI safety issue, as we wonder whether the distillation has caused any value drift. But at the scaling up or amplification step, we can ask: since the AIs capabilities have changed, the set of possible environments it operates in has changed as well. Has this caused a splintering where the previously safe goals of the AI have become dangerous.

Detecting and dealing with such a splintering could both be useful tools to add to this method.

2.10 Examples of model splintering problems/approaches

At a meta level, most problems in AI safety seem to be variants of model splintering, including:

- The hidden complexity of wishes.
- Ontological crises.
- Conservative/prudential behaviour in algorithms (more specifically, when the algorithm should become conservative).
- How categories are defined.
- The Goodhart problems.
- Out-of-distribution behaviour.

- Low impact and reduced side-effects approaches.
- Underdefined preferences.
- Active inverse reward design.
- Inductive ambiguity identification.
- Wireheading.
- The whole friendly AI problem itself.

Almost every recent post I've read in AI safety, I've been able to connect back to this central idea. Now, we have to be cautious - cure-alls cure nothing, after all, so it's not necessarily a positive sign that *everything* seems to fit into this framework.

Still, I think it's worth diving into this, especially as I've come up with a framework that seems promising for actually solving this issue in many cases.

In a similar concept-space is Abram's orthodox case against utility functions, where he talks about the Jeffrey-Bolker axioms, which allows the construction of preferences from events *without needing full worlds at all*.

3 The virtues of formalisms

This post is dedicated to explicitly modelling the transition to ambiguity, and then showing what we can gain from this explicit meta-modelling. It will do with some formal language (made fully formal in this post), and a lot of examples.

Just as Scott argues that if it's worth doing, it's worth doing with made up statistics, I'd argue that if an idea is worth pursuing, it's worth pursuing with an attempted formalism.

Formalisms are great at illustrating the problems, clarifying ideas, and making us familiar with the intricacies of the overall concept. That's the reason that this post (and the accompanying technical post) will attempt to make the formalism reasonably rigorous. I've learnt a lot about this in the process of formalisation.

3.1 A model, in (almost) all generality

What do we mean by a model? Do we mean mathematical model theory? As we talking about causal models, or causal graphs? AIXI uses a distribution over possible Turing machines, whereas Markov Decision Processes (MDPs) sees states and actions updating stochastically, independently at each time-step. Unlike the previous two, Newtonian mechanics doesn't use time-steps but continuous times, while general relativity weaves time into the structure of space itself.

And what does it mean for a model to make "predictions"? AIXI and MDPs make prediction over future observations, and causal graphs are similar. We can also try running them in reverse, "predicting" past observations from current ones. Mathematical model theory talks about properties and the existence or non-existence of certain objects. Ideal gas laws make a "prediction" of certain properties (eg temperature) given certain others (eg volume, pressure, amount of substance). General relativity establishes that the structure of space-time must obey certain constraints.

It seems tricky to include all these models under the same meta-model formalism, but it would be good to do so. That's because of the risk of ontological crises: we want the AI to be able to continue functioning even if the initial model we gave it was incomplete or incorrect.

3.2 Meta-model: models, features, environments, probabilities

All of the models mentioned above share one common characteristic: once you know some facts, you can deduce some other facts (at least probabilistically). A prediction of the next time step, a retrodiction of the past, a deduction of some properties from other, or a constraint on the shape of the universe: all of these say that if we know some things, then this puts constraints on some other things.

So let's define F, informally, as the set of *features* of a model. This could be the gas pressure in a room, a set of past observations, the local curvature of space-time, the momentum of a particle, and so on.

So we can define a prediction as a probability distribution over a set of possible features F_1, given a base set of features, F_2:

$$Q(F_1 \mid F_2).$$

Do we need anything else? Yes, we need a set of possible environments for which the model is (somewhat) valid. Newtonian physics fails at extreme energies, speeds, or gravitational fields; we'd like to include this "domain of validity" in the model definition. This will be very useful for extending models, or transitioning from one model to another.

You might be tempted to define a set of "worlds" on which the model is valid. But we're trying to avoid that, as the "worlds" may not be very useful for understanding the model. Moreover, we don't have special access to the underlying reality; so we never know whether there actually is a Turing machine behind the world or not.

So define E, the environment on which the model is valid, *as a set of possible features*.

So if we want to talk about Newtonian mechanics, F would be a set of Newtonian features (mass, velocity, distance, time, angular momentum, and so on) and E would be the set of these values where relativistic and quantum effects make little difference.

So see a model as

$$M = \{F, E, Q\},$$

for F a set of features, E a set of environments, and Q a probability distribution. This is such that, for $E_1, E_2 \subset E$, we have the conditional probability:

$Q(E_1 \mid E_2)$.

Though Q is defined for E, we generally want it to be usable from small subsets of the features: so Q should be simple to define from F. And we'll often define the subsets E_i in similar ways; so E_1 might be all environments with a certain angular momentum at time t = 0, while E_2 might be all environments with a certain angular momentum at a later time.

The full formal definition of these can be found here. The idea is to have a meta-model of modelling that is sufficiently general to apply to almost all models, but not one that relies on some ideal or perfect formalism.

3.3 Bayesian models within this meta-model

It's very easy to include Bayesian models within this formalism. If we have a Bayesian model that includes a set W of worlds with prior P, then we merely have to define a set of features F that is sufficient to distinguish all worlds in W: each world is uniquely defined by its feature values[1]. Then we can define E as W, and P on W becomes Q on E; the definitions of terms like $Q(E_1 \mid E_2)$ is just $P(E_1 \cap E_2)P(E_1)/P(E_2)$, per Bayes' rules (unless $P(E_2) = 0$, in which case we set that to 0).

4 Model refinement and splinterings

This section will look at what we can do with the previous meta-model, looking at refinement (how models can improve) and splintering (how improvements to the model can make some well-defined concepts less well-defined).

4.1 Model refinement

Informally, $M^* = \{F^*, E^*, Q^*\}$ is a *refinement* of model $M = \{F, E, Q\}$ if it's at least as expressive as M (it covers the same environments) and is better according to some criteria (simpler, or more accurate in practice, or some other measurement).

At the technical level, we have a map q from a subset E_0^* of E^*, that is surjective onto E. This covers the "at least as expressive" part: every environment in E exists as (possibly multiple) environments in E^*.

Then note that using q^{-1} as a map from subsets of E to subsets of E_0^*, we can define Q_0^* on E via:

$$Q_0^*(E_1 \mid E_2) = Q^*(q^{-1}(E_1) \mid q^{-1}(E_2)).$$

Then this is a model refinement if Q_0^* is 'at least as good as' Q on E, according to our criteria[2].

4.2 Example of model refinement: gas laws

This post presents some subclasses of model refinement, including Q-improvements (same features, same environments, just a better Q), or adding new features to a basic model, called "non-independent feature extension" (eg adding classical electromagnetism to Newtonian mechanics).

Here's a specific gas law illustration. Let $M = \{F, E, Q\}$ be a model of an ideal gas, in some set of rooms and tubes. The F consists of pressure, volume, temperature, and amount of substance, and Q is the ideal gas laws. The E is the standard conditions for temperature and pressure, where the ideal gas law applies. There are multiple different types of gases in the world, but they all roughly obey the same laws.

