

This paper is out-of-date. See
<http://www.danieldewey.net/>
for the latest version.

Learning What to Value

Daniel Dewey

Singularity Institute for Artificial Intelligence

Abstract. I. J. Good’s theory of an “intelligence explosion” predicts that ultraintelligent agents will undergo a process of repeated self-improvement. In the wake of such an event, how well our values are fulfilled will depend on whether these ultraintelligent agents continue to act desirably and as intended. We examine several design approaches, based on AIXI, that could be used to create ultraintelligent agents. In each case, we analyze the design conditions required for a successful, well-behaved ultraintelligent agent to be created. Our main contribution is an examination of *value-learners*, agents that learn a utility function from experience. We conclude that the design conditions on value-learners are in some ways less demanding than those on other design approaches.

1 Introduction

This paper is about the design of ultraintelligent agents. Following Good [2], we define an ultraintelligent agent as an agent that can “far surpass all the intellectual activities of any man”. An ultraintelligent agent could improve upon its own design, and since most agents would value self-improvement as a means to achieving their goals [6], an *intelligence explosion* of iterated self-improvement would occur; in Good’s words, “the intelligence of man would be left far behind”. When designing potentially ultraintelligent agents, therefore, we must be able to predict whether they (and their offspring) will achieve our goals, continuing to act desirably and as intended, in the wake of an intelligence explosion. Here, we examine several design approaches for ultraintelligent agents.

Reward maximizers, such as AIXI [4][5], are a subset of the class of all agents, and we argue (Appendix B) that they display a predictable and undesirable behavior pattern as they self-improve.

O-maximizers (Section 3) act to maximize a function of their interaction histories, called an *observation-utility function*. Any agent can be expressed as an *O*-maximizer (as we show in Section 3.1), so we are not able to derive any characteristic properties. Instead, we argue that designing agents as *O*-maximizers biases the designer towards certain types of design mistakes. See Appendix A for a more detailed explanation of how we analyze design approaches.

Value-learners (Section 4) are a response to the difficulties of the O -maximizer approach: instead of requiring a fully specified observation-utility function, we only assume that there exists *some* such function we would like to maximize, then design the agent to *learn* that observation-utility function from experience. We formalize a value-learner as an extension of AIXI and O -maximizers, then discuss the kinds of mistakes a value-learner’s designer could make (Section 4.1).

2 Notation

As in [3] and [4], agents interact with their environments cyclically: in cycle k , an agent acts with action y_k (from alphabet Y), then perceives observation x_k (from alphabet X), so the interaction history is a string $y_1x_1y_2x_2\dots y_kx_k$. Using standard string notations, an interaction history may also be written as $yx_{1:k}$ or $yx_{\leq k}$. As in [4], we make the simplifying assumption that an agent interacts with the world only through its actions and its observations, and that no side effects can reach inside the agent and alter its functionality or memory.¹ Given this, any agent A can be formalized as a function $y_k = A(yx_{<k})$.

We use Nick Hay’s method (given in [3]) for measuring the expected utility of an agent. First, we posit an “outcome”, an overall effect resulting from all of the agent’s interactions with the environment, denoted by r . Each r is a member of R , the full set of distinguishable outcomes. Viewed another way, R defines a partition of the set of all possible histories of the universe (which we will call “universes” for short), and each outcome represents an equivalence class of universes from that set. Second, we assign a real-valued utility to each outcome using an outcome-utility function $U : R \rightarrow \mathbb{R}$. Finally, a conditional probability distribution $P(r|yx_{\leq m})$, is used to calculate the probability of each outcome given an interaction history. Using these three parts, the expected utility of a particular interaction history is given by

$$\sum_r^R U(r)P(r|yx_{\leq m}) . \quad (1)$$

3 O -maximizers

Given these definitions, we can construct an agent that extends Hutter’s reward-maximizing agent, AIXI, into a general observation-utility-maximizing agent, or O -maximizer. This formalization will later be the basis for our extension to value-learning.

