SIBRE: Self Improvement Based REwards for Adaptive Feedback in Reinforcement Learning

Extended Abstract

Somjit Nath
TCS Research
Mumbai, India
somjit.nath@tcs.com

Richa Verma
TCS Research
Delhi, India
richa.verma4@tcs.com

Abhik Ray
BITS-Pilani (Goa)
abhik09@gmail.com

Harshad Khadilkar
TCS Research
Mumbai, India
harshad.khadilkar@tcs.com

ABSTRACT

We propose a generic reward shaping approach for improving the rate of convergence in reinforcement learning (RL), called **S**elf **I**mprovement **B**ased **RE**wards, or **SIBRE**. The approach is designed for use in conjunction with any existing RL algorithm, and consists of rewarding improvement over the agent's own past performance. We prove that SIBRE converges in expectation under the same conditions as the original RL algorithm. The reshaped rewards help discriminate between policies when the original rewards are weakly discriminated or sparse. Experiments on several well-known benchmark environments with different RL algorithms show that SIBRE converges to the optimal policy faster and more stably. We also perform sensitivity analysis with respect to hyper-parameters, in comparison with baseline RL algorithms.

KEYWORDS

Reinforcement Learning; Reward Shaping; Adaptive Feedback

ACM Reference Format:
Somjit Nath, Richa Verma, Abhik Ray, and Harshad Khadilkar. 2021. SIBRE: Self Improvement Based REwards for Adaptive Feedback in Reinforcement Learning: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

Reinforcement learning (RL) is useful for solving sequential decision-making problems in complex environments. Value-based [4, 12], actor-critic and its extensions [8, 9], and Monte-Carlo methods [2] have been shown to match or exceed human performance in games. However, the training effort required for these algorithms tends to be high [3, 7, 10], especially in environments with complex state-action spaces. In this paper, we propose a modification to the reward function (called **SIBRE**, short for Self Improvement Based REward) that aims to improve the rate of learning in episodic environments and thus addresses the problem of sample efficiency through reward shaping. SIBRE is a threshold-based reward for RL algorithms, which provides a positive reward when the agent improves on its past performance, and negative reward otherwise. We observe that this accelerates learning without requiring computationally expensive estimation of baselines [1]. Furthermore, SIBRE can be used in conjunction with any standard RL algorithm: value or policy based, online or offline.

Algorithm 1: Illustration of SIBRE using Q-learning as example

```
Algorithm parameters: step size $\alpha \in (0, 1]$, $\epsilon > 0$, $\beta \in (0, 1)$;
Threshold Update after x episodes;
Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, $\rho = 0$;
foreach episode do
    Initialize S;
    $G = 0$;
    foreach step of episode do
        Choose $A$ from $S$ using policy derived from $Q$ (e.g., ε-greedy);
        Take action $A$, observe $R, S'$;
        $G = G + R$;
        if $S \in$ terminal then
            $R = G - \rho$;
            if episodecount mod $x = 0$ then
                $\rho \leftarrow (1 - \beta)\rho + \beta G$;
            end if
        end if
        $Q(S, A) \leftarrow (1 - \alpha)Q(S, A) + \alpha[R + \gamma \max_a Q(S', a)]$;
        $S \leftarrow S'$;
    end foreach
end foreach
```

Literature on formal reward shaping: Prior literature has shown that the optimal policy learnt by RL remains invariant under reward shaping if the modification can be expressed as a potential function [6]. While the concept is valuable, designing a potential function for each problem could be a difficult task. While SIBRE solves the same problem, the key differences from other well-known reward shaping approaches are (details in [5]),

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

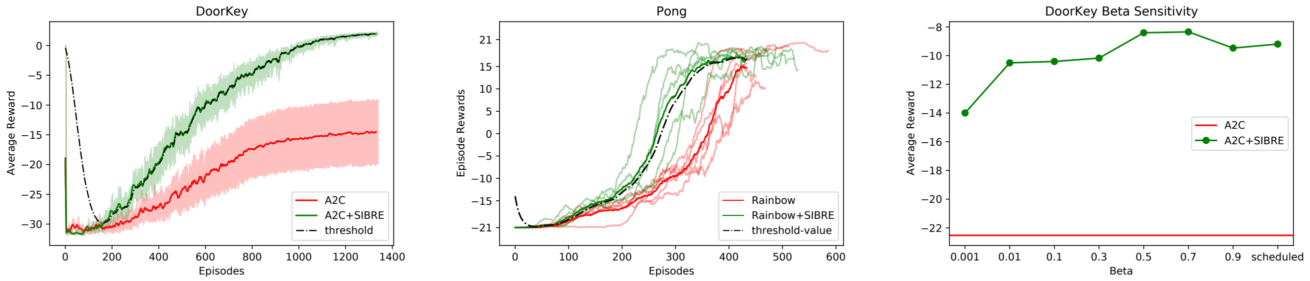

Figure 1: Learning curves across (a) 10 runs on 6x6 DoorKey (b) 5 runs on Pong (c) Beta sensitivity on 6x6 DoorKey

- The reward modification is computationally light (simple average) and can be used to improve the sample efficiency of any RL algorithm.
- SIBRE converges in expectation to the same policy as the original algorithm.
- We empirically observe faster convergence with lower variance on a variety of benchmark environments, with multiple RL algorithms.