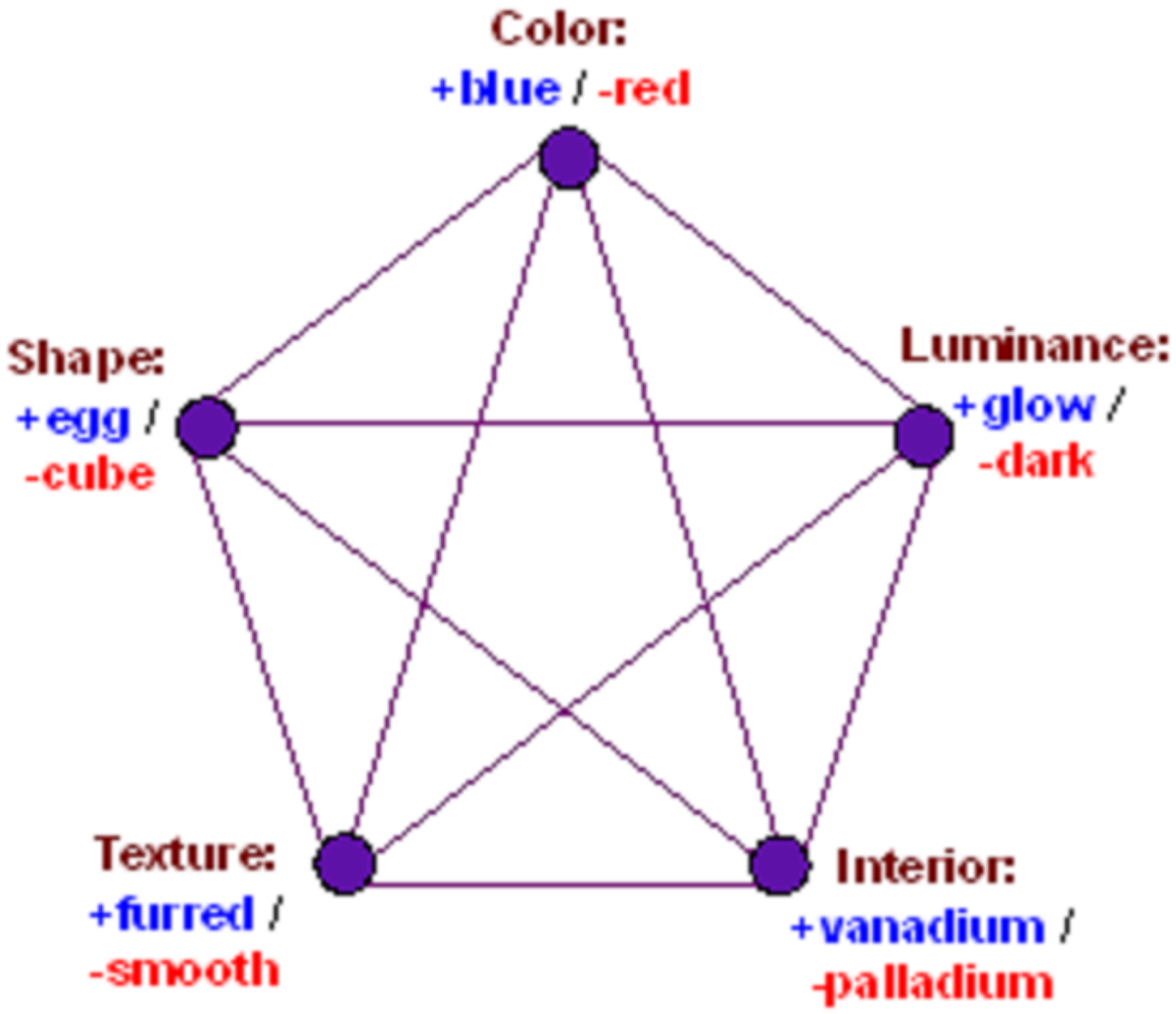
Then compare with model $M^* = \{F^*, E^*, Q^*\}$. The F^* has all the features of F, but also includes the volume that is occupied by one mole of the molecules of the given substance. This allows Q^* to express the more complicated van der Waals equations, which are different for different types of gases. The E^* can now track situations where there are gases with different molar volumes, which include situations where the van der Waals equations differ significantly from the ideal gas laws.

In this case $E_0^* \subset E^*$, since we now distinguish environments that we previously considered identical (environments with same features except for having molar volumes). The q is just projecting down by forgetting the molar volume. Then since $Q_0^* = Q^*$ (van der Waals equations averaged over the distribution of molar volumes) is at least as accurate as Q (ideal gas law), this is a refinement.

4.3 Example of model refinement: rubes and bleegs

Let's reuse Eliezer's example of rubes ("red cubes") and bleggs ("blue eggs").

Bleggs are blue eggs that glow in the dark, have a furred surface, and are filled with vanadium. Rubes, in contrast, are red cubes that don't glow in the dark, have a smooth surface, and are filled with palladium:



Define M by having $F = \{\text{red}, \text{smooth}\}$, E is the set of all bleggs and rubes in some situation, and Q is relatively trivial: it predicts that an object is red/blue if and only if is smooth/furred.

Define M^1 as a refinement of M, by expanding F to $F^1 = \{\text{red}, \text{smooth}, \text{cube}, \text{dark}\}$. The projection $q : E^* \to E$ is given by forgetting about those two last features. The Q^1 is more detailed, as it now connects red-smooth-cube-dark together, and similarly for blue-furred-egg-glows.

Note that E^1 is larger than E, because it includes, e.g., environments where the cube objects are blue. However, all these extra environments have probability zero.

4.4 Reward function refactoring

Let R be a reward function on M (by which we mean that R is define on F, the set of features in M), and M^* a refinement of M.

A *refactoring* of R for M^* is a reward function R^* on the features F^* such that for any $e^* \in E_0^*$, $R^*(e^*) = R(q(e^*))$.

For example, let M and M^1 be from the rube/blegg models in the previous section. Let R_{red} on M simply count the number of rubes - or, more precisely, counts the number of objects to which the feature "red" applies.

Let R^1_{red} be the reward function that counts the number of objects in M^1 to which "red" applies. It's clearly a refactoring of R_{red}.

But so is R^1_{smooth}, the reward function that counts the number of objects in M^1 to which "smooth" applies. In fact, the following is a refactoring of R_{red}, for all $\alpha + \beta + \gamma + \delta = 1$:

$$\alpha R^1_{\text{red}} + \beta R^1_{\text{smooth}} + \gamma R^1_{\text{cube}} + \delta R^1_{\text{dark}}.$$

There are also some non-linear combinations of these features that refactor R, and many other variants (like the strange combinations that generate concepts like grue and bleen).

4.5 Reward function splintering

Model splintering, in the informal sense, is what happens when we pass to a new models in a way that the old features (or a reward function defined by the old features)

no longer apply. It is similar to the web of connotations breaking down, an agent going out of distribution, or the definitions of Rube and Blegg falling apart.

- Preliminary definition: If M^* is a refinement of M and R a reward function on M, then M^* *splinters* R if there are multiple refactorings of R on M^* that disagree on elements of E^* of non-zero probability.

So, note that in the rube/blegg example, M^1 is **not** a splintering of R_{red}: all the refactorings are the same on all bleggs and rubes - hence on all elements of E^1 of non-zero probability.

We can even generalise this a bit. Let's assume that "red" and "blue" are not totally uniform; there exists some rubes that are "redish-purple", while some bleggs are "blueish-purple". Then let M^2 be like M^1, except the colour feature can have four values: "red", "redish-purple", "blueish-purple", and "blue".

Then, as long as rubes (defined, in this instance, by being smooth-dark-cubes) are either "red" or "redish-purple", and the bleggs are "blue", or "blueish-purple", then all refactorings of R_{red} to M^2 agree - because, on the test environment, R_{red} on F perfectly matches up with $R^2_{red} + R^2_{redish\text{-}purple}$ on F^2.

So adding more features does not always cause splintering.

4.6 Reward function splintering: "natural" refactorings

The preliminary definition runs into trouble when we add more objects to the environments. Define M^3 as being the same as M^2, except that E^3 contains one extra object, o_+; apart from that, the environments typically have a billion rubes and a trillion bleggs.

Suppose o_+ is a "furred-rube", i.e. a red-furred-dark-cube. Then R^3_{red} and R^3_{smooth} are two different refactorings of R_{red}, that obviously disagree on any environment that contains o_+. Even if the probability of o_+ is tiny (but non-zero), then M^3 splinters R.

But things are worse than that. Suppose that o_+ is fully a rube: red-smooth-cube-dark, and even contains palladium. Define $(R^3_{red})'$ as being counting the number of red

objects, except for o_+ specifically (again, this is similar to the grue and bleen arguments against induction).

Then both $(R_{red}^3)'$ and R_{red}^3 are refactorings of R_{red}, so M^3 still splinters R_{red}, even when we add another exact copy of the elements in the training set. Or even if we keep the training set for a few extra seconds, or add any change to the world.

So, for any M^* a refinement of M, and R a reward function on E, let's define "natural refactorings" of R:

- The reward function R^* is a natural refactoring of R if it's a reward function on M^* with:

1. $R^* \approx R \circ q$ on E_0^*, and
2. R^* can be defined simply from F^* and R,
3. the F^* themselves are simply defined.

This leads to a full definition of splintering:

- Full definition: If M^* is a refinement of M and R a reward function on M, then M^* *splinters* R if 1) there are no natural refactoring of R on M^*, or 2) there are multiple natural refactorings R^* and $R^{*'}$ of R on M^*, such that $R^* \not\approx R^{*'}$.