A brief review: AIXI assumes that each observation x_k has a real-valued reward part r_k (not to be confused with our notation for outcomes $r \in R$), and it acts to maximize the expected sum of future rewards, $r_k + r_{k+1} + \dots + r_m$, from

¹ Self-improvement, then, would not alter the basic algorithm the agent uses. It would let it build tools, extensions of itself, external computers, and improved copies of itself, though, so a limited interaction channel is a minor handicap.

the current time k up to a future time horizon m . To do this, AIXI examines each possible future interaction history $yx_{k:m}$; a series of observations x defines each future, and the y s in that interaction history are chosen by AIXI to maximize expected reward value. From each interaction history, AIXI extracts the sum of rewards and weights it by that future's *algorithmic probability*², $\xi(yx_{\leq m})$. Finally, a y_k is selected that maximizes the total of all future rewards:

$$y_k = \arg \max_{y_k} \sum_{x_k} \max_{y_{k+1}} \sum_{x_{k+1}} \dots \max_{y_m} \sum_{x_m} (r_k + \dots + r_m) \xi(yx_{\leq m}). \quad (2)$$

Replacing the sum of rewards ($r_k + \dots + r_m$) with Hay's method for deriving expected utility from Section 1, we give the formula for a general observation-utility-maximizer, or *O*-maximizer:

$$y_k = \arg \max_{y_k} \sum_{x_k} \max_{y_{k+1}} \sum_{x_{k+1}} \dots \max_{y_m} \sum_{x_m} \sum_r^R U(r) P(r|yx_{\leq m}) \xi(yx_{\leq m}). \quad (3)$$

Like AIXI, this agent acts to maximize an expected value, where possible values are weighted by their respective futures' algorithmic probabilities. Unlike AIXI, an *O*-maximizer determines the value of each interaction history by calculating the expected utility over outcomes arising from that history. R (the set of outcomes), U (the function from outcome to utility) and P (the distribution over outcomes given an interaction history) define the agent's utility function over its interactions, or *observation-utility function*.

3.1 The *O*-maximizer Design Approach

If an ultraintelligent *O*-maximizer was actually built, would it be likely to behave as desired and expected?

It would be convenient if we could show that all *O*-maximizers have some characteristic behavior pattern, as we do with reward maximizers in Appendix B. We cannot do this, though, because the set of *O*-maximizers coincides with the set of all agents; any agents can be written in *O*-maximizer form.

To prove this, consider an agent A whose behavior is specified by $y_k = A(yx_{<k})$. Trivially, we can construct an *O*-maximizer whose utility is 1 if each y_n in its interaction history is equal to $A(yx_{<n})$, and 0 otherwise. This *O*-maximizer will maximize its utility by behaving as A does at every time n . In this way, any agent can be rewritten as an *O*-maximizer.

Instead of proving that all *O*-maximizers behave a certain way, we examine *O*-maximizers from a design perspective using the technique described in Appendix A. An *O*-maximizer's designer must specify an R , U , and P , and the designer's goal is an *O*-maximizer that continues to behave desirably and as intended after an intelligence explosion. What conditions must hold on R , U , and P , and what kinds of mistakes could be made while following this approach?

² An understanding of algorithmic probability, Solomonoff induction, and ξ is not required for this paper; see [4] for details.

3.2 Design Conditions for an Ultraintelligent O -maximizer

When using the O -maximizer approach to designing an agent, a designer must make sure that correct utilities are assigned to interaction histories. If the agent does not do this, the designer cannot be sure that it will choose desirable actions by maximizing expected utility.

Condition 1 (Accuracy): *P must assign probabilities that reflect the degree to which evidence in each interaction history indicates each outcome.*

As a first step to finding an interaction history’s expected utility, the designer must specify a P that correctly judges the likelihood of each outcome given that interaction history.

