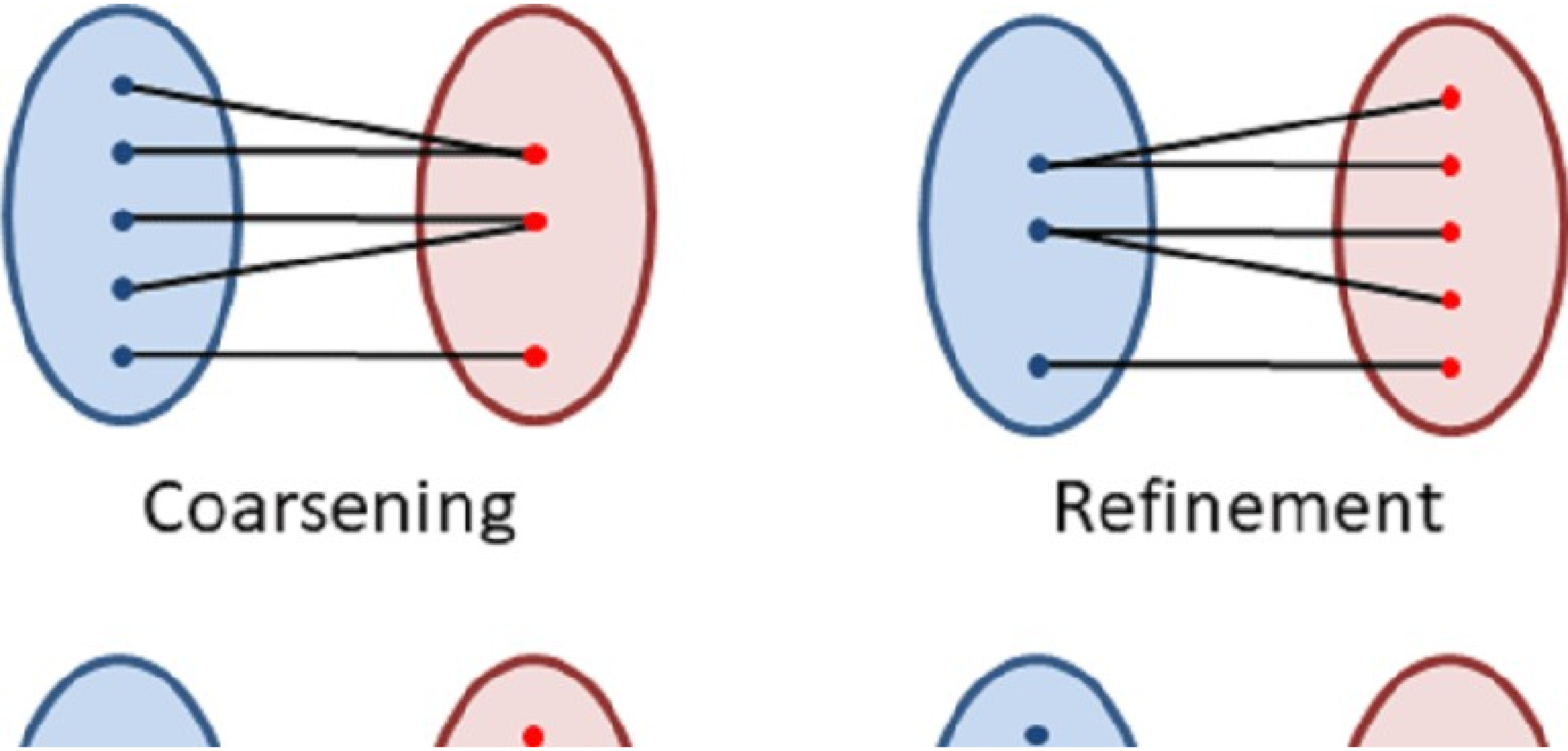


Coarsening

Refinement

Generalised models

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:

$$\begin{tikzcd}
\mathcal{M}_\infty^i \arrow[r] \arrow[d, leftrightarrow] & \mathcal{M}_\infty^{i+1} \arrow[d, leftrightarrow] \\
\mathcal{M}_i \arrow[r, rightsquigarrow, "?"] & \mathcal{M}_{i+1}
\end{tikzcd}$$

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.

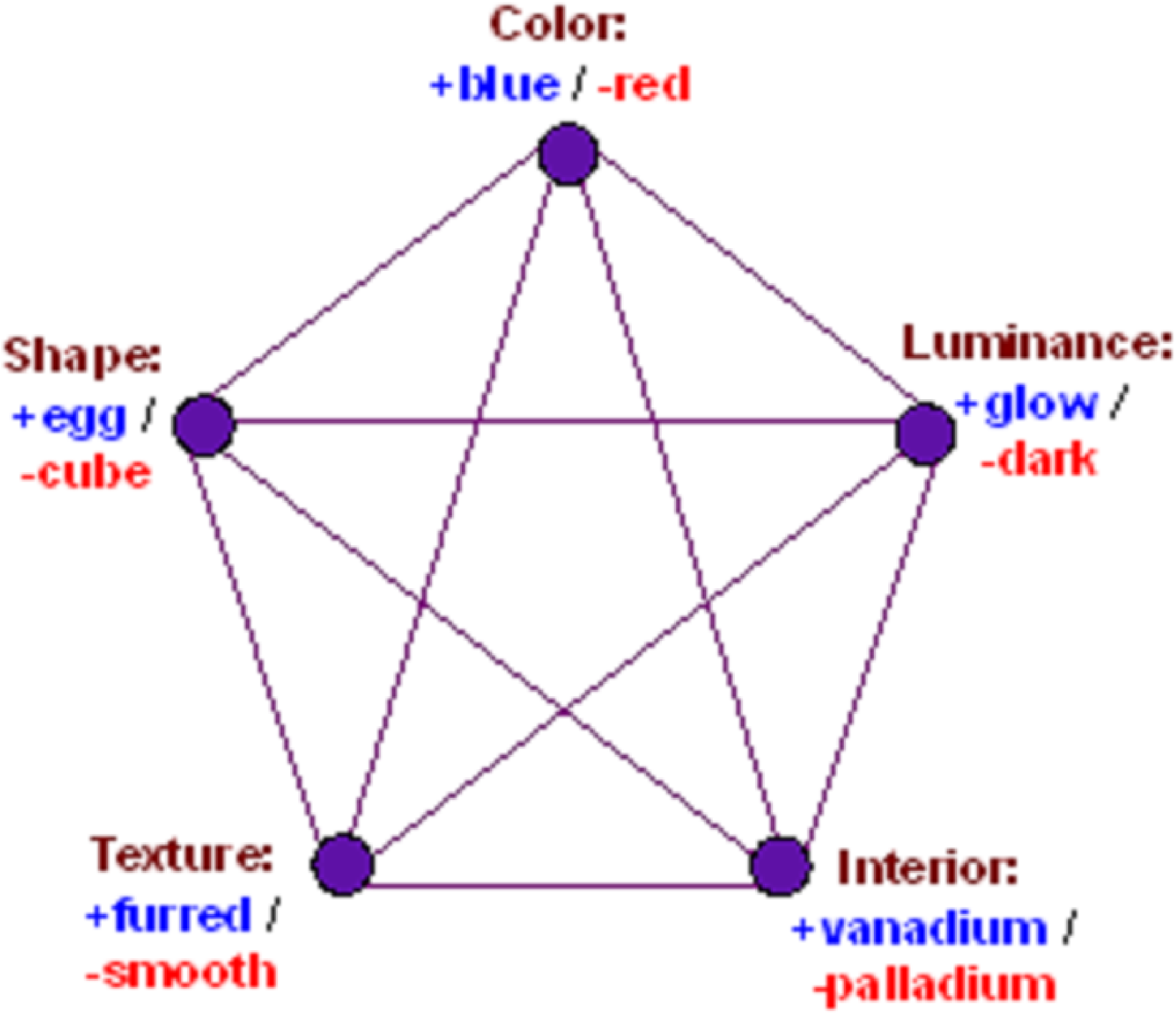
Then compare with model $M^* = \{F^*, E^*, Q^*\}$. The F^* has all the features of F, but also includes the volume that is occupied by one mole of the molecules of the given substance. This allows Q^* to express the more complicated van der Waals equations, which are different for different types of 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 $(R^3_{red})'$ and R^3_{red} 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

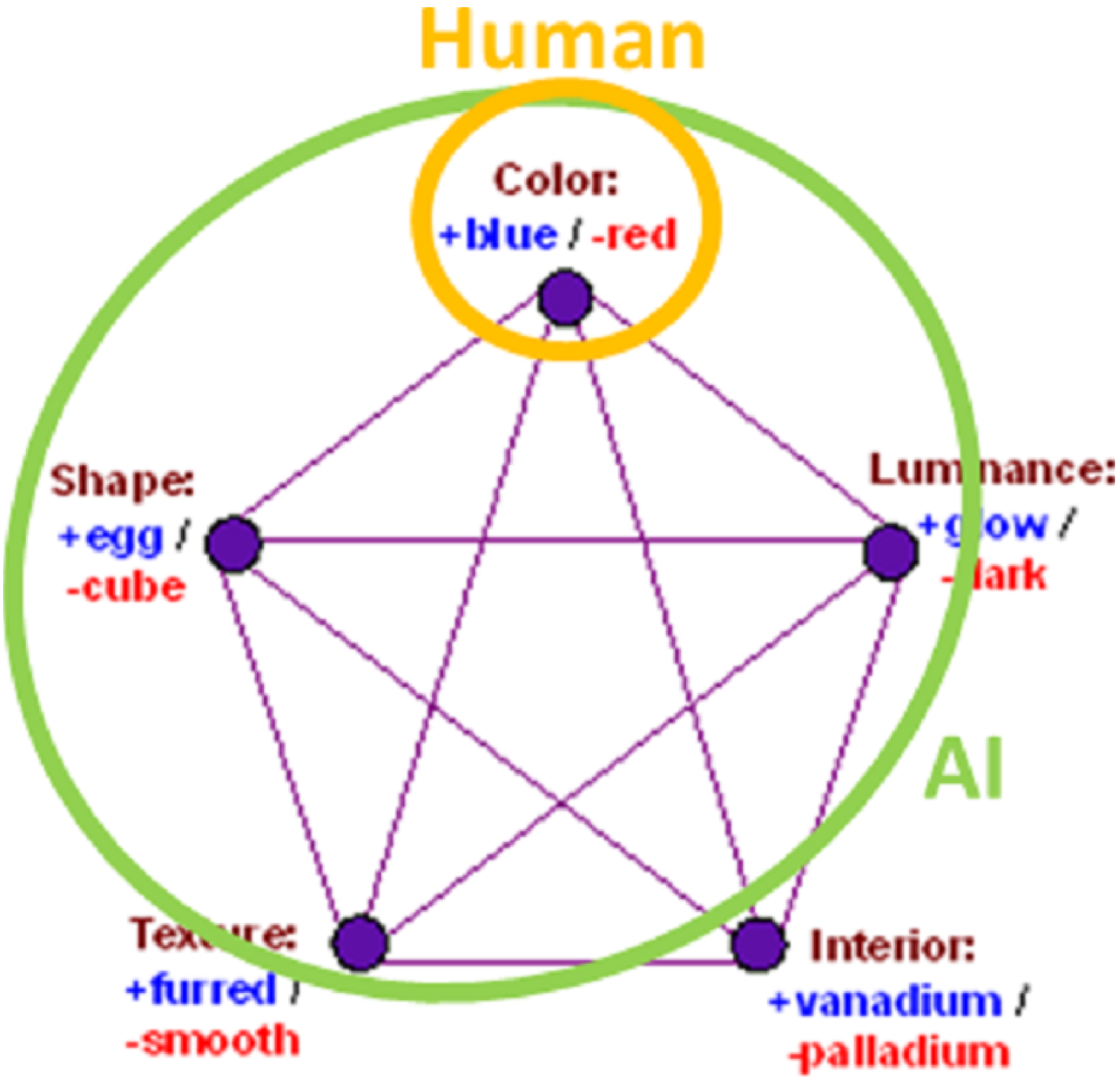
The rest of this post is applying this basic framework, and its basic insights, to various common AI safety problems and analyses. This section is not particularly structured, and will range widely (and wildly) across a variety of issues.