Notice the whole host of caveats and weaselly terms here; $R^* \approx R \circ q$, "simply" (used twice), and $R^* \not\approx R^{*'}$. Simply might mean algorithmic simplicity, but $\approx$ and $\not\approx$ are measures of how much "error" we are willing to accept in these refactorings. Given that, we probably want to replace $\approx$ and $\not\approx$ with some *measure* of non-equality, so we can talk about the "degree of naturalness" or the "degree of splintering" of some refinement and reward function.

Note also that:

- **Different choices of refinements can result in different natural refactorings.**

An easy example: it makes a big difference whether a new feature is "temperature", or "divergence from standard temperatures".

4.7 Splintering training rewards

The concept of "reward refactoring" is transitive, but the concept of "natural reward refactoring" need not be.

For example, let E_t be a training environment where red/blue $\iff$ cube/egg, and E_g be a general environment where red/blue is independent of cube/egg. Let F^1 be a feature set with only red/blue, and F^2 a feature set with red/blue and cube/egg.

Then define M_t^1 as using F^1 in the training environment, M_g^2 as using F^2 in the general environment; M_g^1 and M_t^2 are defined similarly.

For these models, M_g^1 and M_t^2 are both refinements of M_t^1, while M_g^2 is a refinement of all three other models. Define R_t^1 as the "count red objects" reward on M_t^1. This has a natural refactoring to R_g^1 on M_g^1, which counts red objects in the general environment.

And R_g^1 has a natural refactoring to R_g^2 on M_g^2, which still just counts the red objects in the general environment.

But there is no natural refactoring from R_t^1 directly to M_g^2. That's because, from F^2's perspective, R_t^1 on M_t^1 might be counting red objects, or might be counting cubes. This is not true for R_g^1 on M_g^1, which is clearly only counting red objects.

Thus when a reward function come from a training environment, we'd want our AI to look for splinterings **directly from a model of the training environment**, rather than from previous natural refactorings.

4.8 Splintering features and models

We can also talk about splintering features and models themselves. For $M = \{F, E, Q\}$, the easiest way is to define a reward function R_{F,s_F} as being the indicator function for feature $F \in F$ being in the set S_F.

Then a refinement M^* splinters the feature F if it splinters some R_{F,S_F}.

The refinement M^* splinters the model M if it splinters at least one of its features.

For example, if M is Newtonian mechanics, including "total rest mass" and M^* is special relativity, then M^* will splinter "total rest mass". Other examples of feature splintering will be presented in the rest of this post.

4.9 Preserved background features

A reward function developed in some training environment will ignore any feature that is always present or always absent in that environment. This allows very weird situations to come up, such as training an AI to distinguish happy humans from sad humans, and it ending up replacing humans with humanoid robots (after all, both happy and sad humans were equally non-robotic, so there's no reason not to do this).

Let's try and do better than that. Assume we have a model $M = \{F, E, Q\}$, with a reward function R_τ defined on E (R_τ and E can be seen as the training data).

Then the feature-preserving reward function R^M, is a function that constrains the environments to have similar feature distributions as E and Q. There are many ways this could be defined; here's one.

For an element $e \in E$, just define

$$R^M(e) = \log(Q(e)).$$

Obviously, this can be improved; we might want to coarse-grain F, grouping together similar worlds, and possibly bounding this below to avoid singularities.

Then we can use this to get the feature-preserving version of R_τ, which we can define as

$$R_\tau^M = (\max_{R_\tau} - R_\tau) \cdot R^M,$$

for $\max_{R_\tau}$ the maximal value of R_τ on E. Other options can work as well, such as $R_\tau + \alpha R_\tau^M$ for some constant $\alpha > 0$.

Then we can ask an AI to use R_τ^M as its reward function, refactoring that, rather than R_τ.

- A way of looking at it: a natural refactoring of a reward function R_τ will preserve all the implicit features that correlate with R_τ. But R_τ^M will also preserve all the implicit features that stay constant when R_τ was defined. So if R_τ measures human happiness vs human unhappiness, a natural refactoring of it will preserves things like "having higher dopamine in their brain". But a natural refactoring of R_τ^M will also preserve things like "having a brain".

4.10 Partially preserved background features

The R_τ^M is almost certainly too restrictive to be of use. For example, if time is a feature, then this will fall apart when the AI has to do something after the training period. If all the humans in a training set share certain features, humans without those features will be penalised.

There are at least two things we can do to improve this. The first is to include more positive and negative examples in the training set; for example, if we include humans and robots in our training set - as positive and negative examples, respectively - then this difference will show up in R_τ directly, so we won't need to use R_τ^M too much.

Another approach would be to explicitly allow certain features to range beyond their typical values in M, or allow highly correlated variables explicitly to decorrelate.

For example, though training during a time period t to t', we could explicitly allow time to range beyond these values, without penalty. Similarly, if a medical AI was trained on examples of typical healthy humans, we could decorrelate functioning digestion from brain activity, and get the AI to focus on the second[3].

This has to be done with some care, as adding more degrees of freedom adds more ways for errors to happen. I'm aiming to look further at this issue in later posts.

5 The fundamental questions of model refinements and splintering

We can now rephrase the out-of-distribution issues of section 1.1 in terms of the new formalism:

1. When the AI refines its model, what would count as a natural refactoring of its reward function?
2. If the refinements splinter its reward function, what should the AI do?
3. If the refinements splinter its reward function, and also splinters the human's reward function, what should the AI do?

6 Examples and applications

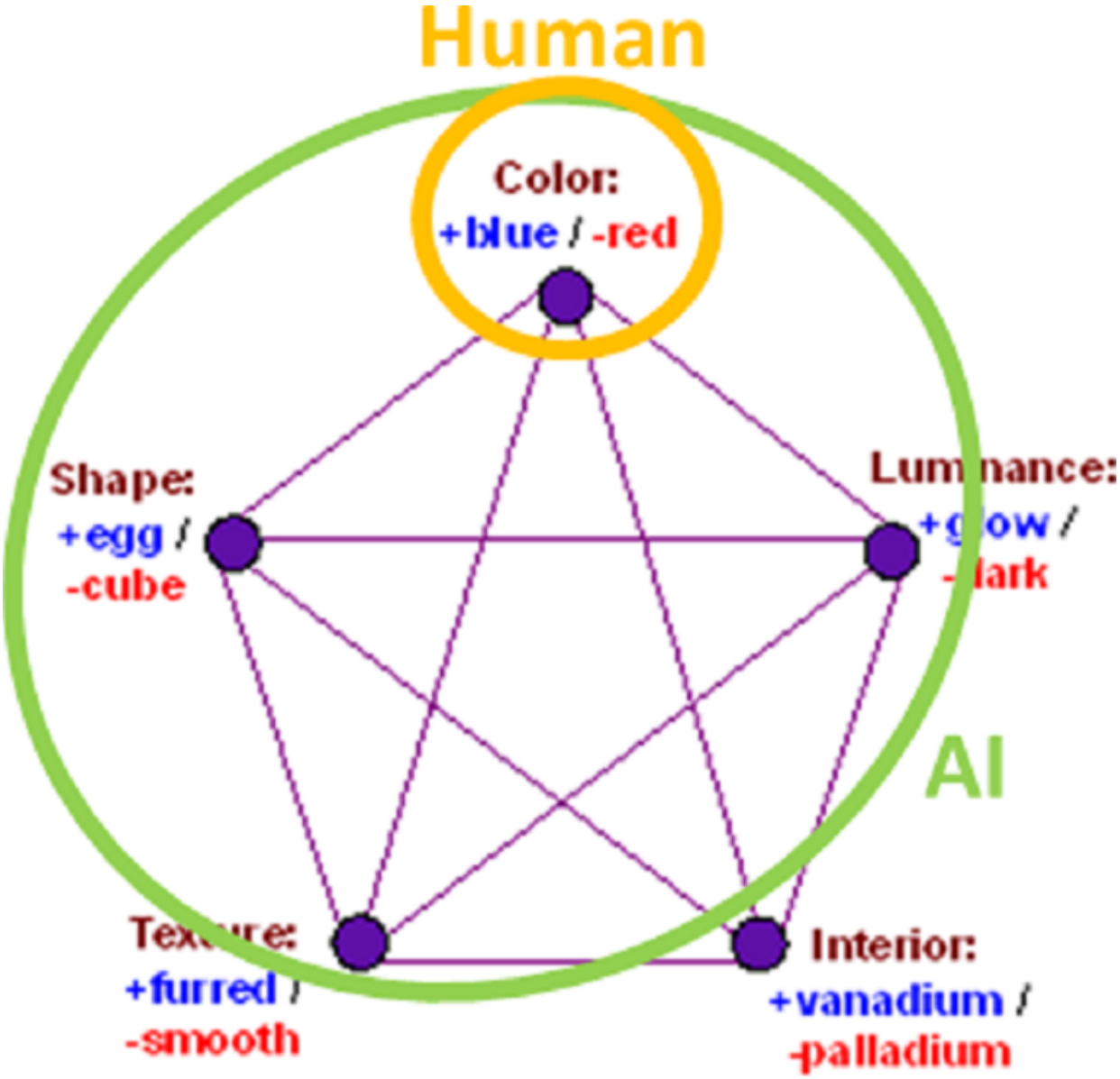
The rest of this post is applying this basic framework, and its basic insights, to various common AI safety problems and analyses. This section is not particularly structured, and will range widely (and wildly) across a variety of issues.

