Formalizing Two Problems of Realistic World-Models

Nate Soares
Machine Intelligence Research Institute
nate@intelligence.org

Abstract

An intelligent agent embedded within the real world must reason about an environment which is larger than the agent, and learn how to achieve goals in that environment. We discuss attempts to formalize two problems: one of induction, where an agent must use sensory data to infer a universe which embeds (and computes) the agent, and one of interaction, where an agent must learn to achieve complex goals in the universe. We review related problems formalized by Solomonoff and Hutter, and explore challenges that arise when attempting to formalize analogous problems in a setting where the agent is embedded within the environment.

1 Introduction

An intelligent agent embedded in the real world faces an induction problem: how can it learn about the environment in which it is embedded, about the universe which computes it? Solomonoff (1964) formalized an induction problem faced by agents which must learn to predict an environment which does not contain the agent, and this formalism has inspired the development of many useful tools, including Kolmogorov complexity and Hutter's AIXI. However, a number of new difficulties arise when the agent must learn about the environment in which it is embedded.

An agent embedded in the world also faces an interaction problem: how can an agent learn to achieve a complex set of goals within its own universe? Legg and Hutter (2007) have formalized an "intelligence measure" which scores the performance of agents that learn about and act upon an environment that does not contain the agent, but again, new difficulties arise when attempting to do the same in a naturalized setting.

This paper examines both problems. Section 2 introduces Solomonoff's formalization of an induction problem where the agent is separate from the environment, and Section 3 discusses troubles that arise when attempting to formalize the analogous naturalized induction problem. Section 4 discusses Hutter's interaction problem, and Section 5 discusses an open problem related to formalizing an analogous naturalized interaction problem.

Formalizing these problems is important in order to fully understand the problem faced by an intelligent agent embedded within the universe: a general artificial intelligence must be able to learn about the environment which computes it, and learn how to achieve its goals from inside its universe. Section 6 concludes with a discussion of why a theoretical understanding of agents interacting with their own environment seems necessary in order to construct highly reliable smarter-than-human systems.

Research supported by the Machine Intelligence Research Institute (intelligence.org). Published as Technical report 2015–3.

2 Solomonoff's Induction Problem

Solomonoff (1964) posed one of the earliest and simplest descriptions of a problem in which an agent must construct realistic world-models and promote correct hypotheses based on observations, performing reasoning akin to scientific induction. Intuitively, the problem considered by Solomonoff runs as follows: the universe is separated into an *agent* and an *environment*. Every turn, the agent observes one bit of output from the environment. The task of the agent is to, in each turn, predict its next observation.

To formally describe the agent's performance, it is necessary to decide what counts as a possible environment, then to decide how to measure how well an agent predicts an environment, and then to choose the distribution over environments against which the agent will be scored.

What counts as a possible environment? In Solomonoff's formalization, the goal is to consider hypothetical agents which can learn an arbitrarily complex environment, and so Solomonoff chooses the set of environments to be anything that is computable. This can be formalized by defining the set of all environments as the set $\mathcal{T}$ of all Turing machines with access to an advance-only output tape.

How are an agent's predictions scored? Consider an environment $M \in \mathcal{T}$ where M_n denotes the n^{th} bit on the output tape of M. Let an agent A be a function which takes a string $M_{\prec t}$ of observations made before turn t, which outputs a prediction of M_t in the form of a rational number interpreted as the probability that $M_t = 1$. For convenience, define $A_t := A(M_{\prec t})$. To score A against M on all time steps, it is necessary to account for the fact that M may eventually stop outputting bits; define $\lceil M \rceil$ to be the last turn in which M outputs a bit (this value may be ∞). Then A may be scored against M using standard logarithmic loss:[1]

$$\mathcal{S}_M(A) := \sum_{t=1}^{\lceil M \rceil} M_t \log(A_t) + (1 - M_t) \log(1 - A_t). \quad (1)$$

Against which distribution over Turing machines should the agent be scored? The answer determines which agents are considered to be "good predictors." If the agent is to be evaluated against its ability to learn one specific environment, the trivial distribution containing only that environment may be chosen—but then the high-scoring agents would be agents which have that environment hard-coded into them; this would hardly be a problem of *learning*. The choice of distribution defines the manner in which agents must be biased to achieve a high score: how should predictors be biased?

