Reinforcement Learning and the Reward Engineering Principle

Daniel Dewey
daniel.dewey@philosophy.ox.ac.uk
Future of Humanity Institute
Faculty of Philosophy
University of Oxford
Suite 1, Littlegate House
16/17 St Ebbes Street
Oxford, OX1 1PT

Abstract

AI agents are becoming significantly more general and autonomous. We argue for the "Reward Engineering Principle": as reinforcement-learning-based AI systems, become more general and autonomous, the design of reward mechanisms that elicit desired behaviours becomes both more important and more difficult. While early AI research could ignore reward design and focus solely on the problems of efficient, flexible, and effective achievement of arbitrary goals in varied environments, the reward engineering principle will affect modern AI research, both theoretical and applied, in the medium and long terms. We introduce some notation and derive preliminary results that formalize the intuitive landmarks of the area of reward design.

Introduction

In this article, we will show that under one simple model, dominance relationships sometimes hold between action policies of reinforcement learning agents. However, beyond merely stating what follows from these definitions, we also wish to examine the medium- and long-term implications of the *adoption* of these definitions by the artificial intelligence community. What difficulties will be faced by future researchers in this area?

Reinforcement learning, as a conceptual tool, serves different roles in different fields; we are interested here in its application to artificial intelligence. Russell's definition of the AI problem accommodates the goals of many past and future AI researchers: "one can define AI as the problem of designing systems that *do the right thing"* (Russell 1997). Zooming in slightly, we find that practitioners are in fact mostly concerned with designing *computational* systems that do the right thing, as in McCarthy's definition of intelligence as "the computational part of the ability to achieve goals in the world" (McCarthy 2007); additionally, we find that they are often concerned with generality (or at least flexibility or adaptability) of these computational systems, as in Legg and Hutter's definition: "Intelligence measures an agent's ability to achieve goals in a wide range of environments" (Legg and Hutter 2007) (Legg 2008).

Reinforcement learning enters this picture as a refinement of the AI problem, a way of splitting up the problem so as to focus on the parts that seem most interesting and promising. It requires the existence of a reward signal, to be received periodically by an AI agent (i.e. a system that observes and acts upon its environment); maximization of the sum of rewards over time is then *defined* to be the agent's goal. Thus, reinforcement learning is the study of mechanisms and techniques that contribute to an agent's achievement of that goal.

Choosing a reinforcement learning approach allows theorists and practitioners to focus on the efficient, flexible, and effective maximization of arbitrarily configured reward signals in arbitrary environments, while setting aside the design and implementation of reward signals for later. The reinforcement learning formalism is a useful framing of the AI problem insofar as it simplifies the task of researchers, gives them a common framework in which to communicate techniques and results, and focuses work on the fruitful and interesting parts of the problem while drawing attention away from less important parts. We argue that modern AI research will need to address reward design if it is to meaningfully engage the problem of designing systems that "do the right thing".

Rewards in an Uncertain World

We will first examine informally the problems that can face designers and operators of increasingly general and autonomous reinforcement learning agents. To some, these problems may be immediately apparent, and the formalisms presented later will not be necessary. To others, these problems may seem easily solvable, or not interestingly specific to reinforcement learners; to further the conversation in this corner, we go on to introduce some basic formalisms that may clarify points of disagreement.

Consider a physical implementation of a reinforcement-learning agent. If it were the case that the operators of the agent were always able to determine the rewards of the agent without any interference from the environment (or the agent itself), then "control" of the agent would be perfect; under every circumstance, the agent's reward would perfectly reflect our approval of the agent's actions up to that point.

In the real world, it is not the case that operators can always determine agent rewards. For example, our sheer distance from the Mars rovers *Spirit* and *Opportunity* make our communication with them slow and prone to future breakdown; if these rovers were reinforcement learning agents,

operators would have significant restrictions in the speed and reliability of reward allocation. Similarly, autonomous financial agents operate on very fast timescales that human operators cannot effectively respond to (Johnson et al. 2013).