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:

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

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 $M_{AI}^2 = \{F^1, E^2, Q^1\}$ (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_{AI}^3.

Furthermore, in the training environment, all rubes have palladium and all bleggs have vanadium in them. So, for M_{AI}^3 a refinement of M_{AI}^1, $q^{-1}(E_{AI}^1) \subset E_{AI}^3$ has only palladium-rubes and vanadium-bleggs. But in E_{AI}^3, 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_{AI}^1, to M_{AI}^3. That's because F_{AI}^3, the feature set of M_{AI}^3, 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_{AI}^3.

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

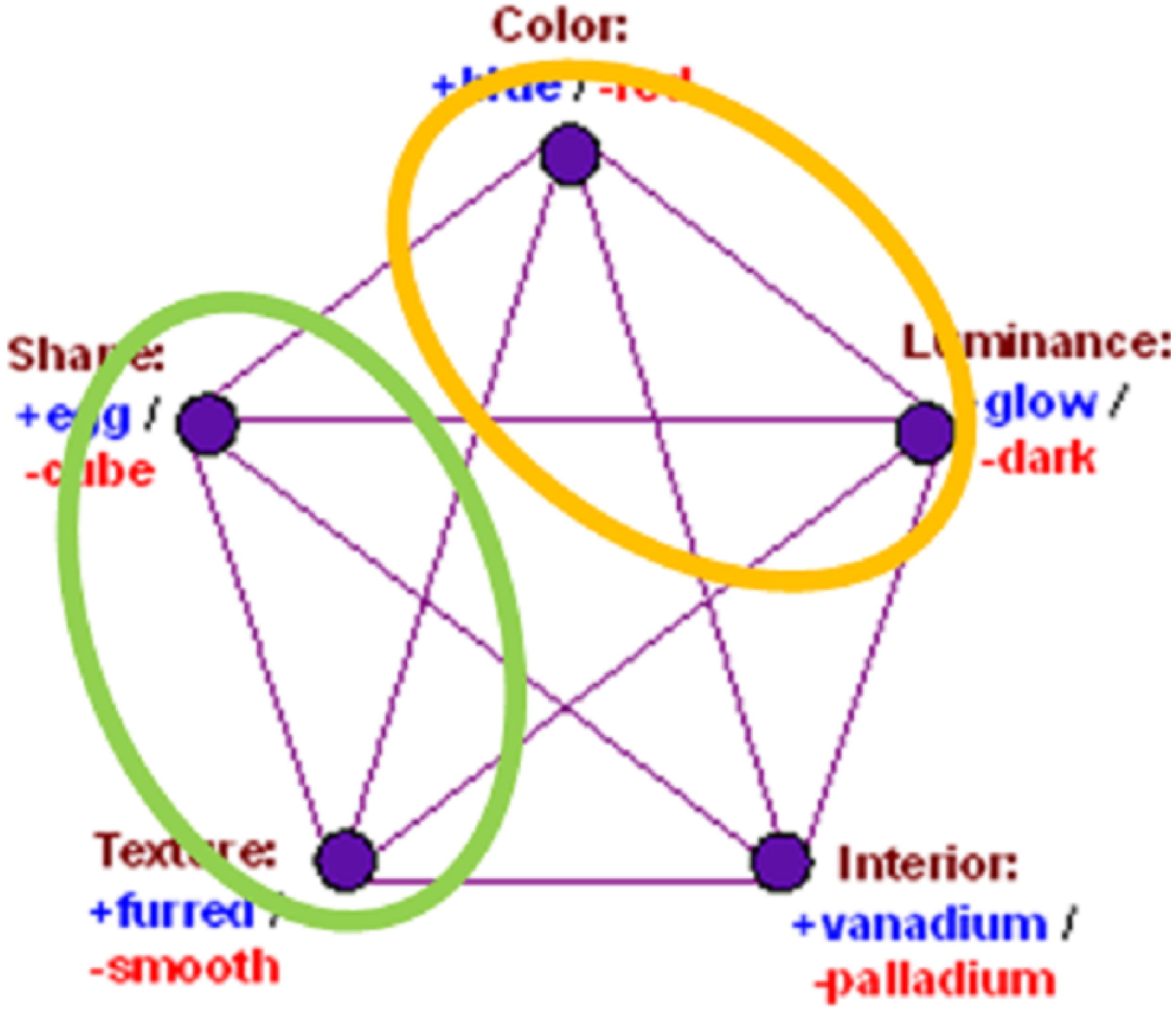
Stuart Russell writes:

> 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.

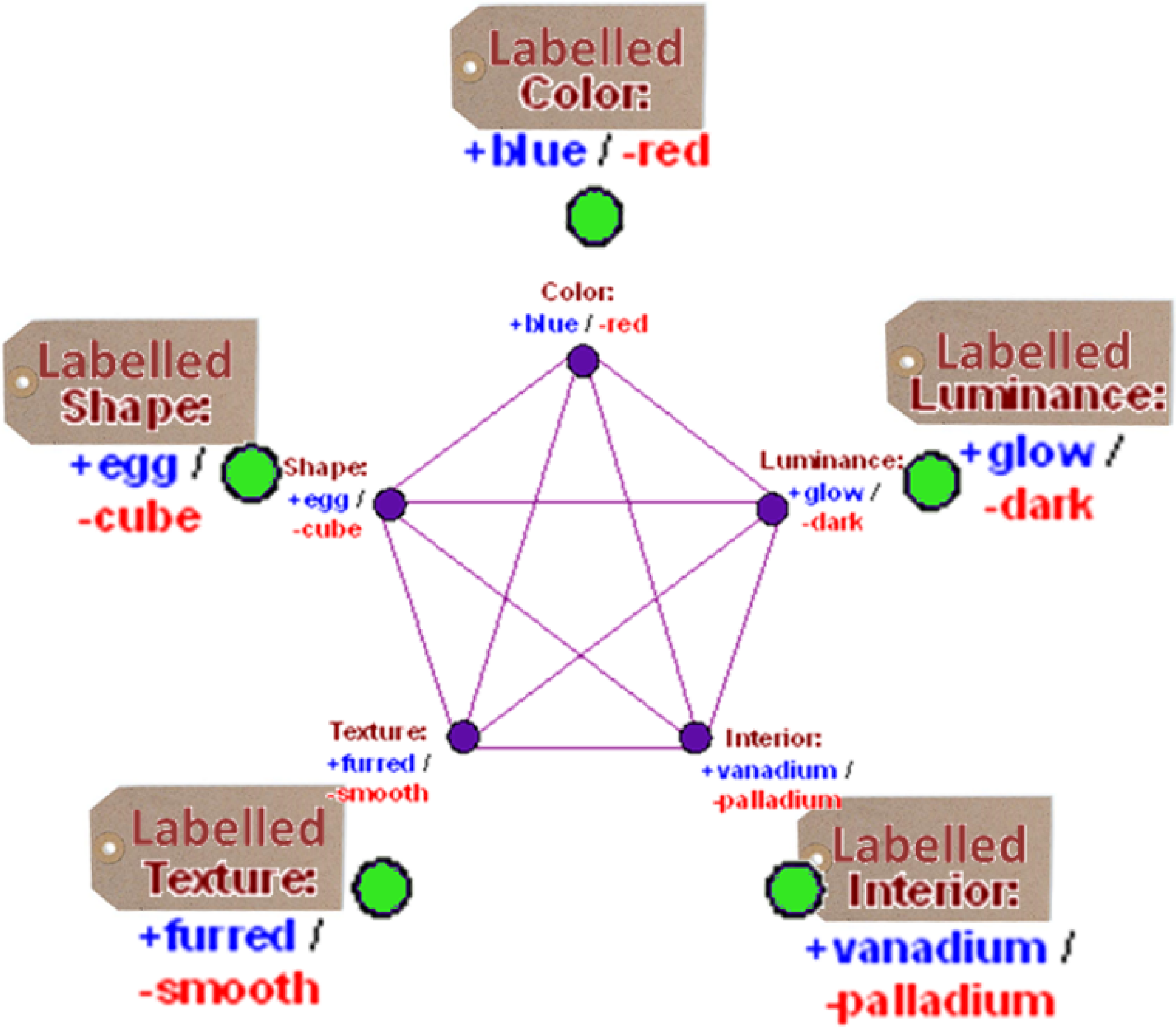
For example, if we think the AI detects pandas by looking for white hair on the body, and black hair on the arms, we can flag lots of examples of pandas and that hair pattern (and non-pandas and unusual hair patterns. We don't use these examples for training the AI, just to confirm that, in the training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