The natural answer comes in the form of an intuition canonized by William of Ockham seven hundred years ago: predictors in the real world do well to prefer the simplest explanation which fits the facts. There are exponentially more possible explanations of increasing complexity (e.g. 2^N N-bit explanations) and so any particular explanation of greater complexity should have less probability. Thus it seems natural to score the agent according to a distribution in which simple environments have greater weight than complex environments. The most natural way to define a simplicity distribution over Turing machines is to fix some universal Turing machine U,[2] and assign probability $2^{-\langle M \rangle}$ to each Turing machine M, where $\langle M \rangle$ is the number of bits needed to specify M to U.

Now Solomonoff's induction problem may be fully described: An environment is any Turing machine M with an advance-only output tape. An agent A is a function which takes an output history and produces a rational number interpreted as the probability that the next observation will be 1. The agent is scored according to $S_M(A)$ against a simplicity distribution. Formally, *Solomonoff's induction problem* is the problem of maximizing the "Solomonoff induction" score

$$\text{SI}(A) := \sum_{M \in \mathcal{T}} 2^{-\langle M \rangle} \cdot S_M(A). \quad (2)$$

1. This score may not converge, in the infinite case, but it is nevertheless useful for comparing agents.

2. U must be chosen such that $\sum_{M \in \mathcal{T}} 2^{-\langle M \rangle} = 1$.

Like many good problem descriptions, this one lends itself readily to an idealized unbounded solution, known as Solomonoff induction:

> **Solomonoff induction.** The agent starts with a simplicity distribution over $\mathcal{T}$. Upon receiving the n^{th} observation o_n, it conditions its distribution on this observation by removing all Turing machines that do not produce n bits, or that do not write o_n as the n^{th} bit on their output tape. It then predicts that the $(n+1)^{\text{th}}$ bit is a 1 with probability equal to the measure on remaining Turing machines which write 1 as the $(n+1)^{\text{th}}$ bit on their output tape.

Indeed, it is in terms of this idealized solution that Solomonoff originally posed his induction problem (Solomonoff 1964).

A Solomonoff inductor is a very powerful predictor. It can "learn" any computable environment: Solomonoff (1978) showed that given any computable probability distribution over bit strings, a Solomonoff inductor's predictions will converge to the true probabilities.

With his induction problem, Solomonoff provides a full description of a scenario in which an agent must learn an arbitrarily complex computable environment separate from the agent. Insights from the induction problem have proven useful in practice: This problem became the basis of algorithmic information theory (Hutter, Legg, and Vitanyi 2007). The simplicity prior over Turing machines is the celebrated "universal prior" (Solomonoff 2003). Solomonoff induction is a crucial ingredient in Hutter's AIXI (2000) that solves the analogous universal decision problem, and many of Solomonoff's insights are present in the Legg-Hutter "universal measure of intelligence" (2007). Solomonoff's work served as the basis for Kolmogorov complexity (Solomonoff 1960), a powerful conceptual tool in computer science.

Unfortunately, the prediction problem faced by agents acting in the real world is not Solomonoff's induction problem: it is a problem of an agent modeling a world in which the agent is embedded as a subprocess, where the agent is made out of parts of the world and computed by the universe. Formally describing this more realistic problem turns out to be significantly more difficult.

3 The Naturalized Induction Problem

In Solomonoff's induction problem, the agent and its environment are fundamentally separate processes, connected only by an observation channel. In reality, agents are embedded *within* their environment; the universe consists of some ontologically continuous substrate (atoms, quantum fields) and the "agent" is just a part of the universe in which we have a particular interest. What, then, is the analogous prediction problem for

agents embedded within (and computed by) their environment?

