Generating Interpretable Reinforcement Learning Policies using Genetic Programming

Daniel Hein*
Siemens AG, Corporate Technology,
Munich, Germany
hein.daniel@siemens.com

Steffen Udluft
Siemens AG, Corporate Technology,
Munich, Germany
steffen.udluft@siemens.com

Thomas A. Runkler
Siemens AG, Corporate Technology,
Munich, Germany
thomas.runkler@siemens.com

ABSTRACT

The search for interpretable reinforcement learning policies is of high academic and industrial interest. Especially for industrial systems, domain experts are more likely to deploy autonomously learned controllers if they are understandable and convenient to evaluate. Basic algebraic equations are supposed to meet these requirements, as long as they are restricted to an adequate complexity. In our recent work "Interpretable policies for reinforcement learning by genetic programming" published in Engineering Applications of Artificial Intelligence 76 (2018), we introduced the *genetic programming for reinforcement learning* (GPRL) approach. GPRL uses model-based batch reinforcement learning and genetic programming and autonomously learns policy equations from pre-existing default state-action trajectory samples. Experiments on three reinforcement learning benchmarks demonstrate that GPRL can produce human-interpretable policies of high control performance.

CCS CONCEPTS

• **Theory of computation → Reinforcement learning**; • **Software and its engineering → Genetic programming**.

KEYWORDS

Interpretable reinforcement learning, genetic programming, model-based, symbolic regression

ACM Reference Format:
Daniel Hein, Steffen Udluft, and Thomas A. Runkler. 2019. Generating Interpretable Reinforcement Learning Policies using Genetic Programming. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3326755

1 INTRODUCTION

Although the search of interpretable and understandable reinforcement learning (RL) policies is of high academic and industrial interest, little has been published in that direction [3]. The novel *genetic programming for reinforcement learning* (GPRL) approach, as introduced in [2], learns policy representations based on basic algebraic equations of low complexity.

Inspired by behaviorist psychology, RL is concerned with how software agents ought to take actions in an environment in order to maximize their received accumulated rewards. In RL, the acting agent is not explicitly told which actions to implement. Instead, the agent must learn the best action strategy from the observed environment's rewards in response to the agent's actions. Generally, such actions affect both the next reward and subsequent rewards. [4]

In batch RL, we consider applications where online learning approaches, such as classical temporal-difference learning, are prohibited for safety reasons, since these approaches require exploration of system dynamics. In contrast, batch RL algorithms generate a policy based on existing data and deploy this policy to the system after training. In this setting, either the value function or the system dynamics are trained using historic operational data.

In the proposed GPRL approach, the performance of a population of basic algebraic equations is evaluated by testing the individuals on a world model using the Monte Carlo method [4]. The combined return value of a number of action sequences is the fitness value that is maximized iteratively from genetic programming (GP) generation to generation. To the best of our knowledge, GP-generated policies have never been combined with such a model-based batch RL approach. To benchmark GPRL, we compare the obtained results to an alternative approach in which GP is used to mimic an existing non-interpretable neural network (NN) policy by symbolic regression.

2 GENETIC PROGRAMMING REINFORCEMENT LEARNING

The basis for GPRL is a data set $\mathcal{D}$ that contains state transition samples $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$ gathered from the dynamics of a real system, where, in state $\mathbf{s}$, action $\mathbf{a}$ was applied and resulted in state transition to $\mathbf{s}'$ with a real value reward r. Note that generally $\mathcal{D}$ can be generated using any (even a random) policy prior to policy training as long as sufficient exploration is involved [4]. Data set $\mathcal{D}$ is used to generate world models $\tilde{g}$ with inputs $(\mathbf{s}, \mathbf{a})$ to predict $(\mathbf{s}', r)$.

To rate the quality of each policy candidate π a fitness value has to be provided for the GP algorithm to advance. In GPRL, the fitness $\tilde{\mathcal{F}}$ of each individual is calculated by generating trajectories using the world model $\tilde{g}$ starting from a fixed set of initial benchmark states and aggregating the expected reward predicted by $\tilde{r}$.

The performance of GPRL is compared to a rather straightforward approach, which utilizes GP to conduct symbolic regression on a data set $\hat{\mathcal{D}}$ generated by a well-performing but non-interpretable RL policy $\hat{\pi}$. $\hat{\mathcal{D}}$ contains tuples $(\mathbf{s}, \hat{\mathbf{a}})$, where $\hat{\mathbf{a}}$ are the generated actions of policy $\hat{\pi}$ on state $\mathbf{s}$. The states originate from trajectories

*Also with Technical University of Munich, Department of Informatics.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326755

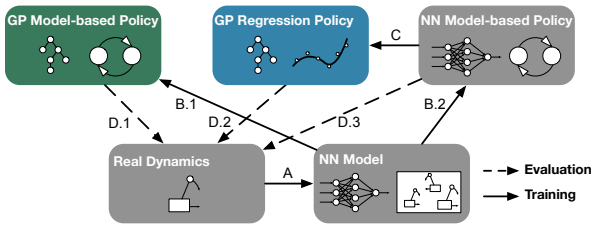

Figure 1: GPRL and its integration in the experimental setup

created by policy $\hat{\pi}$ on world model $\tilde{g}$. One might think that, given an adequate policy of any form and using GP to mimic this policy by means of some regression error with respect to $\hat{\mathbf{a}}$, could also yield successful interpretable RL policies. However, our experimental results clearly indicate that this strategy is only successful for rather small and simple problems and produces highly non-stable and unsatisfactory results for more complex tasks.

