Learning What to Value

Daniel Dewey

Machine Intelligence Research Institute

Abstract. We examine ultraintelligent reinforcement learning agents. Reinforcement learning can only be used in the real world to define agents whose goal is to maximize expected rewards, and since this goal does not match with human goals, AGIs based on reinforcement learning will often work at cross-purposes to us. We define value learners, agents that can be designed to learn and maximize any initially unknown utility function so long as we provide them with an idea of what constitutes evidence about that utility function.

1 Agents and Implementations

Traditional agents [2, 3] interact with their environments cyclically: in cycle k, an agent acts with action y_k, then perceives observation x_k. The interaction history of an agent with lifespan m is a string $y_1 x_1 y_2 x_2 ... y_m x_m$, also written $yx_{1:m}$ or $yx_{\leq m}$. Beyond these interactions, a traditional agent is isolated from its environment, so an agent can be formalized as an agent function from an interaction history $yx_{<k}$ to an action y_k.

Since we are concerned not with agents in the abstract, but with very powerful agents in the real world, we introduce the concept of an **agent implementation**. An agent implementation is a physical structure that, in the absence of interference from its environment, implements an agent function. In cycle k, an unaltered agent implementation executes its agent function on its recalled interaction history $yx_{<k}$, sends the resulting y_k into the environment as output, then receives and records an observation x_k. An agent implementation's behavior is only guaranteed to match its implemented function so long as effects from the environment do not destroy the agent or alter its functionality. In keeping with this realism, an agent implementation's *environment* is considered to be the real world in which we live. We may engineer some parts of the world to meet our specifications, but (breaking with some traditional agent formulations) we do not consider the environment to be completely under our control, to be defined as we wish.

Why would one want to study agent implementations? For narrowly-intelligent agents, the distinction between traditional agents and agent implementations may not be worth making. For ultraintelligent agents, the distinction is quite important: agent implementations offer us better predictions about how powerful agents will affect their environments and their own machinery, and are the basis for understanding real-world agents that model, defend, maintain, and improve themselves.

2   Reinforcement Learning

Reinforcement learning adds to the agent formalism the concept of *reward*, encoded as a scalar r_k in each observation x_k. The reinforcement learning problem is to define an agent that maximizes its total received rewards over the course of its interactions with its environment [3].

In order to think clearly about ultraintelligent reinforcement learning agent implementations, we make use of Hutter's AIXI [3, 4], an optimality notion that solves the reinforcement learning problem in a very general sense. AIXI's optimality means that the best an agent can do is to approximate a full search of all possible future interaction histories $yx_{k:m}$, find the probability of each history, and take the action with the highest expected total reward. A simplified version of this optimality notion, adapted for use with agent implementations, is given by

$$y_k = \arg\max_{y_k} \sum_{x_k y x_{k+1:m}} (r_k + \ldots + r_m) P(yx_{\leq m}|yx_{<k}y_k) . \tag{1}$$

Where AIXI makes its optimality claims using the knowledge that it will continue to maximize expected rewards in the future, incorporating a rigid self-model (an expectimax tree), an agent implementation has no such knowledge, as environmental effects may interfere with future decisions. The optimality notion given here models itself as part of the world, using induction to predict its own future decisions[1]. We have also omitted detail we will not require by simplifying AIXI's Solomonoff prior ξ to P, some "appropriate distribution" over interaction histories.

The trouble with reinforcement learning is that, in the real world, it can only be used to define agents whose goal is to maximize observed rewards. Consider a hypothetical agent implementation *AI-RL* that approximates (1). It is appealing to think that *AI-RL* "has no goal" and will learn its goal from its environment, but this is not strictly true. *AI-RL* may in some sense learn instrumental goals, but its final goal is to maximize expected rewards in any way it can. Since human goals are not naturally instrumental to maximized rewards, the burden is on us to engineer the environment to prevent *AI-RL* from receiving rewards except when human goals are fulfilled.