This is the *naturalized induction problem*, and it is not yet well understood. A good formalization of this problem, on par with Solomonoff's formalization of the computable sequence induction problem, would represent a significant advance in the theory of general reasoning.

An analogous formalization of the naturalized induction problem must yield a scoring metric akin to $\mathrm{SI}(\cdot)$, which scores an algorithm's ability to predict its environment. But what metric is this? There are at least three open questions of naturalized induction:

First, given an algorithm, what is the set of all environments that the algorithm could have to induce? It would be strange to score the agent against all computable environments, as almost all Turing machines will not in fact embed that algorithm. Perhaps the set of environments could be defined with respect to the proposed algorithm, as the set of Turing machines which embed it. But how is that set defined? What does it mean for a Turing machine to "embed" an algorithm? Intuitions about embeddings have been difficult to formalize.

If the naturalized induction problem is to capture the problem of an agent learning about the real world, then the set of environments must contain reality. The set of all environments, therefore, must be a set of "possible realities": what structure is this? Does the set of all Turing machines actually contain our universe? Currently, the Standard Model of physics seems computable to any desired finite precision. But then again, reality looked Newtonian to scientists in centuries past. If artificial agents are to be able to surpass their programmers' scientific knowledge, a formalization of intelligent learning should not presuppose the correctness of present-day physical science. Modern theories as to the nature of physical reality may turn out to be mistaken or incomplete, and an ideal reasoner must be able to adapt to such surprises. What set of possible environments definitely contains reality, in light of the potential for surprises?

Second, given an environment drawn from this set of possible environments, how is the agent's ability to learn that environment scored? Are agents scored better for constructing new sensors? Are they scored better for finding some way to affect their environment so as to make it easier to predict? These are not questions that can be reduced to Solomonoff's prediction task. Formalizing inductive success is much more difficult when the environment can act on the agent's internals, and when the agent-environment boundary can shift over time. Questions of evaluation are further covered in sections 4 and 5.

Third, given a set of possible environments and a scoring metric, what is the distribution against which an agent should be scored? As in Solomonoff's induction problem, this distribution must capture reality's bias towards simplicity, but defining a simplicity distribution over some set of "possible realities" may be nontrivial.

Of course, answers to these questions would be impractical at best and almost certainly uncomputable, but they would yield conceptual tools by which practical programs implementing sufficiently advanced agents (that face the naturalized induction problem) could be evaluated. For example, a formalization of naturalized induction would likely shed light on questions about how a reasoner should let the fact that it exists affect its beliefs, and may further lend insight into what sort of priors an ideal reasoner would use. Unfortunately, it is not yet clear how to approach the problems outlined above.[3]

Can Solomonoff induction be ported into a naturalized context? Perhaps, but the application is not straightforward. Even ignoring problems of ensuring that the environment has something corresponding to the "turns" and "observations" of Solomonoff's induction problem, Solomonoff's approach solves the problem by simply *being larger* than the environment: a Solomonoff inductor contains a distribution over all Turing machines, and one of those is, by assumption, the "real" environment. Solutions of this form don't apply when the agent is a subprocess within the environment.

Computable approximations of Solomonoff induction can be limited to the consideration of only "reasonably sized" environments, but this does not much help. Imagine a Solomonoff inductor which only considers Turing machines which can be specified in length less than l and which run for at most t steps between each turn:[4] this inductor would run for more than t steps per turn, and therefore the environment it is in would run for more than t steps per turn. The inductor would assign zero probability to its own existence!

An agent embedded in an environment must reason about an environment that is larger than itself; this constraint is inherent to naturalized induction. Solutions will require agents to reason about environments which they cannot compute. Reasoning of this form is known as "logically uncertain" reasoning, and it may be possible to port a logically uncertain variant of Solomonoff into a naturalized context. However, a satisfactory theory of reasoning under logical uncertainty does not yet exist. (For further discussion, see Soares and Fallenstein [2015a].)

