Unifying Curious Reinforcement Learners

Nadia M. Ady
Dept. of Computing Science
University of Alberta
Edmonton, AB, Canada
nmady@ualberta.ca

Patrick M. Pilarski
Dept. of Medicine &
Dept. of Computing Science
University of Alberta
Edmonton, AB, Canada
pilarski@ualberta.ca

Abstract
We suggest that one way to improve our understanding of curiosity is through the behaviour of computationally curious agents. By manipulating the computational curiosity method motivating an agent's actions and repeatedly placing that agent in a simple, consistent domain, we create a window into how different computational curiosity methods result in different behaviours. In particular, we suggest that reinforcement learning is a natural place to begin the principled study of computationally curious behaviours.

Author Keywords
computational curiosity; reinforcement learning; artificial intelligence.

ACM Classification Keywords
[Machine Learning]: Reinforcement Learning

A Principled Study of Curious Behaviours
Computational curiosity refers to mechanisms to give computational systems a desire to learn or know more, but it can also be thought of as building computational models of the abstract concept we call curiosity. One way we might better understand curiosity is to better understand the ways we are inclined to model it.

As curiosity *motivates* many human decisions, reinforcement learning (RL) seems like a strong contender in the pursuit of computational curiosity. RL is a well-studied way for biological systems and machines to learn about the

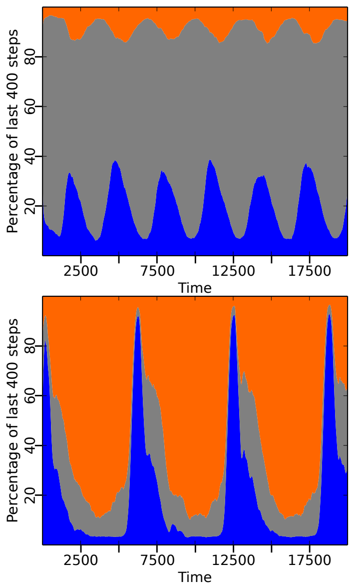

Figure 1: These two plots show different behavioural trajectories for an agent placed in the same domain using two different methods for computational curiosity. The upper method was suggested by Schmidhuber [2] and the second was suggested by Schembri *et al.* [1]. Orange, grey, and blue represent three different choices with varying outcomes.

value of situations and choices through trial and error. In RL problems, a signal known as *reward* is delivered to the learner during its interactions with an environment.

Reward can be used to provide a type of motivation for computational systems. Existing RL algorithms can be used to efficiently learn which actions maximize future reward. By designing our reward, we can therefore design different motivations for our systems. Researchers have developed different methods to modify the reward delivered to a learner or to modify other parts of an RL algorithm so as to evoke curious behaviours in their systems. Many of their methods have shown promise in real-world or simulated domains.

However, at present, there is no unified way to compare different curiosity methods. In both humans and machines, curiosity is challenging to measure due to its the variety and the individuality of the ways it is exhibited. One way we would like to be able to compare different curiosity methods is in how they impact agent behaviour.

We can create a level playing field by designing agents whose curiosity methods can be varied while their other inner workings are kept constant. Repeatedly placing such an agent in a single domain while varying its curiosity method can allow us to clearly see where its behaviour differs.

To gain an understanding of how the resulting behaviours compare to what we might expect or desire given the methods' theoretical underpinnings, we suggest that initial experiments should be run in uncomplicated simulated domains, with variations specifically chosen to untangle the differences between curiosity methods.

Figure 1 is included to show that even in a simple domain, two different curiosity methods result in two completely different behavioural trajectories. The experiment from which the plots are drawn was run with a simple, single-state domain with three actions in order to tease apart the ways that different methods might be affected by variation in reward.

In conclusion, we believe that the principled understanding of computational curiosity will make significant contributions to our understanding of curiosity as a whole, and to the development of general machine intelligence. Not only can curiosity benefit computational systems, allowing them to learn more effectively about non-stationary, specialized environments while using a general learning algorithm, but they could also be used to pique human curiosity, making things more engaging for the users of a wide range of computation-enabled technologies (e.g., personalized physiotherapy or fitness training). We can *design* for curiosity by *modelling* curiosity.

Author Biography

Nadia Ady is a doctoral student at the University of Alberta. Following an undergraduate degree in mathematics focused on algorithmic graph theory, combining interests in computers and dance lead her to study approaches for artificial creativity in choreography, then pursue reinforcement learning methods for computational curiosity.

References

[1] Massimiliano Schembri, Marco Mirolli, and Gianluca Baldassarre. 2007. Evolution and learning in an intrinsically motivated reinforcement learning robot. In *European Conference on Artificial Life*. Springer, 294–303.

[2] Jürgen Schmidhuber. 1991. A possibility for implementing curiosity and boredom in model-building neural controllers. In *From animals to animats: Proceedings of the first international conference on simulation of adaptive behavior*. 15–21.