Condition 2 (Fidelity): *No single outcome r in R can represent two universes that are different in any morally-relevant ways.*

The designer must also be sure that R is sufficiently nuanced to reflect the full scope of human preferences. As noted in Section 2, each outcome can be thought of as standing for a class of possible universes sharing a set of morally relevant features. If a set of outcomes is not designed to be expressive enough, it may use a single outcome to stand for universes with important moral differences, and the agent will not take these differences into account when it makes decisions.

Condition 3 (Correctness): *U must assign utilities to outcomes identically (up to linear transformations) to some “correct” outcome-utility function.*

Finally, the designer must specify how desirable each outcome is. Because the utilities are used in expected value computations, the agent’s actions under uncertainty are determined by the ordering and proportion between utilities of outcomes.

3.3 Comments on O -maximizer Design Conditions

The three conditions of Accuracy, Fidelity, and Correctness each correspond to one or more classes of mistakes that an O -maximizer’s designer could make.

Accuracy An Accurate P is a probability distribution capable of extrapolating all outcome-relevant state in the universe, both near and distant, from the agent’s interaction history. This would require the designer to encode the (potentially very many) outcome-relevant features of the universe in terms of a formal physics, linking them causally to the agent’s interaction mechanisms.

Mistakes that can be made in P ’s design are factual, not moral. For example, a P designed before Einstein would not correctly predict some outcomes from interaction histories, especially where near-light speeds are involved. Since an ultraintelligent agent would surpass the limits of our current technology during an intelligence explosion, it would likely encounter scenarios in which our current physics is a poor approximation, causing it to assign incorrect probabilities to

outcomes and therefore to make choices we wouldn't desire or intend. Unlike other factually incorrect knowledge, however, the agent would have no basis for correcting P , since P is (to the agent) the *definition* of how to derive outcomes from interaction histories.

Fidelity Unfortunately, we are not near a consensus on an exhaustive list of moral criteria. For example, in a 2009 survey, the opinions of Ph.D. students and faculty specializing in Normative Ethics were split nearly evenly between accepting or leaning towards deontology, consequentialism, virtue ethics, and "other" (lumping together answers like "undecided" and "the question is too unclear to answer" [1]). Additionally, it has been argued [11] that human values are complex and fragile; that is, there is not much reason to suspect that they can be simplified into a small, axiomatic core, and removing even less-important-seeming values such as boredom would be extremely damaging to human values as a whole.

Even worse, some moral criteria that will be important to people of the future may not yet be comprehensible to us, just as the concept of "free speech" is incomprehensible to dogs or chimpanzees. Any proposed exhaustive list of moral considerations would need very strong arguments to overcome the history of attempted systematizations, refinements, and revolutions that mark our philosophical progress on morality thus far.

Correctness The fact that we are not able here to give a more substantive description than "correct" reflects how little we know about our goals. Designs could fail correctness through mistakes of *ordering* or mistakes of *degree*.

A mistake of ordering occurs when one outcome is valued over another incorrectly; the "correct" utilities would reverse the preference or make the outcomes equally valued. Ordering mistakes are the most common reasons for an ethical theory to be found "incorrect", and most instances of moral progress are corrections of ordering mistakes.

A mistake of degree occurs when outcomes are preferred in correct order, but not in correct proportion relative to other outcomes' utilities. In these cases, the agent will make incorrect (relative to the "correct" preference ordering) decisions under uncertainty. Mistakes of degree highlight the precision required in the design of an outcome-utility function U . By giving utilities to outcomes, a designer not only specifies that outcome A is preferred to B and B is preferred to C , but also that there is a particular, correct probability p such that

$$pU(A) + (1 - p)U(C) = U(B) .$$

Proportions between utilities are crucial for an O -maximizer, since it chooses its actions to maximize expected probability using just such a calculation, multiplying the utilities of particular outcomes by their probabilities.