An AGI whose goals do not match ours is not desirable because it will work at cross-purposes to us in many cases. For example, *AI-RL* could benefit by altering its environment to give rewards regardless of whether human goals are achieved. This provides a strong incentive for *AI-RL* to free its rewards from their artificial dependence on fulfillment of human goals, which in turn creates a conflict of interest for us: increasing *AI-RL*'s intelligence makes *AI-RL* more effective at achieving our goals, but it may allow *AI-RL* to devise a way around its restrictions. Worse, increasing intelligence could trigger an intelligence explosion [1, 8] in which *AI-RL* repeatedly self-improves until it far surpasses our ability to contain it. Reinforcement learning is therefore not appropriate for a real-world AGI; the more intelligent a reinforcement learner becomes, the harder it is to

---

[1] Thanks to Peter de Blanc for this idea.

use to achieve human goals, because no amount of careful design can yield a reinforcement learner that works towards human goals when it is not forced to.

Self-rewarding has been compared to a human stimulating their own pleasure center, e.g. using drugs [3]. This metaphor is imperfect: while in humans, pleasure induces satiation and reduces activity, an agent governed by (1) that "hacks" its own rewards will not stop acting to maximize expected future rewards. It will continue to maintain and protect itself by acquiring free energy, space, time, and freedom from interference (as in [5]) in order to ensure that it will not be stopped from self-rewarding. Thus, an ultraintelligent self-rewarder could be highly detrimental to human interests.

3   Learning What to Value

In the following sections, we construct an optimality notion for implemented agents that can be designed to pursue any goal, and which can therefore be designed to treat human goals as final rather than instrumental goals. These agents are called **value learners** because, like reinforcement learners, they are flexible enough to be used even when a detailed specification of desired behavior is not known.

The trouble with the reinforcement learning notion (1) is that it can only prefer or disprefer future interaction histories on the basis of the rewards they contain. Reinforcement learning has no language in which to express alternative final goals, discarding all non-reward information contained in an interaction history. To solve this problem, our more expressive optimality notion replaces the sum of future rewards $(r_1 + \cdots + r_m)$ with some other evaluation of future interaction histories. First we will consider a fixed utility function, then we will generalize this notion to learn what to value.

Observation-Utility Maximizers. Our first candidate for a reward replacement is inspired by Nick Hay's work [2], and is called an observation-utility function. Let U be a function from an interaction history $yx_{\leq m}$ to a scalar utility. U calculates expected utility given an interaction history.

Observation-utility functions deserve a brief explanation. A properly-designed U uses all of the information in the interaction history $yx_{\leq m}$ to calculate the probabilities of different outcomes in the real world, then finds an expected utility by performing a probability-weighted average over the utilities of these outcomes. U must take into account the reliability of its sensors and be able to use local observations to infer distant events; it must also be able to distinguish between any outcomes with different values to humans, and assign proportional utilities to each.

Putting $U(yx_{\leq m})$ in place of the sum of rewards $(r_1 + \ldots + r_m)$ produces an optimality notion that chooses actions so as to maximize the expected utility given its future interaction history:

$$y_k = \arg\max_{y_k} \sum_{x_k y x_{k+1:m}} U(yx_{\leq m}) P(yx_{\leq m}|yx_{<k}y_k) \ . \tag{2}$$

Unlike reinforcement learning, expected observation-utility maximization can be used to define agents with many different final goals. Whereas reinforcement learners universally act to bring about interaction histories containing high rewards, an agent implementing (2) acts to bring about different futures depending upon its U. If U is designed appropriately, an expected utility maximizer could act so as to bring about any set of human goals. Unless we later decide that we don't want the goals specified by U to be fulfilled, we will not work at cross-purposes to such an agent, and increasing its intelligence will be in our best interest.

It is tempting to think that an observation-utility maximizer (let us call it $AI\text{-}OUM$) would be motivated, as $AI\text{-}RL$ was, to take control of its own utility function U. This is a misunderstanding of how $AI\text{-}OUM$ makes its decisions. According to (2), actions are chosen to maximize the expected utility given its future interaction history according the current utility function U, not according to whatever utility function it may have in the future. Though it *could* modify its future utility function, this modification is not likely to maximize U, and so will not be chosen. By similar argument, $AI\text{-}OUM$ will not "fool" its future self by modifying its memories.