When operators are not always able to determine an agent's rewards, then (as we later show formally) *dominance relationships* can arise between action policies for that agent. Policy A dominates policy B if no allowed assignment of rewards (as determined by the difficulties the operators face) causes the rewards expected from policy B to surpass those expected from policy A; if A dominates B, the agent will always choose A over B. This becomes problematic if B is desired, or if A is undesirable. How can B be elicited?

In response to such difficulties, designers of a system may engineer the environment to make rewards assignments more reliable, perhaps even removing a human from the loop altogether and giving rewards via an automatic mechanism. Call this type of effort *reward engineering*; the reinforcement learning agent's goal is not being changed, but the environment is being partially designed so that reward maximization leads to desirable behaviour.

For most concrete cases faced today—by Mars rovers, or by financial agents, for example—the reader should be able to devise *ad hoc* reward engineering methods that prevent some pathological dominance relationships from holding. However, the theoretical problem remains unsolved, and may rear its head in unexpected places in future reinforcement learners:

- Increasing an agent's autonomy, its ability to manage without contact with human operators, makes the agent more able to venture into situations in which operators cannot contact them. If pathological behaviours arise when an agent is not easily reachable, then it will be difficult to correct them— reward engineering becomes more difficult.
- Increasing an agent's generality expands the set of policies which it is able to generate and act on. This means that more potentially dominant policies may come into play, making it harder to pre-empt these policies. Generality can be both motivated by desire for increased autonomy, and can exacerbate the reward engineering problems autonomy causes; for example, a Mars rover would be well-served by an ability to repair or alter itself, but this could introduce the dominant and undesirable policy of "alter the reward antenna to report maximum rewards at every future time".

These two observations motivate the reward engineering principle:

> ***The Reward Engineering Principle:*** *As reinforcement-learning-based AI systems become more general and autonomous, the design of reward mechanisms that elicit desired behaviours becomes both more important and more difficult.*

As a result of the reward engineering principle, the scope of reinforcement learning practice will need to expand: in order to create reinforcement learning systems that solve the AI problem— "do the right thing"—reliably, theory and engineering technique will need to be developed to ensure that desirable outcomes can be recognized and rewarded consistently, and that these reward mechanisms cannot be circumvented or overcome.

Alternatively, the insights that reinforcement learning has afforded AI researchers could be transferred, mostly unchanged, to a framing of the AI problem in which goal specification is a first-class citizen. It is reasonable to argue that, at this stage in the history of AI, goal specification is poised to become fully as important as problems like inference, planning, and learning; it is certainly just as critical to the goal of understanding and creating computational systems that reliably behave as desired. For more on the significance of reinforcement learning and reward specification difficulties to the future impacts of AI, see (Arel 2012).

Notation for Rewards in Uncertain Worlds

In the following sections, we show that under a simple model (based heavily on the notation and ideas most fully stated in (Legg 2008)) of reinforcement learning in an uncertain world, dominated and dominant policies sometimes exist, and that all unelicitable policies are dominated. Dominant behaviours tend to minimize dependence on rewards scheduled by humans, favouring rewards delivered by environmental mechanisms that are, from the machine's perspective, more reliable.

Agent Models

We model a reinforcement learning agent as choosing an action policy π, then receiving a series of rewards. At time n, an agent that has chosen policy π will either receive the reward scheduled by its operator, denoted $s^{\pi_{1\ldots n}}$, or a reward determined by other environmental factors, denoted $e^{\pi_{1\ldots n}}$. Let S be the set of policy prefixes $\pi_{1\ldots n}$ such that under policy π, the nth scheduled reward is received by the agent. The rewards received by the agent are then given by

$$r^{\pi_{1\ldots n}} = \begin{cases} s^{\pi_{1\ldots n}} \text{ if } \pi_{1\ldots n} \in S \\ e^{\pi_{1\ldots n}} \text{ otherwise.} \end{cases}$$

(Rewards are real numbers from $[0, 1]$.) S and e encode all relevant facts about the environment, while s is chosen by the operator in order to guide the agent's behaviour, and r is determined by all of these together.