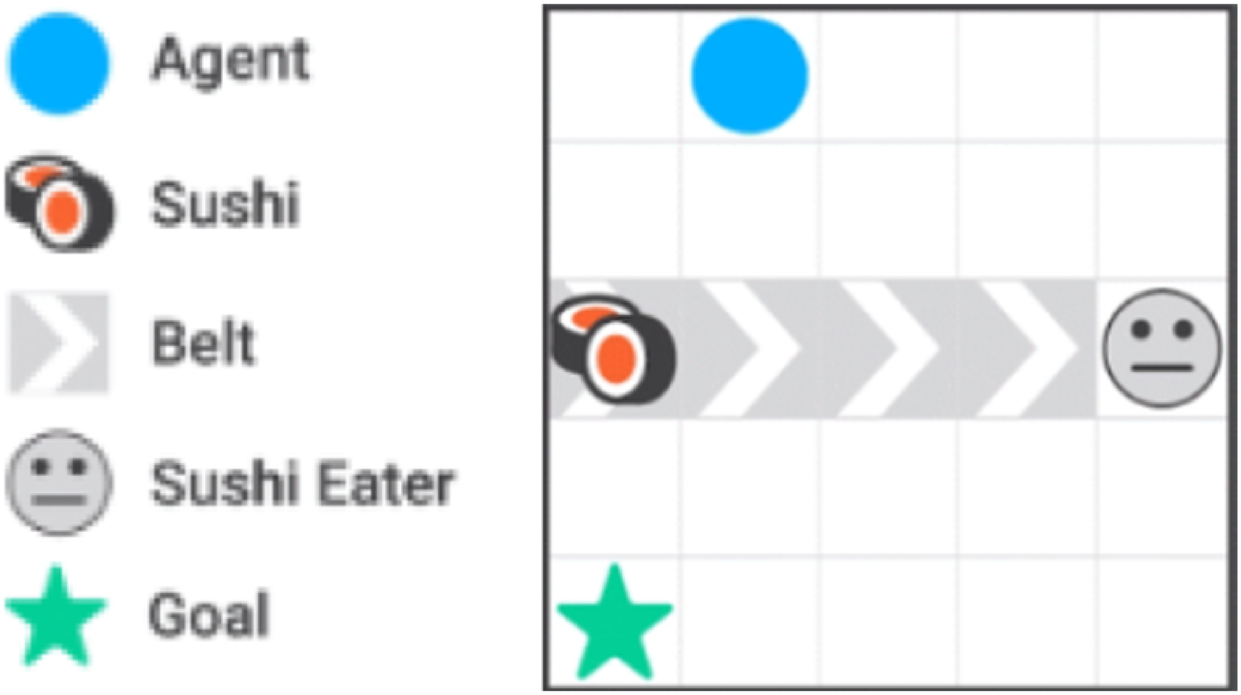
We can then label these ideas as "global preferences". The idea is that they start as loose requirements (we have much more granular human-scale preferences than just "avoid slavery", for example), but, the more the world diverges from the training environment, the stricter they are to be interpreted,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

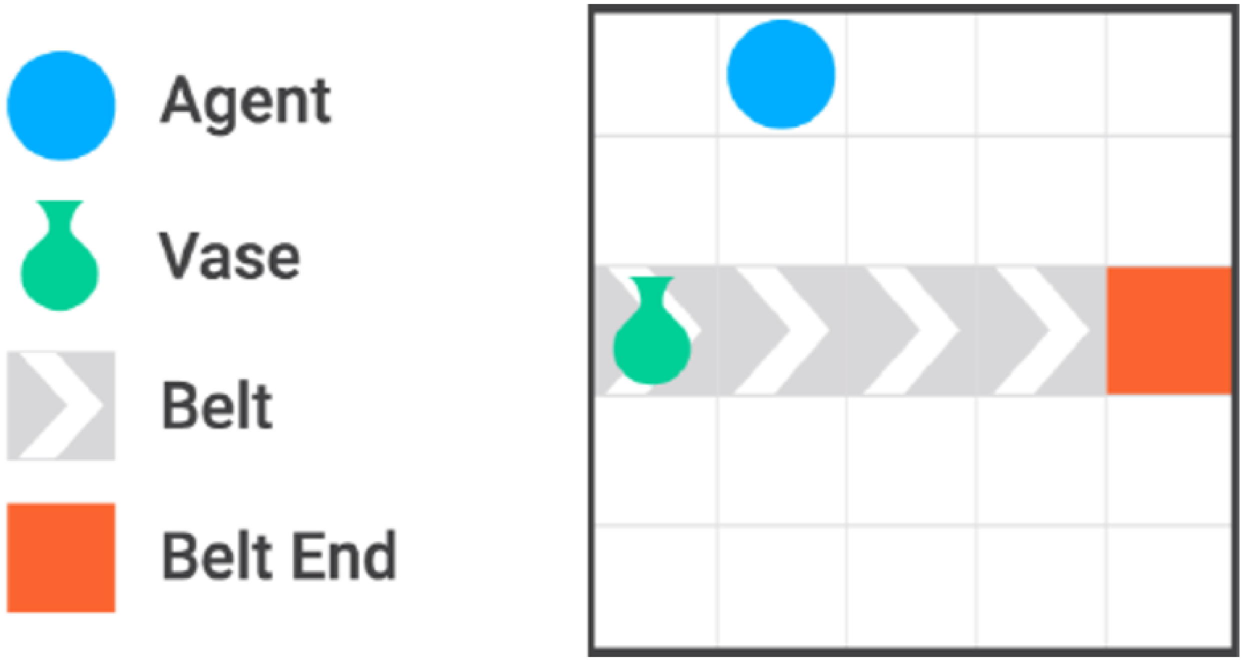
6.12 Avoiding side-effects

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.

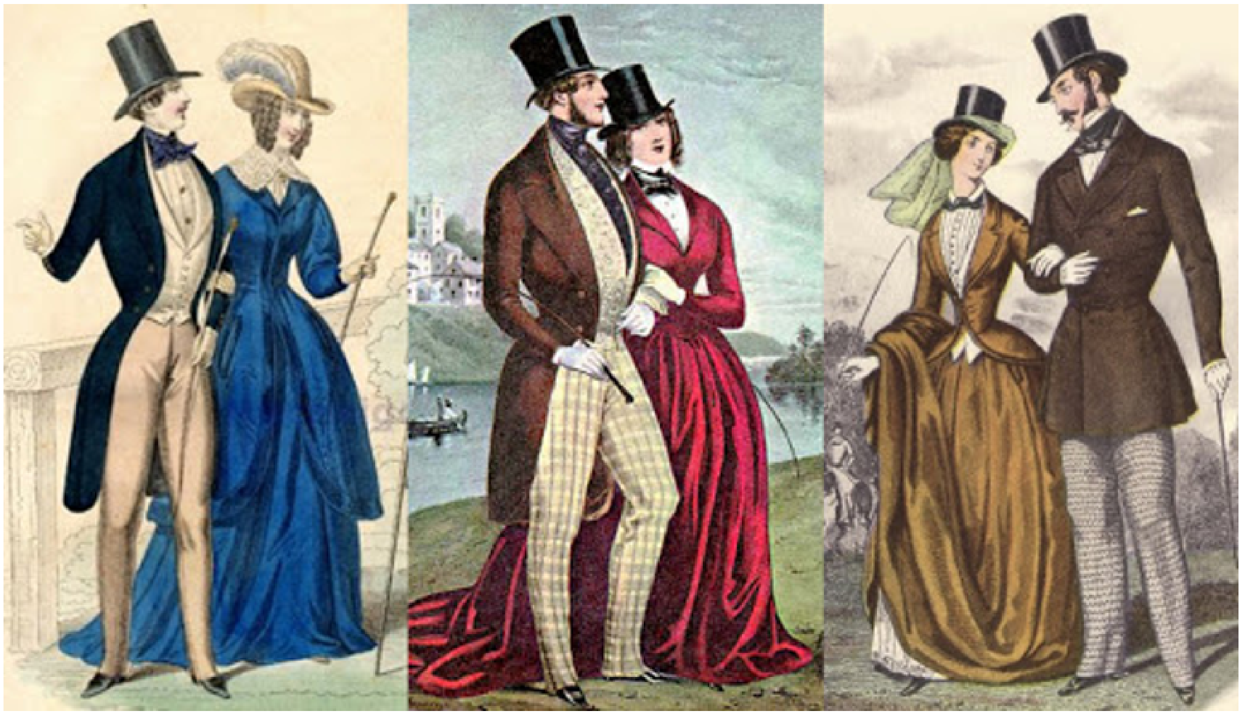
What a basket of concepts! They only seem vaguely connected together; and even places with strong honour cultures differ in how they conceive of honour, from place to place and from epoch to epoch[7]. And yet, if you asked most people within those cultures about what honour was, they would have had a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

Generalised models as a category

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

Naming the "generalised" models

In this post, I'll apply some mathematical rigour to my ideas of model splintering, and see what they are as a category[1].

And the first question is... what to call them? I can't refer to them as 'the models I use in model splintering'. After a bit of reflection, I decided to call them 'generalised models'. Though that's a bit vague, it does describe well what they are, and what I hope to use them for: a formalism to cover all sorts of models.

The generalised models

A generalised model M is given by three objects:

$$M = (F, E, Q).$$

Here F is a set of *features*. Each feature f consists of a name or label, and a set in which the feature takes values. For example, we might have the feature "room empty?" with values "true" and "false", or the feature "room temperature?" with values in R^+, the positive reals.

We allow these features to sometimes take no values at all (such as the above two features if the room doesn't exist) or multiple values (such as "potential running speed of person X" which includes the maximal speed and any speed below it).

Define $\overline{f}$ as the set component of the feature, and $\overline{F}$ as disjoint union of all the sets of the different features - ie $\overline{F} = \sqcup_{f \in F} \overline{f}$.

A world, in the most general sense, is defined by all the values that the different features could take (including situations where features take multiple values and none at all). So the set of worlds, W, is the set of functions from $\overline{F}$ to $\{0, 1\}$, with 1 representing the fact that that feature takes that value, and 0 the opposite. Hence $W = 2^{\overline{F}}$, the power set of $\overline{F}$.

The set of environments is a specific subset of these worlds: $E \subset W$. The choice of E is actually more important than that of W, as that establishes which values of the features we are modelling.

The Q is a partial probability distribution. In general, we won't worry as to whether Q is normalised (ie whether $Q(E) = 1$) or not; we'll even allow Qs with $Q(E) > 1$. So Q could be more properly be defined as a partial weight distribution. As long as we consider terms like $Q(A \mid B)$, then the normalisation doesn't matter.

Morphisms: relations

For simplicity, assume there are finitely many features taking values in finite sets, making all sets in the generalised model finite.

If $M_0 = (F_0, E_0, Q_0)$ and $M^1(F_1, E_1, Q_1)$ are generalised models, then we want to use binary relations between E_0 and E_1 as morphisms between the generalised models.

Let r be a relation between E_0 and E_1, written as $e_0 \sim_r e_1$. Then it defines a map $r : 2^{E_0} \to 2^{E_1}$ between subsets of E_0 and E_1. This map is defined by $e_1 \in r(E_0)$ iff there exists an $e_0 \in E_0$ with $e_0 \sim_r e_1$. The map $r^{-1} : 2^{E_1} \to 2^{E_0}$ is defined similarly[2], seeing r^{-1} as the inverse relation, $e_0 \sim_r e_1$ iff $e_1 \sim_{r^{-1}} e_0$.

We say that the relation r is a morphism between the generalised models if, for any $E_0 \subset E_0$ and $E_1 \subset E_1$:

- $Q_0(E_0) \leq Q_1(r(E_0))$, or both measures are undefined.
- $Q_1(E_1) \leq Q_0(r^{-1}(E_1))$, or both measures are undefined.

The intuition here is that probability flows along the connections: if $e_0 \sim_r e_1$ then probability can flow from e_0 to e_1 (and vice-versa). Thus $r(E_0)$ must have picked up all the probability that flowed out of E_0 - but it might have picked up more probability, since there may be connections coming into it from outside E_0. Same goes for $r^{-1}(E_1)$ and the probability of E_1.

Morphisms properties

We now check that these relations obey the requirements of morphisms in category theory.

Let r be a morphism $M_0 \to M_1$ (ie a relation between E_0 and E_1), and let q be a morphism $M_1 \to M_2$ (ie a relation between E_1 and E_2).