4 Hutter's Interaction Problem

Even a full description of naturalized induction would not completely describe the problem faced by an intel-

3. Orseau and Ring (2012) give a characterization of the problem which *humans* face, in implementing a space-time embedded agent, but their problem description requires that we provide a distribution ρ which already characterizes our beliefs about the environment, and so sheds little light on questions of naturalized induction.

4. Such as AIXI^{tl}, a computable approximation of Hutter's AIXI (2000).

ligence acting in the world. Real agents must not only predict their environment, but act upon it.

With this in mind, Hutter (2000) extends Solomonoff's induction problem to an *interaction* problem, in which an agent must not only learn the external environment but interact with it. Hutter's interaction problem runs similarly to Solomonoff's induction problem: the universe is separated into *agent* and *environment*, and the agent gets to observe the environment through an input channel. But now, an "output channel" is added, which lets the agent affect the environment by one "action" per turn.

As before, a formalization requires answers to the questions of (1) what counts as an environment; (2) how an agent is scored on each environment; and (3) against which distribution over environments the agent is scored. In Hutter's interaction problem, the first and last answers follow readily from Solomonoff induction, with some minor tweaks. It is the answer to the second question, of scoring, where Hutter's interaction problem provides new insight.

Again, the set of all environments can naturally be defined as the set $\mathcal{T}$ of all Turing machines. However, instead of having an advance-only output tape, environments are now Turing machines which take an observation/action history and compute the next observation to be sent to the agent. That is, fix some countable set $\mathcal{O}$ of observations which can be sent from environment to agent, and some countable set $\mathcal{A}$ of actions which can be sent from agent to environment, and then consider Turing machines which take a finite list of actions and compute a new observation $\mathcal{O}$. An agent, then, is any function which takes a finite list of observations and computes a new action A.[5] Again, the distribution over environments will be the "universal" simplicity distribution (with respect to some fixed universal Turing machine U).

It remains to decide how an agent is scored: what counts as the "success" of an agent A interacting with an environment M? Hutter (2000) formalizes interaction as follows. First, observations are defined such that one part of the observation is a *reward*; that is, elements of $\mathcal{O}$ are tuples (o, r) where r is a rational number between 0 and 1, and o is additional observation data. Let $M_t^A \in \mathcal{O}$ denote the t^{th} output of the machine M when interacting with A, and let $A_t^M \in \mathcal{A}$ denote the t^{th} action of A when interacting with M. That is,

$$\begin{aligned} M_1^A &:= M() \\ A_1^M &:= A(M()) \\ M_t^A &:= M(A_{\prec t}^M) \\ A_t^M &:= A(M_{\preceq t}^A). \end{aligned}$$

Let r_t^A denote the reward part of the observation M_t^A. Restrict consideration to the set $\mathcal{T}_r$ of environments where rewards converge, that is, to environments M such that $0 \leq \sum_{t=1}^{\lceil M \rceil} r_t^A \leq 1$ for all agents A. The total rewards observed by an agent A interacting with M are then used to score the agent, that is, define

$$R_M(A) := \sum_{t=1}^{\lceil M \rceil} r_t^A. \tag{3}$$

This function measures the ability of A to learn and manipulate M in order to maximize observed rewards over time.

This choice of scoring mechanism yields a full description of Hutter's interaction problem: it describes a setting where an agent must interact with an environment in order to learn and maximize rewards. Indeed, this scoring metric is used to define the "universal measure of intelligence" of Legg and Hutter (2007):

$$\mathrm{LH}(A) := \sum_{M \in \mathcal{T}_r} 2^{-\langle M \rangle} \cdot R_M(A). \tag{4}$$

We refer to the problem of finding agents which score highly according to $\mathrm{LH}(\cdot)$ as *Hutter's interaction problem.*

As before, this problem description lends itself readily to an idealized solution. In this case, the solution is Hutter's AIXI (2000), which in fact was the mechanism by which Hutter first posed the interaction problem:

> **AIXI.** The agent starts with a universal prior, which it keeps consistent with observation using Solomonoff induction (modified in the natural way for this problem). AIXI chooses its action as follows: it has some fixed time horizon h, and considers all possible sequences of h actions. It computes the expected reward (according to its distribution over environments) for each sequence, and then outputs the first action in the sequence that leads to highest rewards.