2 DESCRIPTION OF SIBRE & RESULTS

Consider an episodic Markov Decision Process (MDP) specified by the tuple $< \mathcal{S}, \mathcal{A}, \mathcal{R}, P >$ [11], where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the set of possible rewards, and P is the transition function. We assume the existence of a reinforcement learning algorithm for learning the optimal mapping $\mathcal{S} \rightarrow \mathcal{A}$. It follows that the value of optimal reward depends on both the values of the step and terminal rewards, as well as on the size of the grid. In this paper, we retain the original step rewards R_k for time step k within the episode, but replace the terminal reward for episode t by a baseline-differenced value of the total return $G_t : \mathcal{S}, \mathcal{A}, \mathcal{S} \rightarrow \mathbb{R}$:

$$r_{k,t}(s_k, a_k, s_{k+1}) = \begin{cases} G_t - \rho_t, & s_{k+1} \in \mathcal{T} \\ R_k, & \text{otherwise} \end{cases} \tag{1}$$

where k is the step within an episode, t is the number of the episode, $\mathcal{T}$ is the set of terminal states, G_t is the return for episode t, and ρ_t is the performance threshold at episode t. Note that the return G_t is based on the original reward structure of the MDP. If the original step reward at k is R_k, then $G_t = \sum R_k$. The net effect of SIBRE is to provide a positive terminal reward, if $G_t \geq \rho_t$ and negative otherwise, which gives the notion of self-improvement. Also, we assume that a number x of episodes is run after every threshold update, allowing the q-values to converge with respect to the latest threshold value. Note that x can be a different number from one update to another. This assumption is necessary for proving that this modification to the rewards does not affect convergence to the optimal policy which we prove in [5]. Once the q-values have converged, the threshold can be updated using the relation,

$$\rho_{t+1} = \begin{cases} \rho_t + \beta_t (\sum_{y=t-x+1}^{t} \frac{G_y}{x} - \rho_t) & \text{if updating q-values} \\ \rho_t & \text{otherwise} \end{cases},$$

where $\beta_t \in (0, 1)$ is the step size and is assumed externally defined according to a fixed schedule. For all our experiments we used x=1 and a scheduled beta whereby we start with lower β and gradually increase the weight on the return, basically increasing the value of β. The entire algorithm is described in Algorithm 1.

Results: The learning curves are shown in Fig. 1 (a) and (b) on DoorKey and Pong. In both these cases, we can see how integration of SIBRE can help in accelerating learning and thus faster convergence. We also show the parameter sensitivity curves with respect to the introduced parameter β in Fig. 1 (c) where β-schedule starts with 0.001 increased linearly to 0.1 after every 10% of total episodes. As seen from the figure, SIBRE is robust to varying values of β.

SIBRE learns the value of a threshold which it aims to beat after each episode. We believe that once it has learnt the threshold properly, we get optimal performance. When we use the same model to learn on a bigger state-space with same reward structure, the value of the threshold provides a high initial value to beat and this helps in easy transfer of learning. Also, after the transfer, the value of the threshold can also yield information about possible negative transfer across environments. In Fig. 2 we do see such improvement while transferring from 5x5 to 8x8 grid in Doorkey.

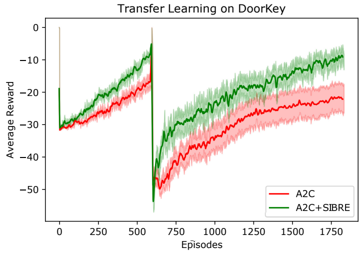

Figure 2: Transfer learning from 5x5 to 8x8 grid in DoorKey

Experiments on a variety of other domains, further hyper-parameter analysis and extension to continuing MDPs along with the exact hyper-parameters for reproduction of such results are presented in [5].

Key Takeaways: In this work, we showed that an adaptive, self-improvement based modification to the terminal reward (SIBRE) has empirically better performance, both qualitative and quantitative, than the original RL algorithms on a variety of environments. We were able to prove,analytically, that SIBRE converges to the same policy in expectation,as the original algorithms.

REFERENCES

[1] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research* 5, Nov (2004), 1471–1530.

[2] Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard L Lewis, and Xiaoshi Wang. 2014. Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning. In *Advances in neural information processing systems*. 3338–3346.

[3] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep RL. In *International conference on machine learning*. 1928–1937.

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep RL. *Nature* 518, 7540 (2015), 529.

[5] Somjit Nath, Richa Verma, Abhik Ray, and Harshad Khadilkar. 2020. SIBRE: Self Improvement Based REwards for Adaptive Feedback in Reinforcement Learning. arXiv:2004.09846 [cs.LG]

[6] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, Vol. 99. 278–287.

[7] Jakub Pachocki, Greg Brockman, Jonathan Raiman, Susan Zhang, Henrique Pondé, Jie Tang, Filip Wolski, Christy Dennison, Rafal Jozefowicz, Przemyslaw Debiak, et al. 2018. OpenAI Five, 2018. *URL https://blog. openai. com/openai-five* (2018).

[8] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2015. Trust Region Policy Optimization. *CoRR* abs/1502.05477 (2015). arXiv:1502.05477 http://arxiv.org/abs/1502.05477

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347

[10] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.

[11] Richard S. Sutton and Andrew G. Barto. 2018. *Introduction to Reinforcement Learning* (2nd ed.). MIT Press, Cambridge, MA, USA.

[12] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep RL with Double Q-Learning.. In *AAAI*, Vol. 2. Phoenix, AZ, 5.