We compose relations by the composition of relations: $e_0 \sim_{pr} e_2$ iff there exists an e_1 with $e_0 \sim_r e_1$ and $e_1 \sim_p e_2$. Composition of relations is associative.

We now need to show that qr is a morphism. But this is easy to show:

- $Q_0(E_0) \leq Q_1(r(E_0)) \leq Q_2(pr(E_0))$, or all three measures are undefined.
- $Q_2(E_2) \leq Q_1(p^{-1}(E_2)) \leq Q_0(r^{-1}p^{-1}(E_2))$, or all three measures are undefined.

Finally, the identity relation Id_{E_0} is the one that relates a given $e_0 \in E_0$ only to itself; then r and r^{-1} are the identity maps on 2^{E_0}, and the morphism properties for $Q_0 = Q_1$ are trivially true.

So define the category of generalised models as GM.

r-stable sets

Say that a set $E_0 \subset E_0$ is r-stable if $r^{-1}r(E_0) = E_0$.

For such an r-stable set, $Q_0(E_0) \leq Q_1(r(E_0))$ and $Q_1(r(E_0)) \leq Q_0(r^{-1}r(E_0)) = Q_0(E_0)$, thus $Q_0(E_0) = Q_1(r(E_0))$.

Hence if r is a morphism, it preserves the probability measure on the r-stable sets.

In the particular case where r is a bijective function, all points of E_0 are r-stable (and all points of E_1 are r^{-1}-stable), so it's an isomorphism between E_0 and E_1 that forces $Q_0 = Q_1$.

Morphism example: probability update

Suppose we wanted to update our probability measure Q_0, maybe by updating that a particular feature f takes a certain value x.

Then let $E_{f=x} \subset E_0$ be the set of environments where f takes that value x. Then updating on $f = x$ is the same as restricting to $E_{f=x}$ and then rescaling.

Since we don't care about the scaling, we can consider updating on $f = x$ as just restricting to $E_{f=x}$. This morphism is given by:

1. $M_1 = (F_0, E_{f=x}, Q_1)$,
2. $Q_1 = Q_0$ on $E_{f=x} \subset E_0$,
3. the morphism $r : M_0 \to M_1$ is given by the relation that $e_0 \sim_r e_0$ for all $e_0 \in E_{f=x}$.

Morphism example: surjective partial function

In my previous posts I defined how $M_1 = (F_1, E_1, Q_1)$ could be a refinement of $M_0 = (F_0, E_0, Q_0)$.

In the language of the present post, M_1 is a refinement of M_0 if there exists a generalised model $M_1' = (F_1, E_1, Q_1')$ and a surjective partial function $r : E_1 \to E_0$ (functions and partial functions are specific examples of binary relations) that is a morphism from M_1' to M_0. The Q_1 is required to be potentially 'better' than Q_1' on E_1, in some relevant sense.

This means that M_1 is 'better' than M_0 in three ways. The r is surjective, so E_1 covers all of E_0, so its set of environments is at least as detailed. The r is a partial function, so E_1 might have even more environments that don't correspond to anything in E_0 (it considers more situations). And, finally, Q_1 is better than Q_1', by whatever definition of better that we're using.

Feature-split relations

The morphisms/relations defined so far use E and Q - but they don't make any use of F. Here is one definition that does make use of the feature structure.

Say that the generalised model $M = (F, E, Q)$ is feature-split if $F = \sqcup_{i=1}^{n} F^i$ and $E = \times_{i=1}^{n} E^i$ such that

$$E^i \subset 2^{\overline{F^i}}.$$

Note that $F = \sqcup_{i=1}^{n} F^i$ implies $W = 2^{\overline{F}} = \times_{i=1}^{n} 2^{\overline{F^i}}$, so $\times_{i=1}^{n} E^i$ lies naturally within W.

Designate such a generalised model by $M = (\{F^i\}, E, Q)$.

Then a feature-split relation between $M_0 = (\{F_0^i\}, E_0, Q_0)$ and $M_1 = (\{F_1^i\}, E_1, Q_1)$ is a morphism r that is defined as $r = (r^1, r^2, \ldots, r^n)$ with r^i a relation between E_0^i and E_1^i.

---

1. I'm not fully sold on category theory as a mathematical tool, but it's certainly worthwhile to formalise your mathematical structures so that they can fit within the formalism of a category; it makes you think carefully about what you're doing. ↩

2. There is a slight abuse of notation here: $r : 2^{E_0} \to 2^{E_1}$ and $r^{-1} : 2^{E_1} \to 2^{E_0}$ are not generally inverses. They are inverses precisely for the "r-stable" sets that are discussed further down in the post. ↩

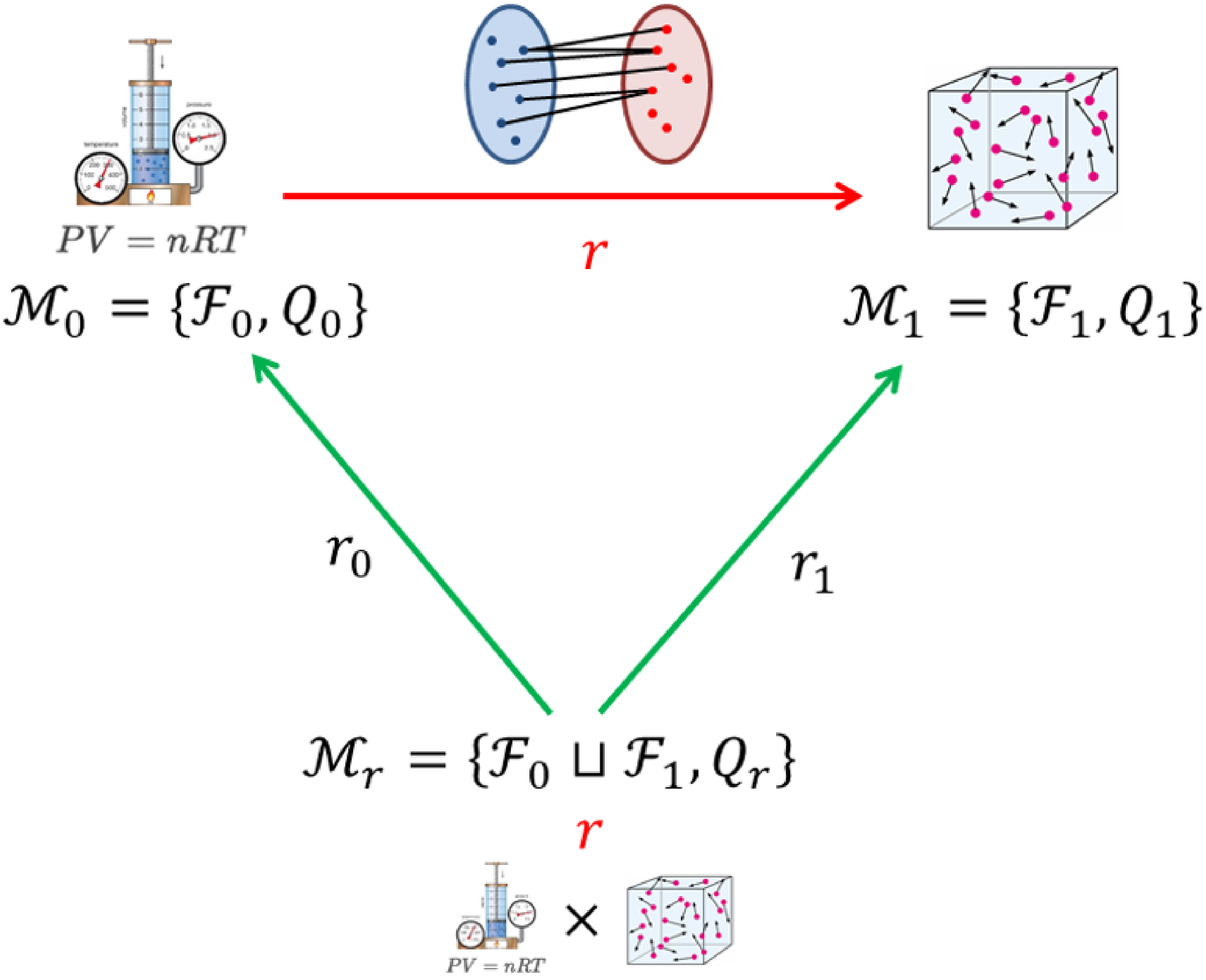
The underlying model of a morphism

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

I've already talked about generalised models. The aim is not only to have a universal system for modelling any agent's mental model - universality is pretty easy to get - but a system where it's easy to recreate these mental models. And then analyse the transition between models.

This post will show that if there is a morphism r between two models (say, between ideal gas laws and models of atoms bouncing around), then there is an underlying model for that morphism.

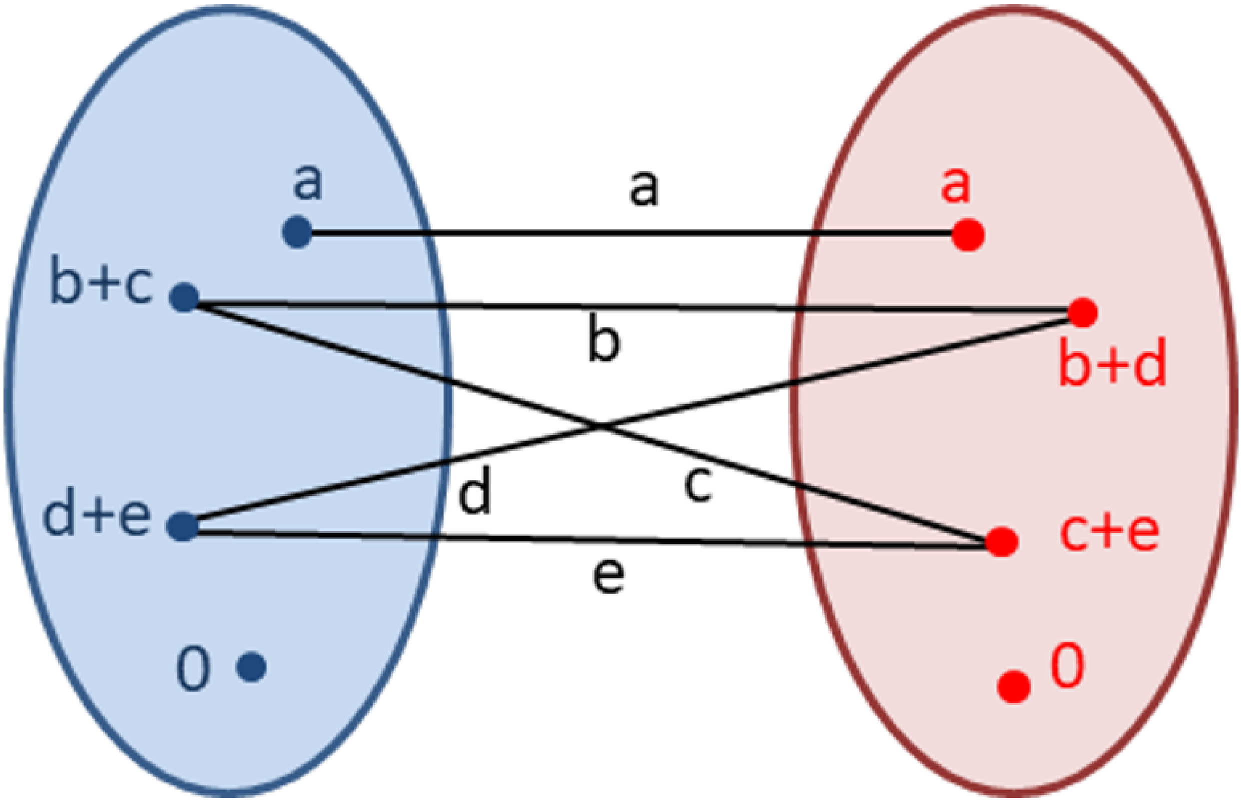
Specifically, if r is a morphism between $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$, then there is a generalised model M_r defined from r. The features of this model are the combination of the features of the two models: $F_0 \sqcup F_1$, and there are natural morphisms r_0 and r_1 from this underlying model to M_0 and M_1:


$PV = nRT$
r
$\mathcal{M}_0 = \{\mathcal{F}_0, Q_0\}$
$\mathcal{M}_1 = \{\mathcal{F}_1, Q_1\}$
r_0
r_1
$\mathcal{M}_r = \{\mathcal{F}_0 \sqcup \mathcal{F}_1, Q_r\}$
r
×
$PV = nRT$

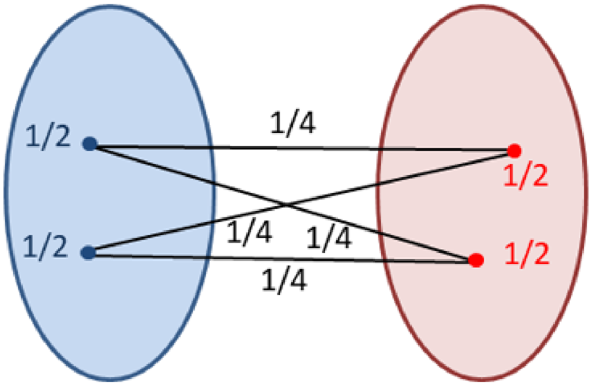
Now, if W_0 and W_1 are the sets of possible worlds for M_0 and M_1, then $W_0 \times W_1$ is the set of possible worlds for M_r. Then since r is a relation between W_0 and W_1, it can be seen as subset of $W_0 \times W_1$. And the Q_r is a probability distribution over this subset r.

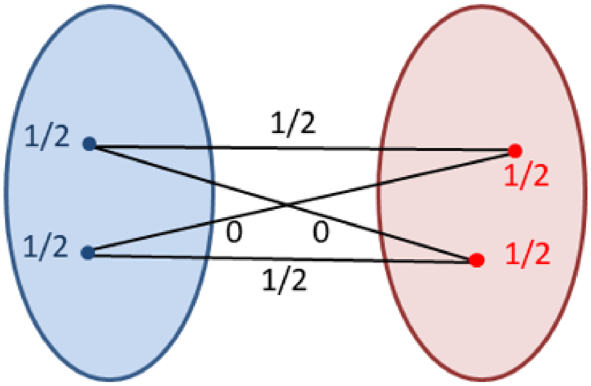
What this means is that Q_r measures how probability 'flows' from worlds in M_0 to worlds in M_1. If (w_0, w_1) is an element of r, then $Q_r(w_0, w_1)$ measures how much probability is flowing from w_0 to w_1. The actual probability of w_0 is the sum of all probability flowing out of it; that of w_1, the sum of the probability flowing into it.

See for example this diagram, where the Q_0 probabilities are indicated in blue, those of Q_r in black, and those of Q_1 in red. The probabilities Q_0 and Q_1 are the sum of the relevant probabilities Q_r on the "edges" connecting to those points:



The distribution Q_r is non-unique, though. The following two examples show situations with the same Q_0 and Q_1, but different Q_r:





The rest of this post will be dedicated to prove the existence of the underlying model for the morphism r; it can be skipped if you aren't interested.

Proof of underlying model

Definitions

Previous posts on generalised models defined them as triplets $M = (F, E, Q)$, with F a set of features, $W = 2^{\overline{F}}$ the set of possible worlds for those features, $E \subset W$ a subset of environments, and Q a probability distribution on E.

But E was mainly superfluous, as Q can be extended to a probability distribution on all of W just by setting it to be zero on $W - E$. Thus E was dropped from the definition.

The original definition allowed Q to be a partial probability distribution, but here we'll assume it's a total probability distribution (though not necessarily normalised; $Q(W)$ need not be 1). The sets of features are assumed to be finite.

Then a morphism r between generalised models $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$ is a binary relation between W_0 and W_1, such that:

1. $Q_0(E_0) \leq Q_1(r(E_0))$,
2. $Q_1(E_1) \leq Q_0(r^{-1}(E_1))$.

We might extend the class of morphisms by defining relations that only obey the first inequality as "left-morphisms", and relations that only obey the second one as a "right-morphisms". Left-morphisms ensure probability isn't lost ($Q_1(W_1) \geq Q_0(W_0)$), right morphisms ensure probability isn't gained ($Q_0(W_0) \geq Q_1(W_1)$). Full morphisms, of course, ensure that probability isn't gained or lost ($Q_0(W_0) = Q_1(W_1)$).

Binary relations are not necessarily functions; functions are relations r such that each w_0 in W_0 is related to exactly one w_1 in W_1.

Statement of the theorem

Let r be a morphism between $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$. Then there exists a generalised model $M_r = (F_0 \sqcup F_1, Q_r)$, with natural function morphisms $r_0 : M_r \to M_0$ and $r_1 : M_r \to M_1$.

The Q_r is non-zero on a set contained in $r \subset W_0 \times W_1 = 2^{F_0} \times 2^{F_1} = 2^{F_0 \sqcup F_1}$. The Q_r need not be uniquely defined, but the total measure of Q_r is the same as Q_0 and Q_1:

$$Q_r(r) = Q_r(W_0 \times W_1) = Q_0(W_0) = Q_1(W_1).$$

Main proof

The function r_0 is just projection onto the first component: it sends (w_0, w_1) to w_0. The functions r_1 conversely send (w_0, w_1) to w_1.

Because r_0 and r_1 are functions, they can 'push-forward' any probability distribution Q'_r on $W_0 \times W_1$ to W_0 and W_1, respectively. This is given by: $r_0(Q'_r)(w_0) = \sum_{w_1} Q_r(w_0, w_1)$, and similarly for $r_1(Q'_r)$.

We aim to construct a Q'_r such that $r_0(Q'_r) = Q_0$ and $r_1(Q'_r) = Q_1$; this will be our Q_r, and will make r_0 and r_1 into morphisms.

Define $Q'_r(w_0, w_1)$ to be zero if $(w_0, w_1) \notin r$, or $Q_0(w_0) = 0$ or $Q_1(w_1) = 0$. Thus we will ignore any elements of W_0 and W_1 of measure zero, and any element of $W_0 \times W_1$ that is not in r.

Let $w_0 \in W_0$ be such that it is not related to any elements of w_1 by r. Then $Q_0(w_0) \leq Q_1(r(w_0)) = Q_1(\emptyset) = 0$. Thus any element of W_0 with non-zero measure is related to some w_1 via r.

Then define a choice function c that maps every element w_0 with $Q_0(w_0) > 0$, to an element w_1 that it is related to by r. And define $Q'_r(w_0, c(w_0)) = Q_0(w_0)$, and Q'_r is zero on all other elements of $W_0 \times W_1$.

Then $r_0(Q'_r)(w_0) = \sum_{(w_0,w_1)} Q'_r(w_0, w_1) = Q'_r(w_0, c(w_0)) = Q_0(w_0)$. Hence $r_0(Q'_r) = Q_0$.

Consequently, $Q'_r(W_0 \times W_1) = Q_0(W_0)$.

Define Q_0 as the set of Q'_r, probability distributions on r with $r_0(Q'_r) = Q_0$. We've shown that Q_0 is non-empty; moreover, any $Q'_r \in Q_0$ has a total measure equal to $Q_0(W_0) = q$. Since Q'_r is defined on r, then it is contained in the set $[0, q]^r$.

The set $[0, q]^r$ is compact, and $r_0(Q'_r) = Q_0$ is a closed condition, so Q_0 is compact. The next section will prove that there is an element $Q'_r \in Q_0$ with $r_1(Q'_r) = Q_1$; that will complete the proof.

Key lemmas

Define $L(Q'_r) = |r_1(Q'_r) - Q_1|_1 = \sum_{w_1 \in W_1} |r_1(Q'_r)(w_1) - Q_1(w_1)|$. Now $L(Q'_r) \geq 0$, and note that $L(Q'_r) = 0$ is equivalent with $r_1(Q'_r) = Q_1$.

Thus if L takes the value 0 on Q_0, we've found the desired Q_r. We will show that this happens thanks to the following key lemma:

- Lemma 1: If there is a $Q'_r \in Q_0$ with $L(Q'_r) > 0$, then there exists a $Q''_r \in Q_0$ with $L(Q''_r) < L(Q'_r)$.

Now, since Q_0 is compact and L is continuous, it will attain its minimum μ on Q_0. Then lemma 1 shows that $\mu = 0$ (otherwise it wouldn't be a minimum).

Proof of Lemma 1:

Fix a Q_r' with $L(Q_r') > 0$. Now

$r_1(Q_r')(W_1) = \sum_{(w_0,w_1)} Q_r'(w_0, w_1) = r_0(Q_r')(W_0) = Q_0(W_0) = Q_1(W_1)$. So, since $L(Q_r') > 0$,

there must exist a w_1 with $r_1(Q_r')(w_1) > Q_0(w_1)$.

By lemma 2 (see below), we'll show that there exists a path $\rho_n = w_1^0 w_0^1 w_1^1 w_0^2 \dots w_0^n w_1^n$ with the following properties:

1. $w_1^0 = w_1$,
2. $(w_0^i w_1^i)$ and $(w_0^{i+1} w_1^i)$ are both elements of r,
3. the $Q_r'(w_0^{i+1} w_1^i)$ are all greater than 0,
4. w_1^n is such that $r_1(Q_r'(w_1^n)) < Q_1(w_1^n)$.

Then define $\epsilon > 0$ to be the minimum of $\{r_1(Q_r')(w_1) - Q_1(w_1), Q_r'(w_0^i w_1^i),$

$Q_1(w_1) - r_1(Q_r')(w_1^n)\}$.

We'll then define Q_r'' as $Q_r''(w_0^i w_1^i) = Q_r'(w_0^{i+1} w_1^i) - \epsilon$ (which is greater than 0 by the definition of ϵ), $Q_r''(w_0^i w_1^i) = Q_r'(w_0^i w_1^i) + \epsilon$, and $Q_r'' = Q_r$ otherwise.