AIXI is an incredibly powerful and elegant agent. As noted by Veness et al. (2011), AIXI captures "the major ideas of Bayes, Ockham, Epicurus, Turing, von Neumann, Bellman, Kolmogorov, and Solomonoff" in a single equation. Barring a few minor quibbles,[6] AIXI fully

5. Hutter (2000) uses a generalization in which both agent and environment may be stochastic; in this case it is necessary for agent and environment to receive a history of both observations and actions. In the deterministic version, used here for ease of exposition, the agent (environment) does not need to be told the history of actions (observations) because past actions (observations) may simply be recomputed.

6. The finite time horizon of AIXI is both arbitrary and

characterizes a solution to Hutter's interaction problem: while AIXI is uncomputable, it demonstrates the method by which a high LH(·) score may be attained. Indeed, computable approximations of AIXI have already yielded interesting results (Veness et al. 2011).

If the problem faced by intelligent agents acting in the real world to achieve goals was characterized by Hutter's interaction problem, then this problem description would fully characterize the problem of constructing smarter-than-human systems, and the problem of general intelligence would be reduced to one of approximating AIXI.

However, Hutter's interaction problem does not capture the problem faced by an agent acting in the real world to achieve goals. Rather, it characterizes the problem of an agent attempting to maximize sensory rewards from an environment that can only affect the agent via sensory information.

While this problem description has yielded many insights, the distinction is important: the simplifying assumptions of Hutter's interaction problem mask a number of difficult open problems.

4.1 The Agent is Not Separate from the Environment

Hutter's interaction problem, like Solomonoff's induction problem, assumes an impregnable separation between the agent and the environment. In Solomonoff's case, there is (figuratively speaking) a small slit through which the environment feeds sensory information to the agent. Hutter adds a second slit, through which the agent outputs motor signals to the environment. However, the separation remains otherwise complete. Thus, the questions of naturalized induction remain unanswered, and Hutter's interaction problem yields little new insight there.

For this reason, Hutter's interaction problem cannot capture certain realistic scenarios that intelligent agents may actually face: the Legg-Hutter measure of intelligence is ill-defined in any situation where the universe cannot crisply be divided into "agent" and "environment," when interactions cannot be crisply divided into "input" and "output." For example, consider the following simple setting in which it matters that the agent is embedded within its environment:

> **The Heating Up game.** An agent A faces a box containing prizes. The box is designed to allow only one prize per agent, and A may execute the action P to take a single prize. However, there is a way to exploit the box, cracking it open and allowing A to take all ten prizes. A can attempt to do this by executing the action X. However, this procedure is computationally very expensive: it requires reversing a hash. The box has a simple mechanism to prevent this exploitation: it has a thermometer, and if it detects too much heat emanating from the agent, it self-destructs, destroying all its prizes.

If the agent heats up too much, it gets reward 0, no matter what action it takes. If it does not heat up too much, then it gets reward 1 for action P or reward 10 for action X. But the amount of heat generated by the agent, of course, is dependent upon which action the agent chooses.

This scenario captures an important aspect of reality: a generally intelligent agent must be able to consider the consequences of overheating (along with many other consequences of being embedded within a universe). However, this scenario can't be modeled as an interaction problem. The Legg-Hutter measure of intelligence does not pit agents against scenarios such as these; there is no combination of $M \in \mathcal{T}_r$ and A which captures this sort of problem.

When evaluating an agent in a Heating Up game, the agent cannot be treated as separate from the environment. Rather, the agent must be located *within* the environment, and then somehow scored according to what it "could have done." Is it possible for a clever agent to compute X without ever once getting too hot? This question depends upon the specific environment and upon the agent's specific hardware.