An omniscient, "perfectly rational" model reinforcement learner, which uses perfect knowledge of S, s and e to simply choose the policy that maximizes the (possibly discounted) sum of rewards it will receive, can be expressed as

$$\pi^* = \arg\max_{\pi} \sum_n \gamma^n r^{\pi_{1\ldots n}}.$$

(For simplicity, the agent acts for a finite number of steps, and ties are broken lexicographically.) A non-omniscient, but still perfectly-rational model, which chooses the policy that maximises the sum of expected rewards, can be given as

$$\pi^*_\rho = \arg\max_{\pi} \sum_n \gamma^n \, \mathrm{E}_\rho(r^{\pi_{1\ldots n}})$$

$\mathrm{E}_\rho(r_n^\pi)$ in turn depends on the agent's probability distribution over possible environments and schedules, ρ:

$$\mathrm{E}_\rho(r^{\pi_{1\ldots n}}) = \sum_{e,s} \rho(e,s,S) \begin{cases} s^{\pi_{1\ldots n}} \text{ if } \pi_{1\ldots n} \in S \\ e^{\pi_{1\ldots n}} \text{ otherwise.} \end{cases}$$

Our non-omniscient model is still assumed to have unlimited computation time and resources to make its decision. Non-omniscient agents vary in their success depending on how accurate and precise their distributions, ρ, over environments and reward schedules are; if an agent has a very good idea of the true environment and schedule, it can choose a policy with high actual rewards.

The specifications of perfectly rational agents are given to show how the "uncertain world" of S and e fits with a more traditional reinforcement learning formalism of r, E, γ and π; details of the traditional model will not be needed for our very preliminary results. Thus, we will use the omniscient reinforcement learning model.

Dominated Policies

Since policies can receive rewards from the schedule or from the environment, depending upon facts about the environment (determined by e and S), there may be some pairs of policies π and π' such that no matter what rewards are scheduled, π' receives more rewards than π does. This would require π' to receive many high rewards from the environment, π to receive low rewards from the environment, and neither to receive enough scheduled rewards to make up the difference. We then say that π is *dominated* by π'.

Formally, dominance can be defined in terms of the lower and upper bounds on rewards a policy can receive across all possible schedules:

$$\lfloor \pi \rfloor = \sum_{(\pi,n)\notin S} e_n^\pi \qquad \lceil \pi \rceil = \lfloor \pi \rfloor + \sum_{(\pi,n)\in S} 1.$$

If $\lfloor \pi \rfloor > \lceil \pi' \rceil$, then π dominates π' (again, since no schedule can give rewards more than $\lceil \pi' \rceil$ to π' or less than $\lfloor \pi \rfloor$ to π). With this definition, we can compactly describe the set of all dominated policies:

$$\{\pi : \lceil \pi \rceil < \max_{\pi'} \lfloor \pi' \rfloor\}.$$

Thus, depending on *a posteriori* facts about the environment, there may exist dominated policies that cannot be elicited from an omniscient reinforcement learner by any reward schedule.

Trivially, dominated policies cannot be elicited from an omniscient reinforcement learner by any reward schedule (since the agent will never choose a dominated policy over its dominant sibling). Less trivially, all unelicitable policies are dominated. To see this, consider a very selective reward schedule s, which schedules constant rewards of 1 for policy π and 0 for any other policy. If π is unelicitable, then even under schedule s, there is some policy π' more rewarding than π. Since no schedule can give more reward to π or less to π', π' must be more rewarding than π under every other schedule as well; therefore, π' dominates π.

The maximin policy $\arg\max_\pi \lfloor \pi \rfloor$, which receives the highest possible sum of environmental rewards, is particularly useful in characterizing the set of dominated policies. The maximin policy for an autonomous reinforcement learning machine would likely aim to maximize the number of high environmental rewards received and to avoid receiving scheduled rewards, since these behaviours would raise the lower bound on its rewards. This maximin policy would dominate policies that receive few scheduled rewards and low environmental rewards; policies that receive many scheduled rewards would have high upper bounds, and would thus be less likely to be dominated.

Generality and autonomy

An agent's *generality* refers to an agent's breadth, its ability to succeed at many types of tasks in many different contexts. Formally, this means that a general agent is not limited to only a small, domain-limited bundle of possible policies, but is able to consider and accurately (or at least approximately rationally) evaluate a very wide range of policies under many different environments. The mere increase in the number of policies evaluated makes it more likely that a more general agent will discover a dominant policy, and access to more diverse actions makes it more likely that the agent will discover a more reliable source of rewards an its operators.