Figure 1 gives an overview of the relationships between the different policy implementations and the environment instances used for training and evaluation. The world model is the result of supervised ML regression on data originating from the real dynamics (A). GPRL generates a GP model-based policy by training on this world model (B.1), which is an NN model in our experimental setup. Similarly, the NN policy is trained in a model-based manner by utilizing the same NN model (B.2). Whereas the GP regression policy mimics the already existing NN policy by learning to minimize an error with respect to the existing policy's action (C). All of the policies are finally evaluated by comparing their performance on the real dynamics (D.1-D.3).

3 EXPERIMENTS & RESULTS

To evaluate the policies created by GPRL in terms of performance and interpretability, we conducted experiments on three RL benchmarks, i.e., the two well-known toy benchmarks mountain car and cart-pole balancing (CP), and a highly stochastic and multi-dimensional testbed called industrial benchmark [1].

Fig. 2 depicts all individuals of the Pareto fronts of three out of the ten CP experiment runs from complexity 5 to 11. Note how the solutions agree not only on the utilized state variables but also on the floating-point values of the respective factors. Provided with such a policy Pareto chart, experts are more likely to succeed in selecting interpretable policies, since common policy concepts are conveniently identifiable with respect to both, their complexity as well as their model-based performance.

To evaluate the true performance of the two approaches, GP model-based training and GP regression training, the individuals of both sets of Pareto fronts have been tested on the true CP dynamics. Fig. 3 shows the resulting *altered* Pareto fronts compared to the performance of the NN policy. Depicted are the median (green or blue lines) together with the minimum and maximum (semi-transparent green or blue areas) Pareto front penalties. The dashed line depicts the penalty baseline of the NN policy. It is evident that almost for every complexity that the GP model-based approach GPRL is superior to the GP regression idea. Interestingly, the median GP results for complexity 11 and above even outperformed the NN

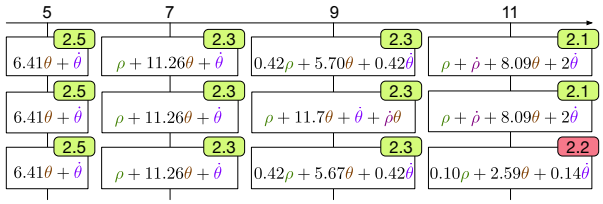

Figure 2: Interpretable CP policies for complexities 5-12

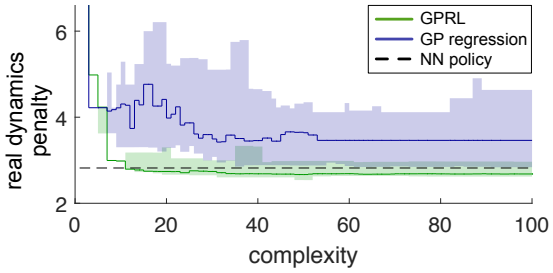

Figure 3: *Altered* Pareto fronts from the evaluation of GPRL and GP regression experiments on the real CP dynamics

policy result. This indicates that the NN policy already over-fitted the surrogate model and exploited its inaccuracies, while the simple GP policy equations generalize better because of their somewhat restricted structure.

Further extensive experimental studies are presented in more detail in [2].

4 CONCLUSION

Our GPRL approach conducts model-based batch RL in order to learn interpretable policies for control problems on the basis of already existing default system trajectories. The policies can be represented by compact algebraic equations or Boolean logic terms. The complete GPRL procedure of (i) training a model from existing system trajectories, (ii) learning interpretable policies by GP, (iii) selecting a favorable solution candidate from a Pareto front has been evaluated for three RL benchmarks.

The GPRL approach has been compared to a straightforward GP utilization as a symbolic regression tool, i.e., fitting the existing non-interpretable NN policy by GP to yield interpretable policies of similar control performance. All of our experiments showed that this strategy is significantly less suitable to produce policies of adequate performance.

REFERENCES

[1] D. Hein, S. Depeweg, M. Tokic, S. Udluft, A. Hentschel, T.A. Runkler, and V. Sterzing. 2017. A benchmark environment motivated by industrial control problems. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–8.

[2] D. Hein, S. Udluft, and T.A. Runkler. 2018. Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence* 76 (2018), 158–169.

[3] F. Maes, R. Fonteneau, L. Wehenkel, and D. Ernst. 2012. Policy search in a space of simple closed-form formulas: Towards interpretability of reinforcement learning. *Discovery Science* (2012), 37–50.

[4] R.S. Sutton and A.G. Barto. 1998. *Reinforcement learning: An introduction*. A Bradford book.