This highlights a host of new "naturalized" questions: Given an environment that embeds an agent, how is the agent located in that environment? How are the actions that it "could have taken" identified? In Hutter's interaction problem, these questions are simplified away: the input and output channels are clearly demarcated; the environment is *defined* in terms of the agent's actions. AIXI, when considering the effects of various sequences of actions, can simply run a Turing machine on the considered action sequence; the behavior of the environment on that sequence of actions is well-defined. When an agent is *embedded within* an environment, the question is more difficult. For simplicity, consider a deterministic agent embedded in a deterministic environment. What does it mean to ask what "would happen" if the part of the environment labeled "agent" outputs something it doesn't? How is the counterfactual defined? Counterfactual reasoning is not yet well understood (Soares and Fallenstein 2015b).

Hutter's interaction problem extends Solomonoff's induction problem to capture a critical aspect of the problem faced by intelligent agents: environments that can be altered by agent decisions. This yields many insights, but moving forward, a *naturalized interaction problem* is necessary: how can an agent learn and manipulate the environment in which it is embedded, to achieve some set of goals? It is this problem which would fully characterize the problem faced by intelligent

disconcerting: for any time horizon h, consider an environment with a button that gives -1 when pressed, pays $+10$ h steps thereafter, and pays out -100 on the step after that. AIXI with time horizon h, after learning that this is the environment, presses the button indefinitely.

systems acting in the real world.

4.2 Goals Cannot Be Specified in Terms of Observation

Ignoring the need for a naturalized interaction problem, Hutter's interaction problem still does not quite capture the problem faced by an agent which must learn and manipulate an environment to achieve some set of goals. Rather, it characterizes the problem faced by an agent which must maximize *rewards*, specified in terms of observations. But most sets of goals cannot be characterized in terms of the agent's observations!

Consider an interaction problem in some approximation of reality where there is a crisp separation between "agent" and "environment," where the input and output channels are clearly demarcated. The agent's input is a video stream, and rewards are only nonzero when there are smiling human faces on the video screen. This agent, if possessing of a high LH(·) score, will very likely gain control of its input stream, such as by placing a photo with many smiling faces in front of the camera and then acting to ensure that it stays there.

Agents with high LH(·) scores are extremely effective at optimizing the extent to which their observations contain rewards; these are not likely to be agents which optimize the desirable feature of the world that the rewards are meant to serve as a proxy for. Rather, they are likely to be agents which are very good at taking over their input channels.

Reinforcement learning techniques, such as having the humans dole out rewards via some reward channel, would not solve the problem. Humans could attempt to prevent the agent from taking over its reward channel by penalizing the agent whenever they notice it performing actions that would give it control over rewards, and this may prevent the agent from executing those plans for a time. However, if the agent scores sufficiently high by LH(·), then once its dominant hypotheses about the environment agree that the humans are controlling the reward channel, it would act to mollify the programmers while searching for ways to gain a decisive advantage over them. If the agent is a sufficiently intelligent problem-solver, it may eventually find a way to wrest control of the reward channel away from the programmers and maintain it permanently (Bostrom 2014, chap. 8).

Even faced with incredibly high-fidelity input channels designed to be expensive to deceive, LH(·) rewards agents that set up Potemkin villages[7] which trigger the reward using minimum resources. An agent optimizing a reward function only optimizes the actual goals if achieving the goals is the cheapest possible way to get the reward inputs. Guaranteeing such a thing is nigh impossible: consider the genetic search process of Bird and Layzell (2002), which, tasked with designing an oscillating circuit, re-purposed the circuit tracks on its motherboard to use as a radio which amplified oscillating signals from nearby computers. Highly intelligent systems might well find ways to maximize rewards using clever strategies that the designers assumed were impossible, or that they never considered in the first case.