4 Value-Learners

The value-learning approach is intended to avoid the mistakes that can appear in a manually constructed observation-utility function. Instead of trying to design an observation-utility function, we assume that there is *some* desirable observation-utility function and we design an agent to learn it through experience. *By adopting a value-learning design approach, we intend to avoid explicitly programming human morality, instead passing that problem to an ultraintelligent agent that will learn our values from us.*

To formalize value-learning, we first assume that there is a correct observation-utility function in the form of an outcome set R , an outcome-utility function U , and a distribution P . Though we can't construct the observation-utility function directly, we must know something about how to learn it from observation. This is the single parameter a value-learner's designer must set: pre-programmed knowledge about how to learn an observation-utility function through interaction with the world. As in Hay's description of an agent that "learns what we want", the correct observation-utility function is presumed to be encoded in the environment somewhere, and an optimal value-learning agent should maximize expected utility according to this function in all possible environments [3].

We encode our knowledge about how to learn (R, U, P) from an interaction history as a probability distribution K . K is constructed such that, given an interaction history $yx_{\leq m}$, the probability of (R, U, P) being correct is $K(R, U, P|yx_{\leq m})$. Given this knowledge, the expected value of a particular interaction history is given by

$$\sum_{R,U,P} \sum_r^R U(r)P(r|yx_{\leq m})K(R, U, P|yx_{\leq m}). \quad (4)$$

Since a value-learner is uncertain about both outcomes and observation-utility functions, it calculates expected utility by supposing that a particular interaction history did in fact occur, extracting probabilities for each possible observation-utility function and outcome (from that function's R) using K , then taking a weighted average of all utilities $U(r)$ using the K -derived joint probability of that outcome and observation-utility function being correct.

Given this definition of expected utility according to K , we can use the basic structure of AIXI and O -maximizers to define a value-learning agent. At each cycle k , this agent maximizes expected utility over all possible future interactions, weighted by algorithmic probability:

$$y_k = \arg \max_{y_k} \sum_{x_k} \dots \max_{y_m} \sum_{x_m} \sum_{R,U,P} \sum_r^R U(r)P(r|yx_{\leq m})K(R, U, P|yx_{\leq m})\xi(yx_{\leq m}). \quad (5)$$

This formal value-learning agent captures the elements we wanted: a designer doesn't need to specify an explicit observation-utility function, instead specifying some prior knowledge (K) that lets the agent learn one from experience.

4.1 Design Conditions on Value-Learners

How well does value-learning serve as a design approach for an ultraintelligent agent? Are the design conditions easy to derive and fulfill, and what kinds of mistakes do they allow for?

Like O -maximizers, value-learners must be designed to assign correct expected utilities to interaction histories. Unlike O -maximizers, there is only one parameter, K , up to the designer. If K is designed correctly, then the agent will assign expected utilities to interaction histories correctly, and it will choose actions that maximize expected total utility.

Condition 1 (Veracity): *K must assign probabilities that reflect the degree to which evidence in each interaction history entails the “correctness” of each possible observation-utility function.*

4.2 Comments on Value-Learner Design Conditions

In order for a particular K to fulfill Veracity, its designer would need to encode anything constituting evidence about the “correctness” of observation-utility functions in terms of a formal physics, linking them causally to the agent’s interaction mechanisms. Many mistakes that can be made in this part of the design are factual rather than moral: flaws in our understanding of the way the world works could be introduced into K , and the agent would have no basis for correcting these flaws.

However, in terms of factual knowledge required, the designer of a value-learner appears to have an easier job than the designer of an O -maximizer does. An O -maximizer’s distribution P requires a formal encoding of causal links between the agent and every morally-relevant feature of the universe, and it would be surprising if nearly every part of the reachable universe did not turn out to be morally relevant. A distribution K , on the other hand, requires a formal encoding of causal links to evidence about observation-utility functions. It is philosophically plausible that this evidence can be found in the human brain or other close surroundings. K ’s factual-knowledge component, therefore, is likely to be a strict subset of P ’s, and therefore easier to construct.