Slightly trickier is the idea that $AI\text{-}OUM$ could act to modify its sensors to report favorable observations inaccurately. As noted above, a properly designed U takes into account the reliability of its sensors in providing information about the real world. If $AI\text{-}OUM$ tampers with its own sensors, evidence of this tampering will appear in the interaction history, leading U to consider observations unreliable with respect to outcomes in the real world; therefore, tampering with sensors will not produce high expected-utility interaction histories.

Value-Learning Agents. Though an observation-utility maximizer can in principle have any goal, it requires a detailed observation-utility function U up front. This is not ideal; a major benefit of reinforcement learning was that it seemed to allow us to apply an intelligent agent to a problem without clearly defining its goal beforehand. Can this idea of learning to maximize an initially unknown utility function be recovered?

To address this, we propose uncertainty over utility functions. Instead of providing an agent one utility function up front, we provide an agent with a pool of possible utility functions and a probability distribution P such that each utility function can be assigned probability $P(U|yx_{\leq m})$ given a particular interaction history. An agent can then calculate an expected value over possible utility functions given a particular interaction history: $\sum_U U(yx_{\leq m})P(U|yx_{\leq m})$.

This recovers the kind of learning what to value that was desired in reinforcement learning agents. In designing P, we specify what kinds of interactions constitute evidence about goals; unlike rewards from reinforcement learning, this evidence is not elevated to an end in and of itself, and so does not lead the agent to seek evidence of easy goals[2] instead of acting to fulfill the goals it has learned.

---

[2] As long as P obeys the axioms of probability, an agent cannot purposefully increase or decrease the probability of any possible utility function through its actions.

Replacing the reinforcement learner's sum of rewards with an expected utility over a pool of possible utility functions, we have an optimality notion for a value-learning agent:

$$y_k = \arg\max_{y_k} \sum_{x_k y x_{k+1:m}} P_1(yx_{\leq m}|yx_{<k}y_k) \sum_U U(yx_{\leq m}) P_2(U|yx_{\leq m}) \qquad (3)$$

A value-learning agent approximates a full search over all possible future interaction histories $yx_{k:m}$, finds the probability of each future interaction history, and takes the action with the highest expected value, calculated by a weighted average over the agent's pool of possible utility functions.

4   Conclusion

Hutter's introduction to AIXI [3] offers a compelling statement of the goals of AGI:

> Most, if not all known facets of intelligence can be formulated as goal-driven or, more precisely, as maximizing some utility function. It is, therefore, sufficient to study goal-driven AI... The goal of AI systems should be to be useful to humans. The problem is that, except for special cases, we know neither the utility function nor the environment in which the agent will operate in advance.

Reinforcement learning, we have argued, is not an adequate real-world solution to the problem of maximizing an initially unknown utility function. Reinforcement learners, by definition, act to maximize their expected observed rewards; they may learn that human goals are in some cases instrumentally useful to high rewards, but this dynamic is not tenable for agents of human or higher intelligence, especially considering the possibility of an intelligence explosion.

Value learning, on the other hand, is an example framework expressive enough to be used in agents with goals other than reward maximization. This framework is not a full design for a safe, ultraintelligent agent; at very least, the design of probability distributions and model pools for utility functions is crucial and non-trivial, and still better frameworks for ultraintelligent agents likely exist. Value learners do not solve all problems of ultraintelligent agent design, but do give a direction for future work on this topic.

References

1. Good, I. J.: Speculations Concerning the First Ultraintelligent Machine. In: F. L. Alt and M. Rubinoff, (eds.) *Advances in Computers*, vol. 6, pp. 31âĂŞ88 (1965)

2. Hay, Nick: Optimal Agents. `http://www.cs.auckland.ac.nz/~nickjhay/honours.revamped.pdf` (2007)
3. Hutter, Marcus: Universal algorithmic intelligence: A mathematical top-down approach. In: *Artiïņ̃cial General Intelligence*, pages 227-âĂŞ290. Springer, Berlin (2007)
4. Hutter, Marcus: `http://www.hutter1.net/ai/uaibook.htm#oneline`
5. Omohundro: The Nature of Self-Improving Artificial Intelligence. `http://omohundro.files.wordpress.com/2009/12/nature_of_self_improving_ai.pdf`
6. 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)
7. Russell, S., Norvig, P.: AI âĂŞ A Modern Approach. Prentice-Hall, Englewood Cliffs, NJ (1995)
8. 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)