A high LH(·) score indicates that an agent is extremely proficient at commandeering its reward channel. Therefore, this intelligence metric does not quite capture the intuitive notion of how well an agent would fulfill a given set of goals.

There is no all-purposes patch for this problem within a sensory rewards framework. We do not care about what the agent *observes*; rather, we care about what *actually happens*. To evaluate the performance of an agent, it is not sufficient to look only at the inputs which the agent has received: one must also look at the outcomes which the agent achieves.

5 Ontology Identification

To evaluate how well an agent achieves some set of goals, it is important to measure the resulting *environment history*, not just the agent's observation history. In Hutter's interaction problem, an "environment history" is the combination of a Turing machine along with an observation/action history. But goals are not defined in terms of Turing machines and $\mathcal{O}/\mathcal{A}$ histories; goals are defined in terms of things like money, or efficient airplane flight patterns, or flourishing humans. How do you measure those things, given a Turing machine and an $\mathcal{O}/\mathcal{A}$ history?

As a matter of fact, it is quite difficult to say what terms our goals are specified in. To leave aside problems of philosophy, and highlight the problem as it pertains to world models, let us imagine that our goals are simple and can be specified according to a structure that seems fairly objective in our environment: assume that agents will be evaluated by how much diamond they create in their environment, where "diamond" is specified concretely in terms of a specific atomic structure. That is, the score of an agent is the count of carbon atoms covalently bound to four other carbon atoms over time.

Now the goals are given in terms of atomic structures, and the environment-history is given in terms of a Turing machine paired with an $\mathcal{O}/\mathcal{A}$ history. How is the Turing machine's representation of atoms identified?

This is the *ontological identification* problem. Whatever set is used for the set of all environments against which the agent is measured, it must be possible to inspect elements of that set and rate them according to our goals, and for that it is necessary to interpret features of the environment in terms of the ontology of the goals. This is an aspect of the naturalized interaction problem where Hutter's interaction problem sheds little insight.

It seems intuitively plausible that any detailed model of reality, in an environment such as the real world

7. A common idiom named after Gregory Potemkin, who set up fake villages to impress Empress Catherine II.

where diamond actually exists, must have some part of its internal structure which roughly corresponds to "atoms." However, the problem is made more difficult by the fact that the ontology of the goals will not actually perfectly match the ontology of reality: how are the atoms identified in a model of reality which runs on quantum mechanics? The model (if accurate) will still have systems that correspond, in some fashion, to the objects we call "atoms," much as the atomic model has systems corresponding to what we call "water." However, the correspondence may be convoluted and full of edge cases. How can the ontology of the goals be reliably mapped onto the ontology of the model? de Blanc (2011) provides a preliminary examination of these questions, but the problem remains open.

Ontology identification is the final step in the formal specification of the problem which is actually faced by an intelligent agent acting in the real world and attempting to fulfill some set of goals. To specify a measure of how well an agent would achieve the intended goals from within a universe, it must be possible to evaluate a model of the universe in terms of the goals. This requires the ability to take a model of reality, running on unknown and potentially surprising physics, and find within it the flawed and leaky abstractions with respect to which our goals are defined.

6 Discussion

The development of smarter-than-human machines could have a large impact upon humanity (Bostrom 2014), and if those systems are not aligned with human interests, the result could be catastrophic (Yudkowsky 2008). Highly reliable agent designs are crucial, and when constructing smarter-than-human systems, testing alone is not enough to guarantee high reliability (Soares and Fallenstein, forthcoming).

In order to justify high confidence that a practical smarter-than-human system will perform well in application, it is important to have a theoretical understanding of the formal problem that the practical system is intended to solve. The problems faced by smarter-than-human systems reasoning within reality are inherently naturalized problems: real systems must reason about a universe which computes the system, a universe that the system is built from.

The formalization of Solomonoff's induction problem yielded conceptual tools, such as the universal prior and Kolmogorov complexity, which are useful for reasoning about programs which predict computable sequences. It would be difficult indeed to construct highly reliable practical heuristics that predict computable sequences without understanding concepts such as simplicity priors.