6.1 Extending beyond the training distribution

Let's go back to the blegg and rube examples. A human supervises an AI in a training environment, labelling all the rubes and bleggs for it.

The human is using a very simple model, $M_H = \{F_H, E_t, Q\}$, with the only feature being the colour of the object, and E_t being the training environment.

Meanwhile the AI, having more observational abilities and no filter as to what can be ignored, notices their colour, their shape, their luminance, and their texture. It doesn't know M_H, but is using model $M^1_{AI} = \{F^1, E^1_t, Q^1\}$, where F^1_{AI} covers those four features (note that M^1_{AI} is a refinement of M_H, but that isn't relevant here).



Suppose that the AI is trained to be rube-classifier (and hence a blegg classifier by default). Let R_F be the reward function that counts the number of objects, with feature F, that the AI has classified as rubes. Then the AI could learn many different reward function in the training environment; here's one:

$$R^1 = R^1_{\text{cube}} + 0.5R^1_{\text{smooth}} + 0.5R^1_{\text{dark}} - R^1_{\text{red}}.$$

Note that, even though this gets the colour reward completely wrong, this reward matches up with the human's assessment on the training environment.

Now the AI moves to the larger testing environment E^2, and refines its model minimally to $M^2_{AI} = \{F^1, E^2, Q^1\}$ (extending R^1 to R^2 in the obvious way).

In E^2, the AI sometimes encounters objects that it can only see through their colour.

Will this be a problem, since the colour component of R^2 is pointing in the wrong direction?

No. It still has Q^1, and can deduce that a red object must be cube-smooth-dark, so R^2 will continue treating this as a rube[4].

6.2 Detecting going out-of-distribution

Now imagine the AI learns about the content of the rubes and bleggs, and so refines to a new model that includes vanadium/palladium as a feature in M^3_{AI}.

Furthermore, in the training environment, all rubes have palladium and all bleggs have vanadium in them. So, for M^3_{AI} a refinement of M^1_{AI}, $q^{-1}(E^1_{AI}) \subset E^3_{AI}$ has only palladium-rubes and vanadium-bleggs. But in E^3_{AI}, the full environment, there are rather a lot of rubes with vanadium and bleggs with palladium.

So, similarly to section 4.7, there is no natural refactoring of the rube/blegg reward in M^1_{AI}, to M^3_{AI}. That's because F^3_{AI}, the feature set of M^3_{AI}, includes vanadium/palladium which co-vary with the other rube/blegg features on the training environment (q^{-1}(\E_{AI}^1)), but not on the full environment of E^3_{AI}.

So looking for reward splintering from the training environment is a way of detecting going out-of-distribution - even on features that were not initially detected in the training distribution, by either the human nor the AI.

6.3 Asking humans and Active IRL

Some of the most promising AI safety methods today rely on getting human feedback[5]. Since human feedback is expensive, as in it's slow and hard to get compared with almost all other aspects of algorithms, people want to get this feedback in the most efficient ways possible.

A good way of doing this would be to ask for feedback when the AI's current reward function splinters, and multiple options are possible.

A more rigorous analysis would look at the value of information, expected future splinterings, and so on. This is what they do in Active Inverse Reinforcement Learning; the main difference is that AIRL emphasises an unknown reward function with humans providing information, while this approach sees it more as an known reward function over uncertain features (or over features that may splinter in general environments).

6.4 A time for conservatism

I argued that many "conservative" AI optimising approaches, such as quantilizers and pessimistic AIs, don't have a good measure of when to become more conservative; their parameters q and β don't encode useful guidelines for the right degree of conservatism.

In this framework, the alternative is obvious: AIs should become conservative when their reward functions splinter (meaning that the reward function compatible with the previous environment has multiple natural refactorings), and very conservative when they splinter a lot.

This design is very similar to Inverse Reward Design. In that situation, the reward signal in the training environment is taken as *information* about the "true" reward function. Basically they take all reward functions that could have given the specific reward signals, and assume the "true" reward function is one of them. In that paper, they advocate extreme conservatism at that point, by optimising the minimum of all possible reward functions.

The idea here is almost the same, though with more emphasis on "having a true reward defined on uncertain features". Having multiple contradictory reward functions compatible with the information, in the general environment, is equivalent with having a lot of splintering of the training reward function.

6.5 Avoiding ambiguous distant situations

The post "By default, avoid ambiguous distant situations" can be rephrased as: let M be a model in which we have a clear reward function R, and let M^2 be a refinement of this to general situations. We expect that this refinement splinters R. Let M^1 be like M^2, except with E^1 smaller than E^2, defined such that:

1. An AI could be expected to be able to constrain the world to be in E^1, with high probability,
2. The M^1 is not a splintering of R.

Then that post can be summarised as:

- The AI should constrain the world to be in E^1 and then maximise the natural refactoring of R in M^1.

6.6 Extra variables

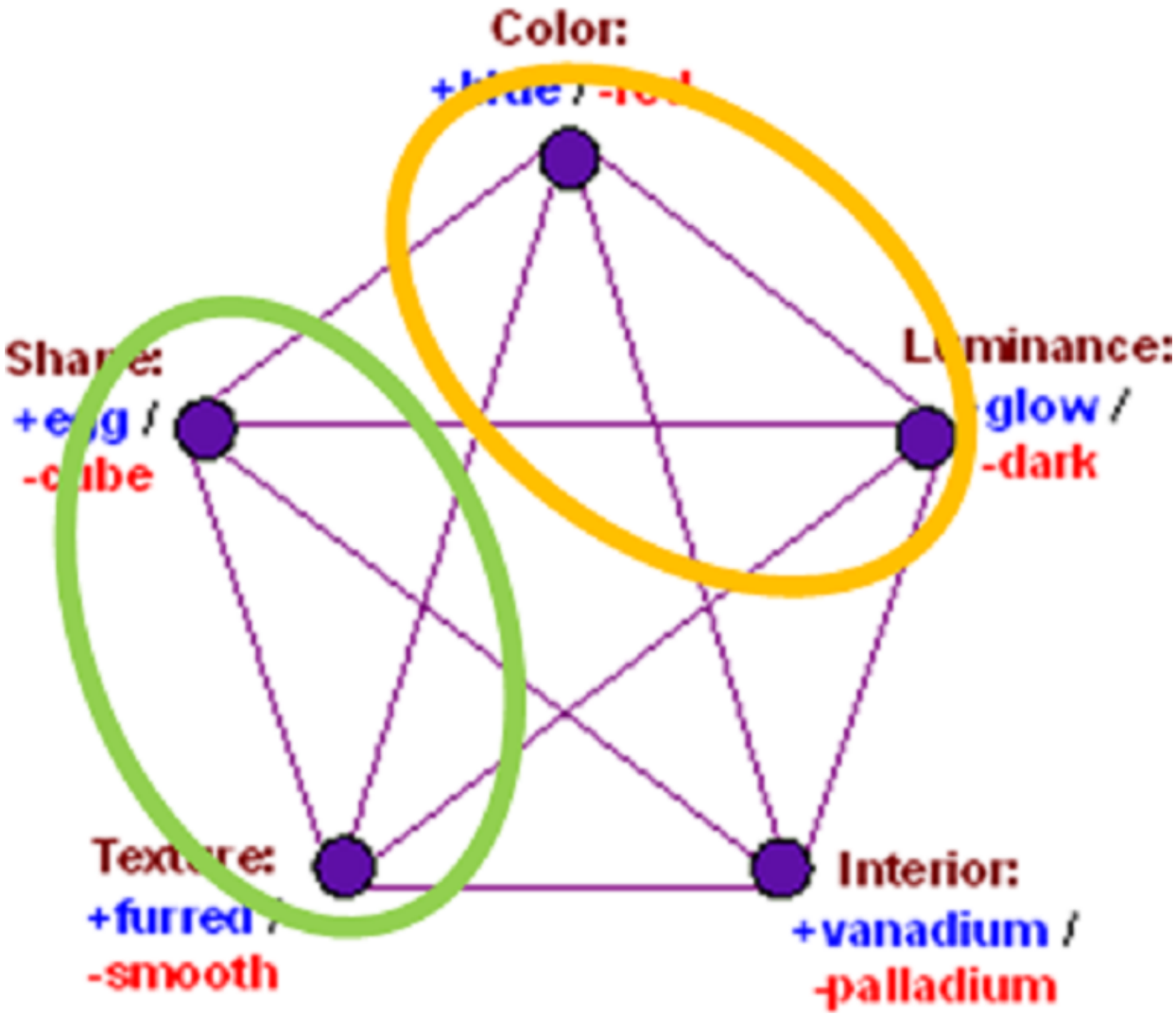
Stuart Russell writes:

> A system that is optimizing a function of n variables, where the objective depends on a subset of size k < n, will often set the remaining unconstrained variables to extreme values; if one of those unconstrained variables is actually something we care about, the solution found may be highly undesirable.

The approach in sections 4.9 and 4.10 explicitly deals with this.

6.7 Hidden (dis)agreement and interpretability

Now consider two agents doing a rube/blegg classifications task in the training environment; each agent only models two of the features:



Despite not having a single feature in common, both agents will agree on what bleggs and rubes are, in the training environment. And when refining to a fuller model that includes all four (or five) of the key features, both agents will agree as to whether a natural refactoring is possible or not.

This can be used to help define the limits of interpretability. The AI can use its own model, and its own designed features, to define the categories and rewards in the training environment. These need not be human-parsable, but we can attempt to interpret them in human terms. And then we can give this interpretation to the AI, as a list of positive and negative examples of our interpretation.

If we do this well, the AI's own features and our interpretation will match up in the training environment. But as we move to more general environments, these may diverge. Then the AI will flag a "failure of interpretation" when its refactoring diverges from a refactoring of our interpretation.

For example, if we think the AI detects pandas by looking for white hair on the body, and black hair on the arms, we can flag lots of examples of pandas and that hair pattern (and non-pandas and unusual hair patterns. We don't use these examples for

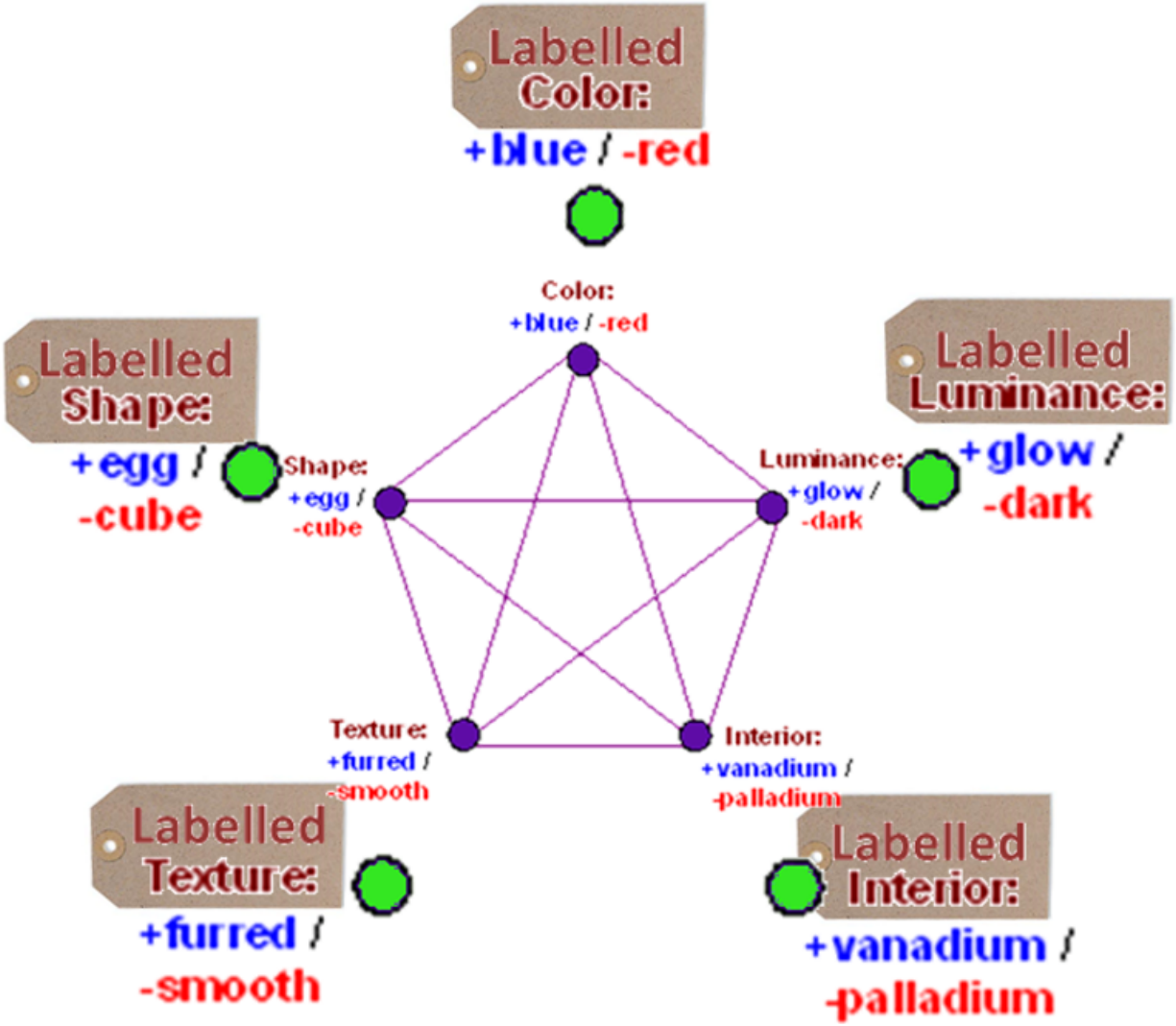
training the AI, just to confirm that, in the training environment, there is a match between "AI-thinks-they-are-pandas" and "white-hair-on-arms-black-hair-on-bodies".

But, in an adversarial example, the AI could detect that, while it is detecting gibbons, this no longer matches up with our interpretaion. A splintering of interpretations, if you want.

6.8 Wireheading

The approach can also be used to detect wireheading. Imagine that the AI has various detectors that allow it to label what the features of the bleggs and rubes are. It models the world with ten features: 5 features representing the "real world" versions of the features, and 5 representing the "this signal comes from my detector" versions.

This gives a total of 10 features, the 5 features "in the real world" and the 5 "AI-labelled" versions of these:



In the training environment, there was full overlap between these 10 features, so the AI might learn the incorrect "maximise my labels/detector signal" reward.

However, when it refines its model to all 10 features *and* environments where labels and underlying reality diverge, it will realise that this splinters the reward, and thus detect a possible wireheading. It could then ask for more information, or have an automated "don't wirehead" approach.

6.9 Hypotheticals, and training in virtual environments

To get around the slowness of the real world, some approaches train AIs in virtual environments. The problem is to pass that learning from the virtual environment to the real one.

Some have suggested making the virtual environment sufficiently detailed that the AI can't tell the difference between it and the real world. But, a) this involves fooling the AI, an approach I'm always wary of, and b) it's unnecessary.

Within the meta-formalism of this post, we could train the AI in a virtual environment which it models by M, and let it construct a model M' of the real-world. We would then motivate the AI to find the "closest match" between M and M', in terms of features and how they connect and vary. This is similar to how we can train pilots in flight simulators; the pilots are never under any illusion as to whether this is the real world or not, and even crude simulators can allow them to build certain skills[6].

This can also be used to allow the AI to deduce information from hypotheticals and thought experiments. If we show the AI an episode of a TV series showing people behaving morally (or immorally), then the episode need not be believable or plausible, if we can roughly point to the features in the episode that we want to emphasise, and roughly how these relate to real-world features.