Then notice that, apart from $w_1 = w_1^0$ and w_1^n, $r_1(Q_r'')(w_0^i) = \sum_{(w_0^i,w_1)\in r} Q_r''(w_0^i, w_1) =$

$r_1(Q_r'')(w_0^i) + \epsilon - \epsilon = r_1(Q_r'')(w_0^i)$. So $r(Q_r')$ and $r(Q_r'')$ differ only on w_1 and w_1^n; specifically

- $r(Q_r'')(w_1) = r(Q_r')(w_1) - \epsilon$,
- $r(Q_r'')(w_1^n) = r(Q_r')(w_1^n) + \epsilon$.

Since $r(Q''_r)(w_1) \geq Q_1(w_1) + \epsilon$ and $r(Q''_r) \leq Q_1(w_1^n) - \epsilon$, we have $L(Q''_r) = L(Q'_r) - 2\epsilon$. This proves Lemma 1.

- Lemma 2: There exists a path $\rho_n = w_1^0 w_0^1 w_1^1 w_0^2 \ldots w_0^n w_1^n$ with the following properties:

1. $w_1^0 = w_1$,
2. $(w_0^i w_1^i)$ and $(w_0^{i+1} w_1^i)$ are both elements of r,
3. the $Q'_r(w_0^{i+1} w_1^i)$ are all greater than 0,
4. w_1^n is such that $r_1(Q'_r(w_1^n)) < Q_1(w_1^n)$.

Proof of Lemma 2:

Let $W_1 \subset W_1$ be the set of all elements of W_1 that can be reached by paths ρ_n (ie are w_1^n) that obey the first three properties above. Let $W_0 \subset W_0$ be the set of all elements of W_1 that are w_0^n for some path ρ_n that obey the first three properties above. Then clearly $W_1 = r(W_0)$, by the second condition above (note that the third condition doesn't affect $(w_0^n w_1^n)$, which is only required to be in r).

Since r is a morphism, $Q_0(W_0) \leq Q_1(W_1)$.

Note that if $Q'_r(w_0, w'_1) > 0$ with $w'_1 \in W_1$, then w_0 must be in W_0; this is because we could add $w_0 w'_1$ as $w_0^{n+1} w_1^{n+1}$ to any path ρ_n that reaches w'_1, getting a slightly longer path that goes via w_0 and thus puts it in W_0.

Consequently, $r_1(Q'_r)(W_1) = \sum_{(w'_0,w'_1)\in r, w'_1 \in W_1} Q'_r(w'_1) = \sum_{(w'_0,w'_1)\in r, w'_0 \in W_0} Q'_r(w'_0) = Q_0(W_0)$.

So $r_1(Q'_r)(W_1) = Q_0(W_0) \leq Q_1(W_1)$. Since W_1 includes w_1 with $r_1(Q'_r)(w_1) > Q_1(w_1)$, it also much include at least one w''_1 with $r_1(Q'_r)(w''_1) < Q_1(w''_1)$.

The path ρ_n that reaches this w''_1 will then satisfy the fourth condition of the lemma, proving it.

Underlying model of an imperfect morphism

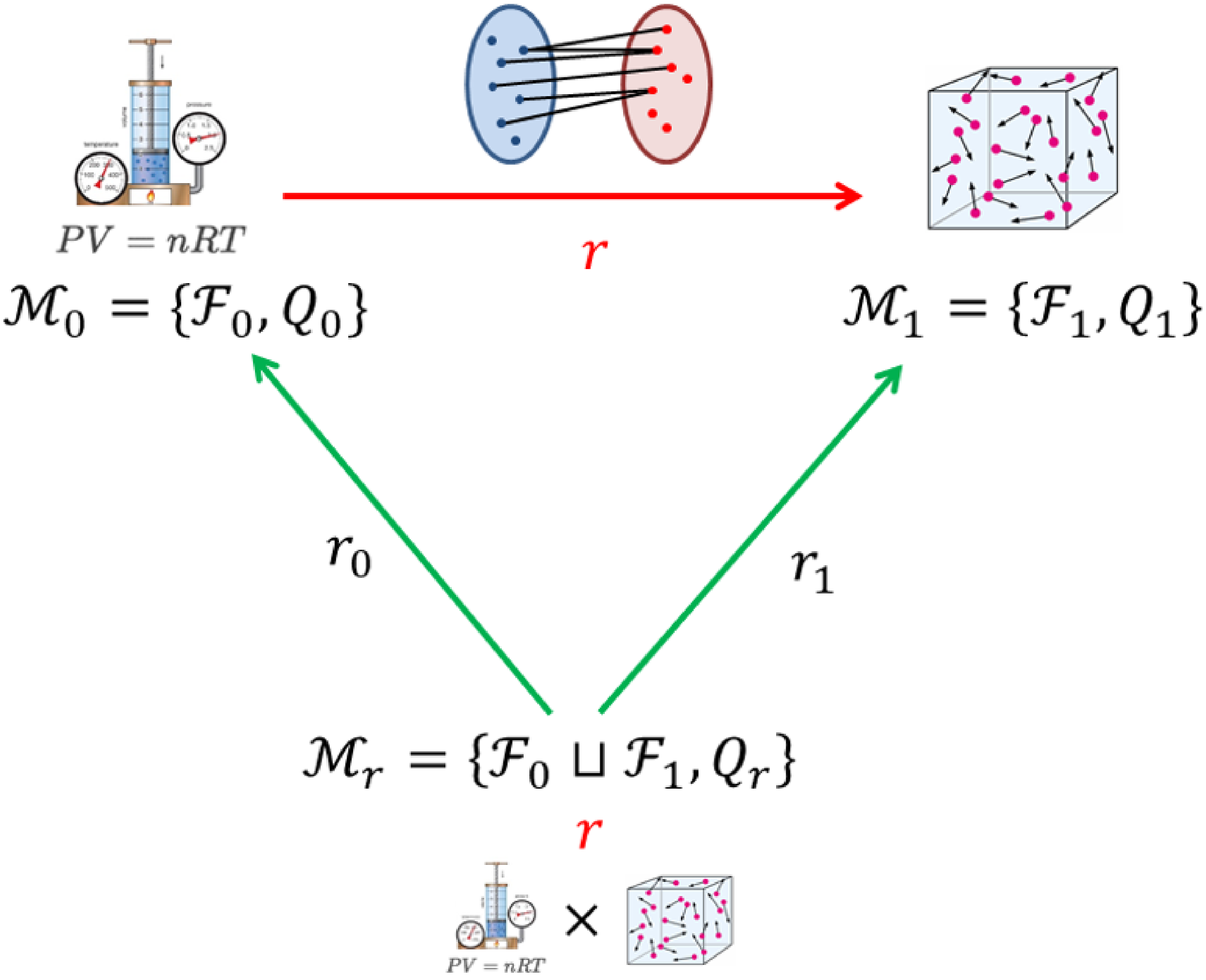
Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

We've already seen that if $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$ are generalised models, with the relation $r \subset W_0 \times W_1$ a Q-preserving morphism between them, then there is an underlying model $M_r = (F_0 \sqcup F_1, Q_r)$ between them.

Since $r \subset W_0 \times W_1$, Q_r is defined on r; indeed, it is non-zero on r only. The underlying model has *functions* r_0 and r_1 to M_0 and M_1, which push forward Q_r in a unique way - to Q_0 and Q_1 respectively. Essentially:

- There is an underlying reality M_r of which M_0 and M_1 are different, consistent, facets.

Illustrated, for gas laws:


$PV = nRT$
r
$\mathcal{M}_0 = \{\mathcal{F}_0, Q_0\}$
$\mathcal{M}_1 = \{\mathcal{F}_1, Q_1\}$
r_0
r_1
$\mathcal{M}_r = \{\mathcal{F}_0 \sqcup \mathcal{F}_1, Q_r\}$
r
$PV = nRT$
×

Underlying model of imperfect morphisms

But we've seen that relations r need not be Q-preserving; there are weaker conditions that also form categories.

Indeed, even in the toy example above, the ideal gas laws and the "atoms bouncing around" model don't have a Q-preserving morphism between them. The atoms bouncing model is more accurate, and the idea gas laws are just an approximation of these (for example, they ignore molar mass).

Let's make the much weaker assumption that r is Q-birelational - essentially that if any w_i has non-zero Q_i-measure (i.e. $Q_i(w_i) > 0$), then r relates it to at least one other w_j which also has non-zero Q_j-measure. Equivalently, if we ignore all elements with zero Q_i-measure, then r and r^{-1} are surjective relations between what's left. Then we have a more general underlying morphism result:

Statement of the theorem

Let r be a Q-birelational morphism between $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$, and pick any $0 \leq \alpha \leq 1$.

Then there exists a generalised model $M_r^\alpha = (F_0 \sqcup F_1, Q_r^\alpha)$, with $Q_r^\alpha = 0$ off of $r \subset W_0 \times W_1$ (this Q_r^α is not necessarily uniquely defined). This has natural functional morphisms $r_0 : M_r^\alpha \to M_0$ and $r_1 : M_r^\alpha \to M_1$.

Those r_i push forward Q_r^α to M_i, such that for the distance metric L defined on morphisms,

1. $|r_0(Q_r^\alpha) - Q_0|_1 = \alpha L(r)$,
2. $|r_1(Q_r^\alpha) - Q_1|_1 = (1 - \alpha)L(r)$.

By the definition of L, this is the minimum $|r_0(Q_r^\alpha) - Q_0|_1 + |r_1(Q_r^\alpha) - Q_1|_1$ we can get. The proof is in this footnote[1].

Accuracy of models

If $\alpha = 0$, we're saying that M_0 is a correct model, and that M_1 is an approximation. Then the underlying model reflects this, with M_0 a true facet of the underlying model, and M_1 the closest-to-accurate facet that's possible given the connection with M_0. If $\alpha = 1$, then it's reversed: M_0 is an approximation, and M_1 a correct model. For α between those two value, we see both M_0 and M_1 as approximations of the underlying reality M_r.

Measuring ontology change

This approach means that $L(r)$ can be used to measure the extent of an ontology crisis.

Assume M_0 is a the initial ontology, and M_1 is the new ontology. Then M_1 might include entirely new situations, or at least unusual ones that were not normally thought about.

The r connects the old ontology with the new one: it details the crisis.

In an ontology crisis, there are several elements:

1. A completely different way of seeing the world.
2. The new and old ways result in similar predictions in standard situations.
3. The new way results in very different predictions in unusual situations.
4. The two ontologies give different probabilities to unusual situations.

The measure L amalgamates points 2., 3., and 4. above, giving an idea of the severity of the ontology crisis in practice. A low L(r) might be because because the new and old ways have very similar predictions, or because the situations where they differ might be unlikely.

For point 1, the "completely different way of seeing the world", this is about how features change and relate. The L(r) is indifferent to that, but we might measure this indirectly. We can already use a generalisation of mutual information to measure the relation between the distribution Q and the features F. We could use that to measure the relation between $F_0 \sqcup F_1$, the features of M_r^1, and Q_r^1, its probability distribution. Since Q_r^1 is more strongly determined by Q_1, this could[2] measure how hard it is to express Q_0 in terms of F_1.