We expect that naturalized analogs of the induction problem of Solomonoff and the interaction problem of Hutter will yield analogous conceptual tools useful for constructing systems that reason reliably about the universe in which they are embedded. Just as the intelligence metric of Legg and Hutter (2007) fully characterizes the problem of an agent interacting with a separate computable environment to maximize rewards, a corresponding metric derived from a naturalized interaction problem would fully characterize the problem faced by an intelligent agent achieving goals from within a universe.

It is not yet clear, in principle, what sort of reasoners perform well when tasked with acting upon their environment from within. Without a formal understanding of the problem, it would be difficult to justify high confidence in a system intended to face a naturalized interaction problem in reality. It is our hope that gaining a better understanding of these problems today will make it easier to design highly reliable smarter-than-human systems in the future.

References

Bird, Jon, and Paul Layzell. 2002. "The Evolved Radio and Its Implications for Modelling the Evolution of Novel Sensors." In *Congress on Evolutionary Computation. CEC-'02,* 2:1836–1841. Honolulu, HI: IEEE.

Bostrom, Nick. 2014. *Superintelligence: Paths, Dangers, Strategies.* New York: Oxford University Press.

de Blanc, Peter. 2011. *Ontological Crises in Artificial Agents' Value Systems.* The Singularity Institute, San Francisco, CA, May 19. arXiv: `1105.3821 [cs.AI]`.

Hutter, Marcus. 2000. "A Theory of Universal Artificial Intelligence based on Algorithmic Complexity." arXiv: `0004001 [cs.AI]`.

Hutter, Marcus, Shane Legg, and Paul M. B. Vitanyi. 2007. "Algorithmic Probability." *Scholarpedia* 2 (8): 2572.

Legg, Shane, and Marcus Hutter. 2007. "Universal Intelligence: A Definition of Machine Intelligence." *Minds and Machines* 17 (4): 391–444.

Orseau, Laurent, and Mark Ring. 2012. "Space-Time Embedded Intelligence." In *Artificial General Intelligence: 5th International Conference, AGI 2012,* 209–218. Lecture Notes in Artificial Intelligence 7716. New York: Springer.

Soares, Nate, and Benja Fallenstein. Forthcoming. "Aligning Superintelligence with Human Interests: A Technical Research Agenda." In *The Technological Singularity: Managing the Journey,* edited by Jim Miller, Roman Yampolskiy, Stuart Armstrong, and Vic Callaghan, vol. 2. Springer.

———. 2015a. *Questions of Reasoning Under Logical Uncertainty.* Technical report 2015–1. Berkeley, CA: Machine Intelligence Research Institute. `https://intelligence.org/files/QuestionsLogicalUncertainty.pdf`.

———. 2015b. "Toward Idealized Decision Theory." arXiv: `1507.01986 [cs.AI]`.

Solomonoff, Ray J. 1960. *A Preliminary Report on a General Theory of Inductive Inference.* Technical report ZTB-138. Cambridge, MA: Zator Co., November.

Solomonoff, Ray J. 1964. "A Formal Theory of Inductive Inference. Part I." *Information and Control* 7 (1): 1–22.

———. 1978. "Complexity-Based Induction Systems: Comparisons and Convergence Theorems." *IEEE Transactions on Information Theory* 24 (4): 422–432.

———. 2003. "The Kolmogorov Lecture: The Universal Distribution and Machine Learning." *The Computer Journal* 46 (6): 598–601.

Veness, Joel, Kee Siong Ng, Marcus Hutter, William Uther, and David Silver. 2011. "A Monte-Carlo AIXI Approximation." *Journal of Artificial Intelligence Research* 40:95–142.

Yudkowsky, Eliezer. 2008. "Artificial Intelligence as a Positive and Negative Factor in Global Risk." In *Global Catastrophic Risks,* edited by Nick Bostrom and Milan M. Ćirković, 308–345. New York: Oxford University Press.