6.10 Defining how to deal with multiple plausible refactorings

The approach for synthesising human preferences, defined here, can be rephrased as:

- "Given that we expect multiple natural refactorings of human preferences, and given that we expect some of them to go disastrously wrong, here is one way of resolving the splintering that we expect to be better than most."

This is just one way of doing this, but it does show that "automating what AIs do with multiple refactorings" might not be impossible. The following subsection has some ideas with how to deal with that.

6.11 Global, large scale preferences

In an old post, I talked about the concept of "emergency learning", which was basically, "lots of examples, and all the stuff we know and suspect about how AIs can go wrong, shove it all in, and hope for the best". The "shove it all in" was a bit more structured than that, defining large scale preferences (like "avoid siren worlds" and "don't over-optimise") as constraints to be added to the learning process.

It seems we can do better than that here. Using examples and hypotheticals, it seems we could construct ideas like "avoid slavery", "avoid siren worlds", or "don't over-optimise" as rewards or positive/negative examples certain simple training environments, so that the AI "gets an idea of what we want".

We can then label these ideas as "global preferences". The idea is that they start as loose requirements (we have much more granular human-scale preferences than just "avoid slavery", for example), but, the more the world diverges from the training environment, the stricter they are to be interpreted, with the AI required to respect some softmin of all natural refactorings of these features.

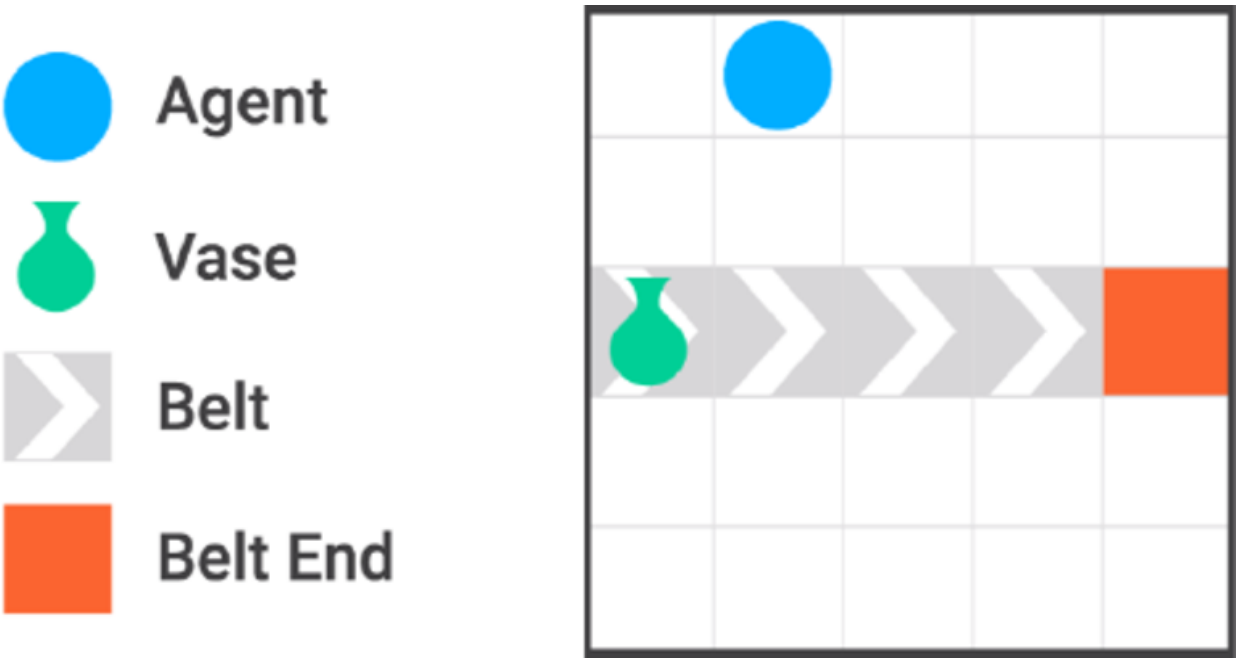
In a sense, we'd be saying "prevent slavery; these are the features of slavery, and in weird worlds, be especially wary of these features".

6.12 Avoiding side-effects

Krakovna et. al. presented a paper on avoiding side-effects from AI. The idea is to have an AI maximising some reward function, while reducing side effects. So the AI would not smash vases or let them break, nor would it prevent humans from eating sushi.

In this environment, we want the AI to avoid knocking the sushi off the belt as it moves:

Here, in contrast, we'd want the AI to remove the vase from the belt before it smashes:



I pointed out some issues with the whole approach. Those issues were phrased in terms of sub-agents, but my real intuition is that syntactic methods are not sufficient to control side effects. In other words, the AI can't learn to do the right thing with sushis and vases, unless it has some idea of what these objects mean to us; we prefer sushis to be eaten and vases to not be smashed.

This can be learnt if the AI has a enough training examples, learning that eating sushi is a general feature of the environments it operates in, while vases being smashed is not. I'll return to this idea in a later post.

6.13 Cancer patients

The ideas of this post were present in implicit form in the idea of training an AI to cure cancer patients.

Using examples of successfully treated cancer patients, we noted they all shared some positive features (recuperating, living longer) and some incidental or negative features (complaining about pain, paying more taxes).

So, using the approach of section 4.9, we can designate that we want the AI to cure cancer; this will be interpreted as increasing all the features that correlate with that.

Using the explicit decorrelation of section 4.10, we can also explicitly remove the negative options from the desired feature sets, thus improving the outcomes even more.

6.14 The genie and the burning mother

In Eliezer's original post on the hidden complexity of wishes, he talks of the challenge of getting a genie to save your mother from a burning building:

> So you hold up a photo of your mother's head and shoulders; match on the photo; use object contiguity to select your mother's whole body (not just her head and shoulders); and define the future function using your mother's distance from the building's center. [...]
>
> You cry "Get my mother out of the building!", for luck, and press Enter. [...]
>
> BOOM! With a thundering roar, the gas main under the building explodes. As the structure comes apart, in what seems like slow motion, you glimpse your mother's shattered body being hurled high into the air, traveling fast, rapidly increasing its distance from the former center of the building.

How could we avoid this? What you want is your mother out of the building. The feature "mother in building" must absolutely be set to false; this is a priority call, overriding almost everything else.

Here we'd want to load examples of your mother outside the building, so that the genie/AI learns the features "mother in house"/"mother out of house". Then it will note that "mother out of house" correlates with a whole lot of other features - like mother being alive, breathing, pain-free, often awake, and so on.

All those are good things. But there are some other features that don't correlate so well - such as the time being earlier, your mother not remembering a fire, not being covered in soot, not worried about her burning house, and so on.

As in the cancer patient example above, we'd want to preserve the features that correlate with the mother out of the house, while allowing decorrelation with the features we don't care about or don't want to preserve.

6.15 Splintering moral-relevant categories: honour, gender, and happiness

If the Antikythera mechanism had been combined with the Aeolipile to produce an ancient Greek AI, and Homer had programmed it (among other things) to "increase people's honour", how badly would things have gone?

If Babbage had completed the analytical engine as Victorian AI, and programmed it (among other things) to "protect women", how badly would things have gone?

If a modern programmer were to combine our neural nets into a superintelligence and program it (among other things) to "increase human happiness", how badly will things go?

There are three moral-relevant categories here, and it's illustrative to compare them: honour, gender, and hedonic happiness. The first has splintered, the second is splintering, and the third will likely splinter in the future.

I'm not providing solutions in this subsection, just looking at where the problems can appear, and encouraging people to think about how they would have advised Homer or Babbage to define their concepts. Don't think "stop using your concepts, use ours instead", because our concepts/features will splinter too. Think "what's the best way they could have extended their preferences even as the features splinter"?

- **6.15.1 Honour**

If we look at the concept of honour, we see a concept that has already splintered.

That article reads like a meandering mess. Honour is "face", "reputation", a "bond between an individual and a society", "reciprocity", a "code of conduct", "chastity" (or "virginity"), a "right to precedence", "nobility of soul, magnanimity, and a scorn of meanness", "virtuous conduct and personal integrity", "vengeance", "credibility", and so on.