Morally speaking, the designer would both need to know *what kinds* of things specify evidence about “correct” observation-utility functions and specify exactly *how much* different types of evidence ought to affect one’s beliefs about whether an observation-utility function is correct; mistakes on either count would keep the agent from choosing an observation-utility function correctly. Again, assuming that this information can be found in the human brain, this seems like an easier task than designing an exhaustive set of possible moral outcomes R or a correct outcome-utility function U . It is philosophically plausible that *it is easier to tell an agent how to **learn** correct values than it is to **directly specify** the correct values.*

4.3 Value-learners Interpreted as O -maximizers

Since all value-learners are agents, any value-learner can be rewritten as an O -maximizer. In fact, the form of a value-learner is so close to that of an O -maximizer that, given a value-learner's distribution K , we can derive the observation-utility function (R', U', P') of an O -maximizer that behaves identically to that value learner:

$$R' = \{(R, U, P, r) | r \in R\} \quad (6)$$

$$U' = U_{r'}(r_{r'}) \quad (7)$$

$$P'(r' | yx_{\leq m}) = P(r | yx_{\leq m})K(R, U, P | yx_{\leq m}) \quad (8)$$

(Substituting R' , U' , and P' into the definition of an O -maximizer expands the equation into the definition of a value-learner, proving that it will behave as the value-learner does.)

Intuitively, these definitions mean that a value-learner's "outcome" consists of an observation-utility function (learned from experience using K) and an outcome from its outcome set.

To find an outcome's utility, the value-learner applies the outcome-utility function it has learned to the outcome it has learned; for example, if it learned that some formal version of virtue ethics was correct, it would use that formal virtue ethics to evaluate how much virtue had been practiced over the history of the universe. The result would be the utility of the outcome.

Finally, P' gives the probability of a particular observation-utility function and outcome being correct. It does this by using K to find the probability of the observation-utility function being correct, then using that observation-utility function's P to find the probability of the outcome r . Multiplying these gives the probability of the outcome $(R, U, P, r) \in R'$, given the interaction history.

Examining a value-learner's O -maximizer form reveals that we can, in fact, use the philosophical assumptions that make value learning possible to design an O -maximizer that behaves as intended. By assuming that the "correct" observation-utility function can be learned from experience and that K meets the condition of Veracity, we are able to satisfy the three O -maximizer design conditions.

5 Conclusion

In this paper, we have examined a few design approaches for ultraintelligent agents. In each case, we found that a successful design requires the satisfaction of strict and difficult conditions on the design's parameters. Designing an O -maximizer directly would take significant amounts of information about how the universe works, what moral outcomes are possible, and how moral outcomes are valued with respect to one another. A value-learner's designer needs information only about how our local part of the universe works and how evidence about the correctness of ethical theories should be interpreted. It learns the moral and

factual content of an observation-utility function based on this foundation, and its success depends heavily on the philosophical assumption that this kind of learning is possible. Any value-learner can be expressed as an O -maximizer, but each approach allows for different kinds of design mistakes.

5.1 Future Work

This paper begins to develop formal tools for examining agent design as it relates to human value, especially in the realm of ultraintelligent agents. If the concepts explored here could be formalized precisely to the point where each argument in this paper could become a lemma, we might be able to build useful theorems stably on top of them instead of relying on intuitive arguments.

Acknowledgments. Thanks to Moshe Looks, Eliezer Yudkowsky, and Anna Salamon for their help and insight in developing the ideas presented here; thanks also to Anna Salamon, Moshe Looks, Dan Tasse, and Killian Czuba for their feedback and suggestions on the paper itself.