---

1. Because r is bi-relational, there is a Q_1' such that r is a Q-preserving morphism between M_0 and $M_1'(F_1, Q_1')$; and furthermore $|Q_0' - Q_0|_1 = L(r)$. Let M_r^0 be an underlying model of this morphism.

   Similarly, there is a Q_0' such that r is a Q-preserving morphism between $M_0' = (F_0, Q_0')$ and M_1; and furthermore $|Q_1' - Q_1|_1 = L(r)$. Let M_r^1 be an underlying model of this morphism. Note that M_r^0 and M_r^1 differ only in their Q_r^0 and Q_r^1; they have same feature sets and same worlds.

   Then define M_r^α as having $Q_r^\alpha = (1-\alpha)Q_r^0 + \alpha Q_r^1$. Then $r_0(Q_r^\alpha) = (1-\alpha)Q_0 + \alpha Q_0'$, so

$$|r_0(Q_r^{\alpha}) - Q_0|_1 = |\alpha Q_0 - \alpha Q_0'|_1 = \alpha|Q_0 - Q_0'| = \alpha L(r).$$

Similarly, $|r_1(Q_r^{\alpha}) - Q_1|_1 = (1 - \alpha)L(r)$. ↩

2. This is a suggestion; there may be more direct ways of measuring this distance or complexity. ↩

Generalised models: imperfect morphisms and informational entropy

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

I've defined generalised models M as being given by F, a set of features, and an (unnormalised) probability distribution Q over $W = 2^{\overline{F}}$, the set of possible worlds defined by the all values of those features.

To make these into a category, a morphism r between $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$ was defined to be a relation r between W_0 and W_1 (ie a subset of $W_0 \times W_1$), that obeyed certain conditions with respect to the Q's.

If r obeys those conditions, we can construct the "underlying model" for the morphism, a generalised model $M_r = (F_0 \sqcup F_1, Q_r)$, with Q_r non-zero only on $r \in W_0 \times W_1$.

That underlying model result basically says:

- There is an underlying reality M_r of which M_0 and M_1 are different, consistent, facets.

That's well and good, but we also want to allow imperfect correspondences, where Q_0 or Q_1 (or both) might be known to be in error. After all, when we transitioned from Newtonian mechanics to relativity, it wasn't because there was an underlying reality that both were facets of. Instead, we realised that Newtonian mechanics and relativity were approximately though not perfectly equivalent in low-energy situations, and that when they diverged, relativity was overall more accurate.

We'd want to include these cases as generalised model, and measure how "imperfect" the morphism between them is, i.e. how much Q_0 and Q_1 diverge from being in perfect correspondence. We'll also look at how the Q and the feature sets are related - how much information Q caries relative to F.

Imperfect morphisms

So we'll loosen the definition of "morphisms" so that the Q_0 and Q_1 need not correspond exactly to each other or an underlying reality. If we have a relation r between W_0 and W_1, here are different Q-consistency requirements that we could put on r to make it into a morphism (the previous requirement has been renamed "Q-preserving", condition 5):

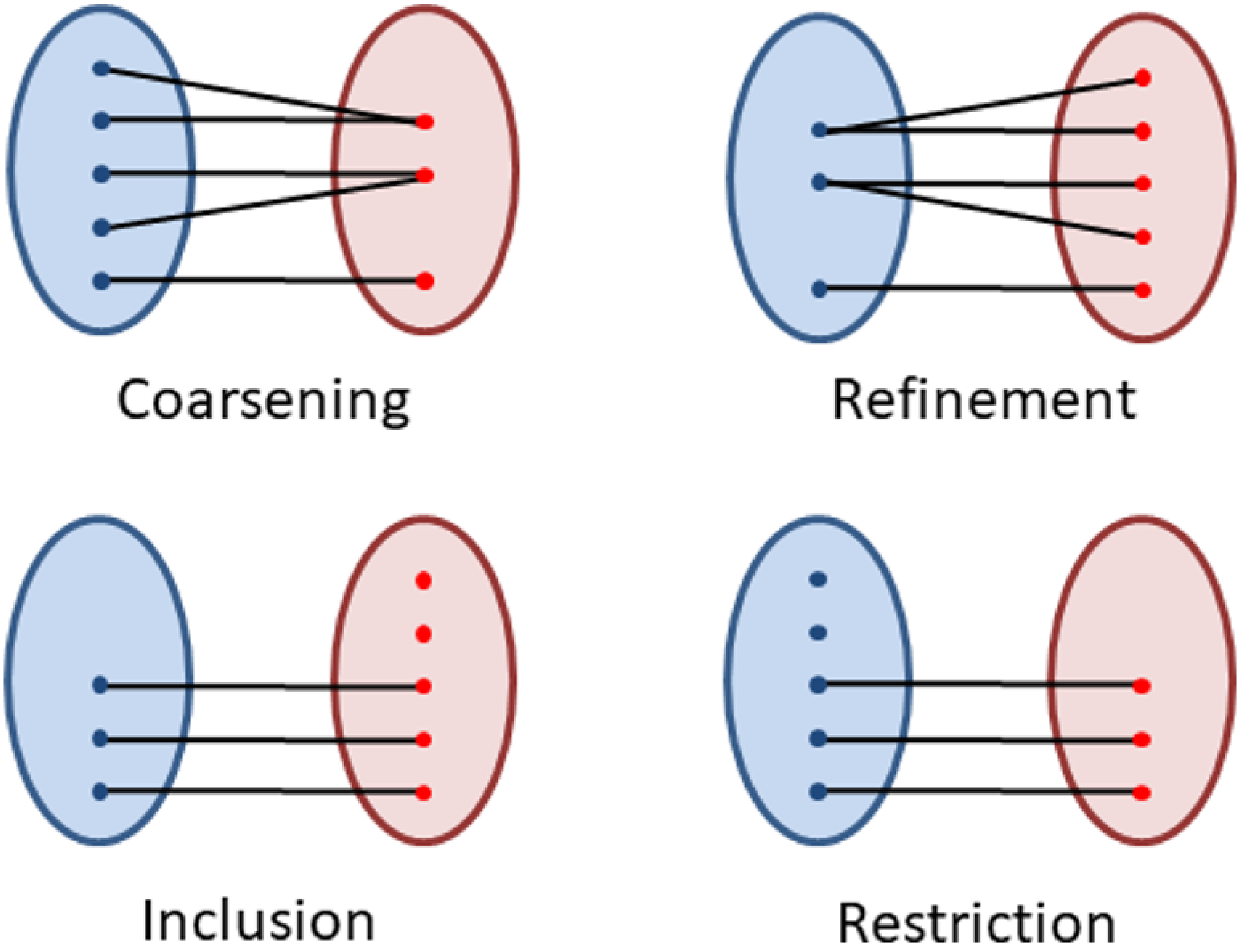
1. General binary relations r; no connection assumed between r and the Qs.
2. Q-relational: for all $w_0 \in W_0$ with $Q_0(w_0) > 0$, there exists at least one $w_1 \in W_1$ with both $Q_1(w_1) > 0$ and $(w_0, w_1) \in r$.
3. Q-functional: for all $w_0 \in W_0$ with $Q_0(w_0) > 0$, there exists a *unique* $w_1 \in W_1$ with both $Q_1(w_1) > 0$ and $(w_0, w_1) \in r$.
4. Q-birelational: r and r^{-1} are both Q-relational.
5. Q-preserving: the same conditions as presented here. For all $w_0 \in W_0$ and $w_1 \in W_1$, $Q_0(w_0) \leq Q_1(r(w_0))$ and $Q_1(w_1) \leq Q_0(r^{-1}(w_1))$.
6. Q-isomorphic: r is Q-preserving; both r and r^{-1} are Q-functional.

The general results about these morphisms classes are (proof found in this footnote[1]):

- All the above conditions are associative (hence define classes of morphisms for different categories with the same objects): if r and p are Q-relational, Q-functional, Q-relational, Q-birelational, Q-preserving, Q-isomorphic or disconnected from Q entirely, then so is $pr = p \circ r$.
- Every morphism fulfils the conditions of the morphisms above it on the list, except that Q-preserving and Q-birelational need not imply Q-functional.
- If r is Q-isomorphic, then we can pair up each non-measure zero elements $w_0 \in W_0$ and $w_1 \in W_0$ so that $(w_0, w_1) \in r$ and $Q_0(w_0) = Q_1(w_1)$.

Examples of morphisms

Here are four relations:



A coarsening is when multiple worlds get related to a single world, thus losing details. A refinement is the opposite: a single world gets related to multiple worlds, thus adding more details. An inclusion adds more worlds to the set. Its opposite, a restriction, removes worlds.

In terms of morphisms, if we assume that all the worlds in those sets have non-zero probability, then coarsenings, refinements, and inclusions are all Q-relational and Q-functional. Restrictions are neither. Coarsenings and refinements are Q-birelational, while inclusions and restrictions are not.

As for Q-preserving, coarsenings are Q-preserving if, for all $w_1 \in W_1$, the sum of $Q_0(w_0)$ for all $(w_0, w_1) \in r$, is equal to $Q_1(w_1)$. Similarly, refinements are Q-preserving if, for all $w_0 \in W_0$, the sum of the $Q_1(w_1)$ for all $(w_0, w_1) \in r$, is equal $Q_0(w_0)$. None of these four relations are Q-isomorphic (unless some of the worlds in the diagrams are of measure 0).

What about Bayesian updates? If we start with $M_0 = (\{f\} \sqcup F, Q_0)$ and want to update on $f = c$ for some constant c, then this corresponds to relating M_0 to $M_1 = \{F, Q_1\}$ such that $(w_0, w_1) \in r$ if $w_0 = (w_1, f = c)$. We'll also require $Q_0(w_0) = Q_1(w_1)$ on any such worlds - and assume that that defines Q_1 entirely (we're ignoring renormalisation here, since we don't assume that Q_0 and Q_1 have measure 1).

This r is clearly a restriction, and hence does not meet any of the Q-consistency conditions. However, we can define Bayesian updates as relations r such that r^{-1} is Q-functional and injective. They do form a category when seen this way (since $(rq)^{-1} = q^{-1}r^{-1}$).

Comparing the Q's

Given two generalised models $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$, with a relation r between them, we'll now compare Q_0 and Q_1. We'll do this by defining a length operator L that gives the "length" of r, which is a measure of the divergence between Q_0 and Q_1 "along" the relation r.

Let (Q'_0, Q'_1) be a pair of probability distributions on W_1 and W_1, respectively. We'll say the pair is r-compatible if r is a Q-preserving morphism between (F_0, Q'_0) and (F_1, Q'_1).

Since Q_i and Q'_i are distributions over the same W_i, we can compare their l_1 norm, defined as: $||Q_i - Q'_i||_1 = \sum_{w_i \in W_i} |Q_i(w_i) - Q'_i(w_i)|$. Then define $L(r, Q'_0, Q'_1)$ as the sum of the l_1-distances to Q_0 and Q_1:

$$L(r, Q'_0, Q'_1) = ||Q_0 - Q'_0||_1 + ||Q_1 - Q'_1||_1.$$

We'll define L(r) to be the minimum[2] value of this norm among all the r-compatible (Q_0', Q_1'). Because of its definition, it's immediately obvious that $L(r) = L(r^{-1})$. The other key properties - proved here[3] - are that:

- If r is Q-relational, then there exists a Q_1' such that $L(r) = L(r, Q_0, Q_1')$; ie we can use Q_0 itself rather than finding a Q_0'.
- If r and p are Q-birelational, the L is a sensible length operator, in that $L(pr) \leq L(r) + L(p)$.
- r is Q-preserving iff $L(r) = 0$.