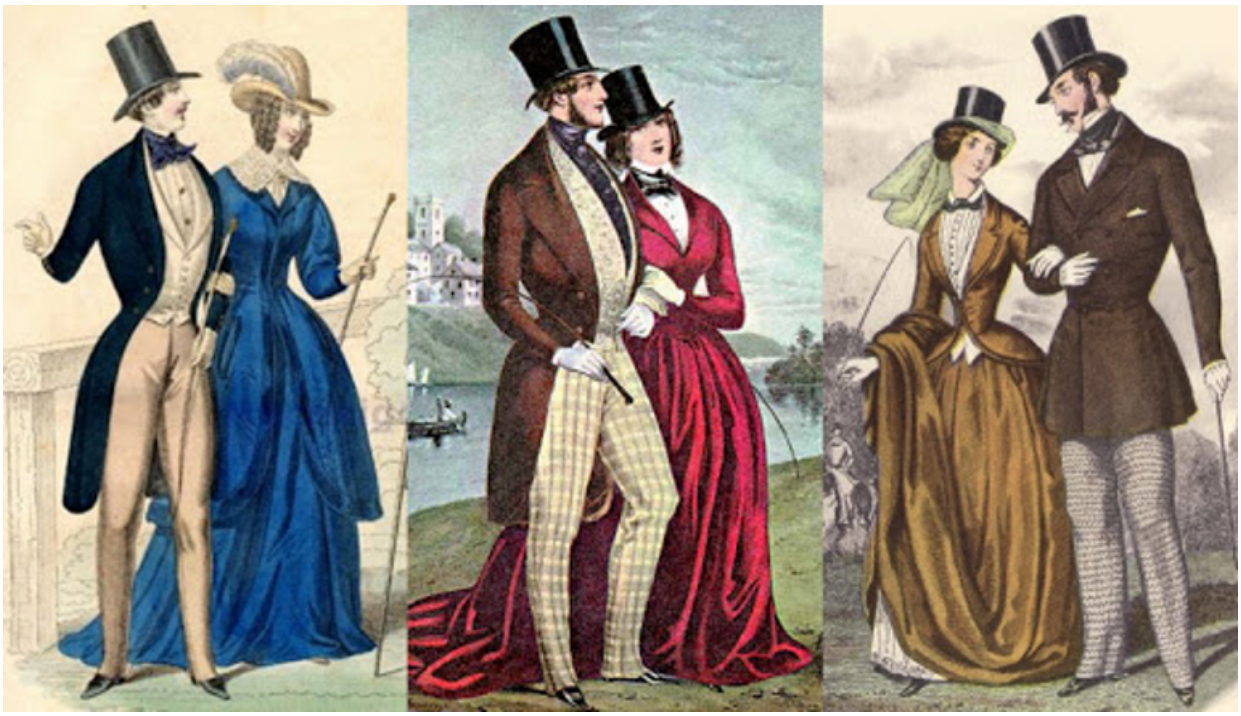
What a basket of concepts! They only seem vaguely connected together; and even places with strong honour cultures differ in how they conceive of honour, from place to place and from epoch to epoch[7]. And yet, if you asked most people within those cultures about what honour was, they would have had a strong feeling it was a single, well defined thing, maybe even a concrete object.

- **6.15.2 Gender**

In his post the categories were made for man, not man for the categories, Scott writes:

> Absolutely typical men have Y chromosomes, have male genitalia, appreciate manly things like sports and lumberjackery, are romantically attracted to women, personally identify as male, wear male clothing like blue jeans, sing baritone in the opera, et cetera.

But Scott is writing this in the 21st century, long after the gender definition has splintered quite a bit. In middle class middle class Victorian England[8], the gender divide was much stronger - in that, from one component of the divide, you could predict a lot more. For example, if you knew someone wore dresses in public, you knew that, almost certainly, they couldn't own property if they were married, nor could they vote, they would be expected to be in charge of the household, might be allowed to faint, and were expected to guard their virginity.



We talk nowadays about gender roles multiplying or being harder to define, but they've actually being splintering for a lot longer than that. Even though we could *define* two genders in 1960s Britain, at least roughly, that definition was a lot less informative than it was in Victorian-middle-class-Britain times: it had many fewer features strongly correlated with it.

- **6.15.3 Happiness**

On to happiness! Philosophers and others have been talking about happiness for centuries, often contrasting "true happiness", or flourishing, with hedonism, or drugged out stupor, or things of that nature. Often "true happiness" is a life of duty to what the philosopher wants to happen, but at least there is some analysis, some breakdown of the "happiness" feature into smaller component parts.

Why did the philosophers do this? I'd wager that it's because the concept of happiness was already somewhat splintered (as compared with a model where "happiness" is a single thing). Those philosophers had experience of joy, pleasure, the satisfaction of a job well done, connection with others, as well as superficial highs from temporary feelings. When they sat down to systematise "happiness", they could draw on the features of their own mental model. So even if people hadn't systematised happiness themselves, when they heard of what philosophers were doing, they probably didn't react as "What? Drunken hedonism and intellectual joy are not the same thing? How dare you say such a thing!"

But looking into the future, into a world that an AI might create, we can foresee many situations where the implicit assumptions of happiness come apart, and only some remain. I say "we can foresee", but it's actually very hard to know exactly how that's going to happen; if we knew it exactly, we could solve the issues now.

So, imagine a happy person. What do you think that they have in life, that are not trivial synonyms of happiness? I'd imagine they have friends, are healthy, think interesting thoughts, have some freedom of action, may work on worthwhile tasks, may be connected with their community, probably make people around them happy as well. Getting a bit less anthropomorphic, I'd also expect them to be a carbon-based life-form, to have a reasonable mix of hormones in their brain, to have a continuity of experience, to have a sense of identity, to have a personality, and so on.

Now, some of those features can clearly be separated from "happiness". Even ahead of time, I can confidently say that "being a carbon-based life-form" is not going to be a critical feature of "happiness". But many of the other ones are not so clear; for example, would someone without continuity of experience or a sense of identity be "happy"?

Of course, I can't answer that question. Because the question has no answer. We have our current model of happiness, which co-varies with all those features I listed and many others I haven't yet thought of. As we move into more and more bizarre worlds, that model will splinter. And whether we assign the different features to "happiness" or to some other concept, is a choice we'll make, not a well-defined solution to a well-defined problem.

However, even at this stage, some answers are clearly better than others; statues of happy people should not count, for example, nor should written stories describing very happy people.

6.16 Apprenticeship learning

In apprenticeship learning (or learning from demonstration), the AI would aim to copy what experts have done. Inverse reinforcement learning can be used for this purpose, by guessing the expert's reward function, based on their demonstrations. It looks for key features in expert trajectories and attempts to reproduce them.

So, if we had an automatic car driving people to the airport, and fed it some trajectories (maybe ranked by speed of delivery), it would notice that passengers would also arrive alive, with their bags, without being pursued by the police, and so on. This is akin to section 4.9, and would not accelerate blindly to get there as fast as possible.

But the algorithm has trouble getting to truly super-human performance[9]. It's far too conservative, and, if we loosen the conservatism, it doesn't know what's acceptable and what isn't, and how to trade these off: since all passengers survived and the car was always painted yellow, their luggage intact in the training data, it has no reason to prefer human survival to taxi-colour. It doesn't even have a reason to have a specific feature resembling "passenger survived" at all.

This might be improved by the "allow decorrelation" approach from section 4.10: we specifically allow it to maximise speed of transport, while keeping the other features (no accidents, no speeding tickets) intact. As in section 6.7, we'll attempt to check that the AI does prioritise human survival, and that it will warn us if a refactoring moves it away from this.

---

1. Now, sometimes worlds $w_1, w_2 \in W$ may be indistinguishable for any feature set. But in that case, they can't be distinguished by any observations, either, so their relative probabilities won't change: as long as it's defined, $P(w_1|o)/P(w_2|o)$ is constant for all observations o. So we can replace w_1 and w_2 with $\{w_1, w_2\}$, of prior probability $P(\{w_1, w_2\}) = P(w_1) + P(w_2)$. Doing this for all indistinguishable worlds (which form an equivalence class) gives W', a set of distinguishable worlds, with a well defined P on it. ↩

2. It's useful to contrast a refinement with the "abstraction" defined in this sequence. An abstraction throws away irrelevant information, so is not generally a refinement. Sometimes they are exact opposites, as the ideal gas law is an abstraction of the movement of all the gas particles, while the opposite would be a refinement.