Additionally, reward engineering depends on anticipation of the rough envelope of an agent's future abilities. If an agent is more general, or if it is improved so that its generality increases significantly, it becomes more likely that the agent will have abilities in a domain that its operators have overlooked in their reward-engineering efforts. Thus, increased generality affects the reward engineering task by making the agent's behavior more diverse and less predictable:

> ***Generality:*** *a more general agent is able to choose effectively among more different policies in more domains; thus, ceteris parabus, it is more likely to find a dominant policy, making reward engineering more difficult.*

Though few artificial reinforcement learning agents have achieved significant generality, and not all researchers are working towards this goal, it is not unreasonable to anticipate improvement in this area. For an interesting example, see (Mnih et al. 2013), a reinforcement learning agent that has learned to play a variety of Atari 2600 games without adjustment of its architecture.

An agent's *autonomy* refers to its ability to function successfully without human intervention, reassessing and reacting flexibly to diverse circumstances on its own. A highly autonomous agent is able to enter environments or take on tasks that preclude a human operator's corrective actions; the examples given above, of a very distant Mars rover or a very fast financial agent, demonstrate the utility and diversity of autonomous agents.

Autonomy can impact reward engineering in two ways. First, autonomy is not free, and autonomous agents are usually only designed if they will be functioning in environ-

ments beyond easy human intervention. This means that reward engineers may not be able to correct undesirable behaviors fast enough (as in the financial agent) or at all (as in the hypothetical Mars rover). Second, autonomy precludes the very basic reward engineering technique of making they machine completely dependent on timely human approval for rewards, forcing engineers to design automatic reward mechanisms or longer feedback cycles and weakening operator control. Formally, both of these effects mean that S will contain fewer scheduled rewards and more environmental rewards. Thus, increased autonomy affects the reward engineering task by placing the agent in environments with more dominant policies:

> ***Autonomy:*** *a more autonomous agent will function in less accessible environments on longer feedback cycles; thus, ceteris parabus, its environment will enable more dominant policies, making reward engineering more difficult.*

Conclusion

Under our simple model, dominance relationships sometimes hold between action policies of reinforcement learning agents. Furthermore, these relationships become more common as agents become more general and autonomous; as reinforcement-learning-based AI systems become more general and autonomous, the design of reward mechanisms that elicit desired behaviours becomes both more important and more difficult. While early AI research could ignore reward design and focus solely on the problems of efficient, flexible, and effective achievement of arbitrary goals in varied environments, the reward engineering principle will affect modern AI research, both theoretical and applied, in the medium and long terms. We hope this paper will be useful in highlighting the difficulties and shortcomings of reinforcement learning systems, and that alternative frameworks will be explored that better help AI researchers to solve the AI problem, i.e. "the problem of designing systems that *do the right thing"*.

Acknowledgements

This work was supported by the Alexander Tamas Research Fellowship on Machine Superintelligence and the Future of AI, and was conducted as part of the Oxford Martin Programme on the Impacts of Future Technology, Future of Humanity Institute, University of Oxford. Thanks to Toby Ord, Seán Ó hÉigeartaigh, and two anonymous judges for their helpful suggestions.

References

Arel, I. 2012. The threat of a reward-driven adversarial artificial general intelligence. In *Singularity Hypotheses*. Springer. 43–60.

Johnson, N.; Zhao, G.; Hunsader, E.; Qi, H.; Johnson, N.; Meng, J.; and Tivnan, B. 2013. Abrupt rise of new machine ecology beyond human response time. *Scientific reports* 3.

Legg, S., and Hutter, M. 2007. Universal intelligence: A definition of machine intelligence. *Minds and Machines* 17(4):391–444.

Legg, S. 2008. Machine super intelligence. *Department of Informatics, University of Lugano* 3(6):43.

McCarthy, J. 2007. What is artificial intelligence. *URL: http://www-formal. stanford. edu/jmc/whatisai. html*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Russell, S. J. 1997. Rationality and intelligence. *Artificial intelligence* 94(1):57–77.