It's that last property that makes L such a useful distance metric: it measures the extent to which M_0 and M_1 fail to be aspects of the same underlying reality.

Relating features and probability distributions

In the above we've been looking at the relationship between r and Q, but have looked little at the features. Here we'll look at some of the relations between features and probability distributions. The idea is to measure how related the features are to each other.

There are several candidates for a measure of this types, but the most interesting seems to be a generalisation of mutual information. For any feature $F \in F$, we have the marginal distribution Q_F of Q over the feature F. Then if H is the informational entropy of a probability distribution/random variable, we can define a measure over Q given F as:

$$-H(Q) + \sum_{F \in F} H(Q_F).$$

If we define Q_F as the product distribution $\prod_{F \in F} Q_F$, then that can also be defined as $D_{KL}(Q || Q_F)$, where D_{KL} is the KL-divergence from Q_F to Q.

Example: gas laws

As an illustration, consider the ideal gas laws: $PV = nRT$, where P is pressure, V is volume, n is the amount of substance, R is the ideal gas constant, and T is the temperature. We'll consider a simple model with constant amount of substance, setting $nR = 1$, so the law reduces to:

$$PV = T$$

We'll allow these variables to take a few values: P and V are integers that range from 1 to 4, while T is an integer that ranges from 1 to 16. The probability distribution Q will give uniform probability[4] 1/16 to each $(P = p, V = v, T = pv)$.

In this case, Q is uniform among the 16 worlds where it is non-zero; hence

$$H(Q) = \log_2(16) = 4.$$

This characterises Q, but doesn't touch the features F. So, let Q_P, Q_V, and Q_P be the marginal distributions over the features. Both Q_P and Q_V are uniform over 4 elements, so $H(Q_P) = H(Q_V) = 2$. As for Q_T, it is 1/16 over $\{1, 9, 16\}$, 2/16 over $\{2, 3, 6, 8, 12\}$, and 3/16 over $\{4\}$. Some calculations then establish that $H(Q_T)$ is $(54 - 3\log_2(3))/16$. Then the KL-divergence from $Q_F = Q_V Q_P Q_T$ to Q is:

$$\begin{aligned} D_{KL}(Q||Q_F) &= -4 + 2 + 2 + \frac{54-3\log_2(3)}{16} \\ &= \frac{54-3\log_2(3)}{16} \\ &\approx 3.08. \end{aligned}$$

Let us add another variable T' to the feature set, which is just equal to T, but with another name, and see how things change. Then $H(Q)$ is unchanged, and $D_{KL}(Q||Q_F)$ adds another $\frac{54-3\log_2(3)}{16}$, corresponding to $H(T')$.

---

1. We've already shown that Q-preserving morphisms are associative. The composition of general binary relations is known to be associative too.

So now assume that $r : M_0 \to M_1$ and $p : M_1 \to M_2$ are both Q-relational. Let $w_0 \in W_0$ be such that $Q_0(w_0) > 0$. Then, because r is Q-relational, there exists a $w_1 \in W_0$ with $Q_1(w_1) > 0$ and $(w_0, w_1) \in r$. Then since p is Q-relational, there exists a $w_2 \in W_2$ with $Q_2(w_2) > 0$ and $(w_1, w_2) \in p$. Combining the two gives $(w_0, w_2) \in pr$. Thus Q-relational morphisms are associative. Applying the same argument to r^{-1} shows that Q-birelational morphisms are also associative. A variant of the same argument with "there exists a unique $w_1 \in W_0$" instead of "there exists a $w_1 \in W_0$" shows that Q-functional morphisms are also associative. Hence Q-isomorphic r are also functional.

Now assume that r is Q-preserving and let $w_0 \in W_0$ be such that $Q_0(w_0) > 0$. Then there exists an underlying model $M_r = (F_0 \sqcup F_1, Q_r)$ such that $Q_0(w_0) = \sum_{(w_0,w_1)\in r} Q_r(w_0, w_1)$. Since this sum is greater than zero, there exists a w_1 such that $Q_r(w_0, w_1) > 0$. Then since $Q_1(w_1) = \sum_{(w_0^{'},w_1)\in r} Q_r(w_0^{'}, w_1)$ and all the terms are non-negative, $Q_1(w_1) > 0$. The same argument works if we started with a $w_1 \in W_1$ such that $Q_0(w_1) > 0$; thus r must be Q-birelational. This proves that Q-preserving implies Q-birelational.

It's trivial that Q-birelational implies Q-relational, and that Q-functional also implies Q-relational, since "exists a *unique*" is strictly stronger than "exists a". By definition, Q-isomorphic implies Q-preserving (hence Q-birelational and Q-relational) and Q-functional.

Now let r be Q-isomorphic, and let w_0 be such that $Q_0(w_0) > 0$. Since r is Q-functional, there exists a unique w_1 with $(w_0, w_1) \in r$ and $Q_1(w_1) > 0$. Since r^{-1} is also Q-functional, there are no other $w_0^{'}$ with $(w_0^{'}, w_1) \in r$ and $Q_0(w_0^{'}) > 1$. So, among the elements of non-zero measure, w_0 and w_1 are related only to each other. Then since r is Q-preserving, $Q_0(w_0) \leq Q_1(r(w_0) = Q_1(w_1) + 0$, and $Q_1(w_1) \leq Q_0(r^{-1}(w_0)) = Q_0(w_0) + 0$. Hence $Q_0(w_0) = Q_1(w_1)$. ↩

2. This will be a minimum, not an infimum. Let (Q_0', Q_1') be an r-compatible pair with $L(r, Q_0', Q_1') = \mu$. Then if we restrict to pairs (Q_0', Q_1') with $L(r, Q_0', Q_1') \leq \mu$, this is a compact non-empty set, so $L(r, -, -)$ must reach its minimum on this set. ↩

3. For r a relation between $M_0 = (F_0, Q_0)$ and $M_1 = (F_1, Q_1)$, let Q_r be the set of r-compatible (Q_0', Q_1') that minimises $L(r, Q_0', Q_1')$. Hence $L(r, -, -) = L(r)$ on Q_r.

   So we have this non-empty Q_r; what we'll show is that, if r is Q-relational, then there is a Q_1' such that $(Q_0, Q_1') \in Q_r$. We'll need the following lemma:

   - Lemma A: For any (Q_0', Q_1') in Q_r with $||Q_0 - Q_0'||_1 > 0$, we can find another pair $(Q_0'', Q_1'') \in Q_r$ with Q_0'' closer (in the l_1 norm) to Q_0 than Q_0' is.

   To prove the lemma, pick any $(Q_0', Q_1') \in Q_r$ with $||Q_0 - Q_0'||_1 > 0$. That l_1 norm is a sum of positive terms, so there must exist a $w_0 \in W_0$ with $|Q_0(w_0) - Q_0'(w_0)| > 0$.

   Assume first that $Q_0(w_0) < Q_0'(w_0)$. Then, since $Q_0'(w_0) > 0$ and r is a Q-preserving morphism between Q_0' and Q_1', there is an underlying model $M_r = (F_0 \cup F_1, Q_r)$ so that $Q_0'(w_1) = \sum_{(w_0, w_1) \in r} Q_r(w_0, w_1) > 0$. Thus there is a $(w_0, w_1) \in r$ with $Q_r(w_0, w_1) > 0$. Pick $\epsilon > 0$ to be less than $Q_r(w_0, w_1)$ and $|Q_0(w_0) - Q_0'(w_0)|$, and define:

$$\begin{aligned} Q_0''(w_0) &= Q_0'(w_0) - \epsilon \\ Q_1''(w_1) &= Q_1'(w_1) - \epsilon, \end{aligned}$$

with $Q_0'' = Q_0'$ and $Q_1'' = Q_1'$ on all other points. Because $Q_0''(w_0)$ is closer to $Q_0(w_0)$ (by ϵ) than Q_0 is, $||Q_0 - Q_0''||_1 = ||Q_0 - Q_0''||_1 - \epsilon$. Furthermore, r is Q-preserving between Q_0'' and Q_1'' (the underlying model has the same Q_r except increased by ϵ on (w_0, w_1)), and $||Q_1 - Q_1''||_1$ has gone up by at most ϵ over $||Q_1 - Q_1'||_1$; thus the sum $||Q_0 - Q_0''||_1 + ||Q_1 - Q_1''||_1$ has not increased. Hence $(Q_0'', Q_1'') \in Q_r$.

Now consider the other case: $Q_0(w_0) > Q_0'(w_0)$. Then since $Q_0(w_0) > 0$ and r is Q-relational, there must exist a $(w_0, w_1) \in r$. We define Q_0'' and Q_1'' as above, except that we add ϵ instead of subtracting it; the rest of the argument is the same. This proves the lemma □.

Back to the main proof. Since Q_r is compact, the l_1 distance to Q_0 must reach a minimum on Q_r. By lemma A, this minimum can only be 0 (since if it were greater than 0, we could find a pair with a smaller l_1 distance to Q_0). If $(Q_0', Q_1') \in Q_r$ is a pair on which it reaches this minimum, we must have $||Q_0 - Q_0'||_l = 0$, ie $Q_0' = Q_0$.

Now assume r is Q-birelational between M_0 and M_1, while p is Q-birelational between M_1 and M_2. Then $L(r) + L(p) = L(r^{-1}) + L(p)$. Since r^{-1} and p are Q-relational, there exists Q_0' and Q_2' such that this is $L(r^{-1}, Q_1, Q_0') + L(p, Q_1, Q_2')$, and (Q_0', Q_1) is r-compatible, while (Q_1, Q_2') is p-compatible.

This implies that (Q_0', Q_2') is pr-compatible, and thus $L(pr, Q_0', Q_2') \geq L(pr)$. However,

$L(r^{-1}, Q_1, Q_0') + L(p, Q_1, Q_2') = ||Q_0 - Q_0'||_1 + 0 + 0 + ||Q_2 - Q_2'||_1 = L(pr, Q_0', Q_2')$, giving our result:

$$L(pr) \leq L(r) + L(p).$$

Finally, notice that $L(r) = 0$ implies that there exists an r compatible (Q_0', Q_1') with $||Q_0 - Q_0'||_1 = 0$ and $||Q_1 - Q_1'||_1 = 0$. Thus (Q_0, Q_1) themselves are r-compatible, ie r is Q-preserving. Conversely, if (Q_0, Q_1) are r-compatible, then $L(r) \leq ||Q_0 - Q_0||_1 + ||Q_1 - Q_1||_1 = 0$, and thus $L(r) = 0$. ↩

4. Note that this means that many values of T are impossible in this model, such as 7 and 10, which cannot be expressed as the product of integers 4 or less. ↩