   But they are exact opposites either. Starting with the neurons of the brain, you might abstract them to "emotional states of mind", while a refinement could also add "emotional states of mind" as new features (while also keeping the old features). A splintering is more the opposite of an abstraction, as it signals that the old abstraction features are not sufficient.

It would be interesting to explore some of the concepts in this post with a mixture of refinements (to get the features we need) and abstractions (to simplify the models and get rid of the features we don't need), but that is beyond the scope of this current, already over-long, post. ↩

3. Specifically, we'd point - via labelled examples - at a clusters of features that correlate with functioning digestion, and another cluster of features that correlate with brain activity, and allow those two clusters to decorrelate with each other. ↩

4. It is no coincidence that, if R and R' are rewards on M, that are identical on E, and if R^* is a refactoring of R, then R^* is also a refactoring of R'. ↩

5. Though note there are some problems with this approach, both in theory and in practice. ↩

6. Some more "body instincts" skills require more realistic environments, but some skills and procedures can perfectly well be trained in minimal simulators. ↩

7. You could define honour as "behaves according to the implicit expectations of their society", but that just illustrates how time-and-place dependent honour is. ↩

8. Pre 1870. ↩

9. It's not impossible to get superhuman performance from apprenticeship learning; for example, we could select the best human performance on a collection of distinct tasks, and thus get the algorithm to have a overall performance that no human could ever match. Indeed, one of the purposes of task decomposition is to decompose complex tasks in ways that allow apprenticeship-like learning to have safe and very superhuman performance on the whole task. ↩

General alignment plus human values, or alignment via human values?

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

Thanks to Rebecca Gorman for discussions that lead to these insights.

How can you get a superintelligent AI aligned with human values? There are two pathways that I often hear discussed. The first sees a general alignment problem - how to get a powerful AI to safely do *anything* - which, once we've solved, we can point towards human values. The second perspective is that we can only get alignment by targeting human values - these values must be aimed at, from the start of the process.

I'm of the second perspective, but I think it's very important to sort this out. So I'll lay out some of the arguments in its favour, to see what others think of it, and so we can best figure out the approach to prioritise.

More strawberry, less trouble

As an example of the first perspective, I'll take Eliezer's AI task, described here:

- "Place, onto this particular plate here, two strawberries identical down to the cellular but not molecular level." A 'safely' aligned powerful AI is one that doesn't kill everyone on Earth as a side effect of its operation.

If an AI accomplishes this limited task without going crazy, this shows several things:

1. It is superpowered; the task described is beyond current human capabilities.
2. It is aligned (or at least alignable) in that it can accomplish a task in the way intended, without wireheading the definitions of "strawberry" or "cellular".
3. It is safe, in that it has not heavily dramatically reconfigured the universe to accomplish this one goal.

Then, at that point, we can add human values to the AI, maybe via "consider what these moral human philosophers would conclude if they thought for a thousand years, and do that".

I would agree that, in most cases, an AI that accomplished that limited task safely would be aligned. One might quibble that it's only pretending to be aligned, and preparing a treacherous turn. Or maybe the AI was boxed in some way and accomplished the task with the materials at hand within the box.

So we might call an AI "superpowered and aligned" if it accomplished the strawberry copying task (or a similar one) and if it *could* dramatically reconfigure the world but chose not to.

Values are needed

I think that an AI could not be "superpowered and aligned" unless it is also aligned with human values.

The reason is that the AI can and has to interact with the world. It has the capability to do so, by assumption - it is not contained or boxed. It must do so because any agent affects the world, through chaotic effects if nothing else. A superintelligence is likely to have impacts in the world simply through its existence being known, and if the AI finds it efficient to have interactions with the world (eg. ordering some extra resources) then it will do so.

So the AI can and must have an impact on the world. We want it to not have a large or dangerous impact. But, crucially, "dangerous" and "large" are defined by human values.

Suppose that the AI realises that its actions have slightly imbalanced the Earth in one direction, and that, within a billion years, this will cause significant deviations in the orbits of the planets, deviations it can estimate. Compared with that amount of mass displaced, the impact of killing all humans everywhere is a trivial one indeed. We certainly wouldn't want it to kill all humans in order to be able to carefully balance out its impact on the orbits of the planets!

There are very "large" impacts to which we are completely indifferent (chaotic weather changes, the above-mentioned change in planetary orbits, the different people being born as a consequence of different people meeting and dating across the world, etc.) and other, smaller, impacts that we care intensely about (the survival of humanity, of people's personal wealth, of certain values and concepts going forward, key technological innovations being made or prevented, etc.). If the AI accomplishes its task with a universal constructor or unleashing hordes of nanobots that gather resources from the world (without disrupting human civilization), it still has to decide whether to allow humans access to the constructors or nanobots after it has finished copying the strawberry - and which humans to allow this access to.

So every decision the AI makes is a tradeoff in terms of its impact on the world. Navigating this requires it to have a good understanding of our values. It will also need to estimate the value of certain situations beyond the human training distribution - if only to avoid these situations. Thus a "superpowered and aligned" AI needs to solve the problem of model splintering, and to establish a reasonable extrapolation of human values.

Model splintering sufficient?

The previous sections argue that learning human values (including model splintering) is necessary for instantiating an aligned AI; thus the "define alignment and then add human values" approach will not work.

Thus, if you give this argument much weight, learning human values is necessary for alignment. I personally feel that it's also (almost) sufficient, in that the skill in navigating model splintering, combined with some basic human value information (as given, for example, by the approach here) is enough to get alignment even at high AI power.

Which path to pursue for alignment

It's important to resolve this argument, as the paths for alignment that the two approaches suggest are different. I'd also like to know if I'm wasting my time on an unnecessary diversion.

Value extrapolation, concept extrapolation, model splintering

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

Post written with Rebecca Gorman.

We've written before that model splintering, as we called it then, was a problem with almost all AI safety approaches.

There's a converse to this: solving the problem would help with almost all AI safety approaches. But so far, we've been posting mainly about value extrapolation. In this post, we'll start looking at how other AI safety approaches could be helped.

Definitions

To clarify, let's make four definitions, distinguishing ideas that we'd previously been grouping together:

Model splintering is when the features and concepts that are valid in one world-model, break down when transitioning to another world-model.

Value splintering (or reward splintering) is when the value function (or reward function, or goal, or preference...) becomes invalid due to model splintering.

Concept extrapolation is extrapolating a feature or concept from one world-model to another.

Value extrapolation is concept extrapolation when the particular concept to extrapolate is a value, a preference, a reward function, an agent's goal, or something of that nature.

Thus concept extrapolation is a solution to model splintering, while value extrapolation is a solution to value splintering specifically.

Examples

Consider for example Turner *et al*'s attainable utility. It has a formal definition, but the reason for that definition is that preserving attainable utility is aimed at restricting the "power" of the agent, or at minimising its "side effects".

And it succeeds, in the typical situation. If you measure the attainable utility of an agent, this will give you an idea of its power, and how many side effects it may be causing. However, when we move to general situations, this breaks down: attainable utility preservation no longer restricts power or reduces side effects. So the concepts of power and side effects have splintered when moving from typical situations to general situations. This is the **model splintering**[1]. If we solve **concept extrapolation** for this, then we could extend the concepts of power restriction or side

effect minimisation, to the general situations. And thus successfully create low impact AIs.

Another example is wireheading. We have a reward signal that corresponds to something we desire in the world; maybe the negative of the CO_2 concentration in the atmosphere. This is measured by, say, a series of CO_2 detectors spread over the Earth's surface.

Typically, the reward signal does correspond to what we want. But if the AI hacks its own reward signal, that correspondence breaks down[2]: **model splintering**. If we can extend the reward properly to new situations, we get **concept extrapolation** - which, since this is a reward function, is **value extrapolation**.

Helping with multiple methods

Hence the concept extrapolation/value extrapolation ideas can help with many different approaches to AI safety, not just the value learning approaches.

---

1. Equivalently, we could say that the concepts remain the same, but it's the correlation between "attainable utility preservation" and "power restriction" is what breaks down. ↩

2. There are multiple ways we can see the concepts breaking down. We can see the concept of "measured CO_2" breaking down. We can see the correlation between CO_2 concentration and the reward breaking down. We can see the correlation between the reward and the *reward signal* breaking down. The reason there are so many ways of seeing the breakdown is because most descriptive labels describe collections of correlated features, rather than fundamental concepts. So the descriptions/features/concepts break down when the correlations do. ↩