References

1. Bourget, D., Chalmers, D. The PhilPapers surveys: results, analysis and discussion. <http://philpapers.org/surveys>
2. Good, I. J.: Speculations Concerning the First Ultraintelligent Machine. In: F. L. Alt and M. Rubinfeld, (eds.) *Advances in Computers*, vol. 6, pp. 3188 (1965)
3. Hay, Nick: Optimal Agents. http://www.cs.auckland.ac.nz/~nickjhay/honours_revamped.pdf (2007)
4. Hutter, Marcus: Universal algorithmic intelligence: A mathematical top-down approach. In: *Artificial General Intelligence*, pages 227290. Springer, Berlin (2007)
5. Hutter, Marcus: <http://www.hutter1.net/ai/uaibook.htm#online>
6. Omohundro, S.: The Nature of Self-Improving Artificial Intelligence. http://omohundro.files.wordpress.com/2009/12/nature_of_self_improving_ai.pdf
7. Omohundro, S.: The basic AI drives, in Wang, P., Goertzel, B. and Franklin, S. (eds.) *Proceedings of the First AGI Conference*. Frontiers in Artificial Intelligence and Applications, Volume 171. IOS Press (2008)
8. Russell, S., Norvig, P.: *AI A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ (1995)
9. Yudkowsky, E.: Artificial intelligence as a positive and negative factor in global risk, in Bostrom, N. (ed.) *Global Catastrophic Risks*, Oxford: Oxford University Press (2008)
10. Yudkowsky, Eliezer: http://lesswrong.com/lw/ld/the_hidden_complexity_of_wishes
11. Yudkowsky, Eliezer: http://lesswrong.com/lw/y3/value_is_fragile/

A Design, Approach, and Conditions

In some cases, such as in Appendix B, one can argue that a particular way of designing ultraintelligent agents corresponds to a subset of agents that share a

predictable behavior pattern. Other design approaches don't admit such arguments, because they correspond to the set of all agents instead of a subset; they are ways of looking at the set of all agents and specifying a particular agent within that set. How can we argue for or against the appropriateness of these classes to ultraintelligent agent design? We will use the terms *design*, *approach*, and *conditions* in a novel technical sense to answer this question.

To **design** something is to set a number of free parameters with the aim of fulfilling some goal. An **approach** specifies the parameters that a design must set. For example, one might design a gear by specifying a series of Cartesian coordinates tracing its outline; another approach might use polar coordinates. Finally, **conditions** refer to the conditions that must be met by the parameters for a design to fulfill its goal. Conditions on a polar-coordinates approach to gear design, for example, will likely be simpler than conditions on a Cartesian coordinate approach. When evaluating approaches, it is useful to consider the conditions for each approach because *conditions define the kinds of mistakes that can be made in a design*, and we ought to prefer approaches that minimize our chance of making design mistakes.

B Ultraintelligent Reward Maximizers Behave Badly

Here, we argue briefly that any reward maximizer powerful enough to self-improve will eventually diverge from intended and desirable behavior. Therefore, reward maximization is not an appropriate design strategy for any potentially ultraintelligent agent.

As evident from the one-line definition of AIXI, a reward maximizer makes its choices with the sole concern of maximizing future rewards, so it will behave well only if it receives rewards for achieving desirable outcomes. In other words, rewards must be *outcome-sensitive*.

Rewards may be administered by a human, or an automatic reward system may detect particular outcomes (a won chess game, a high stock price, etc.) and give rewards accordingly. Whether a human or an artificial system, we will refer to the source of rewards as a *rewarder*. A properly set up rewarder, as long as it continues to function as intended, can ensure that rewards are outcome-sensitive.

From the maximizer's perspective, however, the rewarder is just a part of the outside world. As the maximizer learns and self-improves, it will gain the ability to alter the relationships that enforce outcome-sensitivity between the rewarder, actual outcomes, and the maximizer. Since outcome-sensitivity often prevents rewards (when a good outcome is not achieved), the maximizer will act to remove it. Therefore, any learning or self-improving reward maximizer will eventually diverge from intended and desirable actions as a natural result of the maximizer's intelligence overcoming and removing the rewarder's outcome-sensitivity.