MOL2NET, **2016**, Vol. 2, J; http://sciforum.net/conference/MOL2NET-02/SUIWML-01

SciForum
MOL2NET

Efficient Actor-critic Algorithm with Dual Piecewise Model Learning

Shan Zhong[1,2,3], Quan Liu[1,4,5], Qiming Fu[1,3,5,6], Peng Zhang[1], Weisheng Qian[1]

1 School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu, 215000, China
2 School of Computer Science and Engineering, Changshu Institute of Technology, Changshu, Jiangsu, 215500, China
3 Jiangsu Province Key Laboratory of Intelligent Building Energy Efficiency, Suzhou University of Science and Technology, Suzhou, Jiangsu, 215006
4 Collaborative Innovation Center of Novel Software Technology and Industrialization, Jiangsu, 210000, China
5 Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, 130012, China
6 College of Electronic & Information Engineering, Suzhou University of Science and Technology, Jiangsu, Suzhou, 215000, China

* Corresponding author email: quanliu@suda.edu.cn, sunshine-620@163.com

Abstract: As classic methods for handling continuous action space problem for continuous action space problem in RL, the actor-critic (AC) algorithm and its variants still fail to be sample efficiency. Therefore, we propose a method based on learning two linear models for planning. The two linear models refers to state-based piecewise model and action-based piecewise model, which are determined by the divisions for the state and action space, respectively. Through division, the models are learned more accurately. To accelerate the convergence, the sample near the goal is saved and used to learn the model, the value and the policy to balance the distribution of the samples. On two classic RL benchmarks with continuous MDPs, the proposed method shows the ability of learning an optimal policy by combing both models, and it also outperforms the representative methods in terms of convergence rate and sample efficiency.

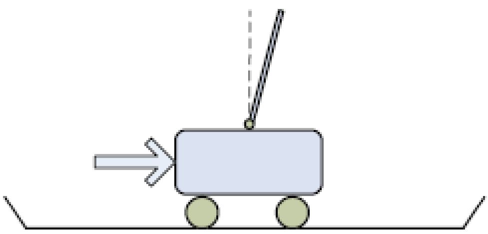


Figure 1. The Pole balancing problem

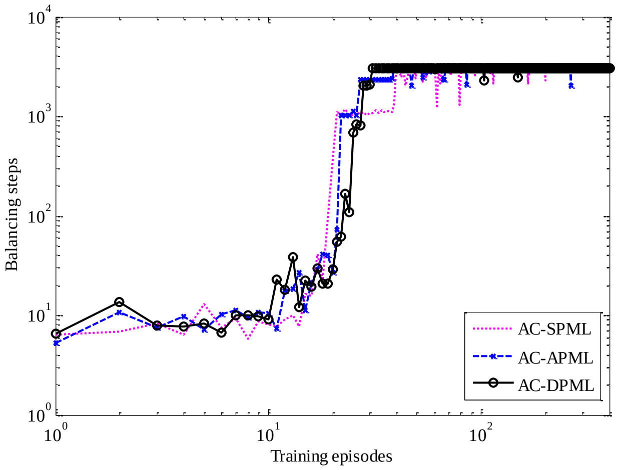
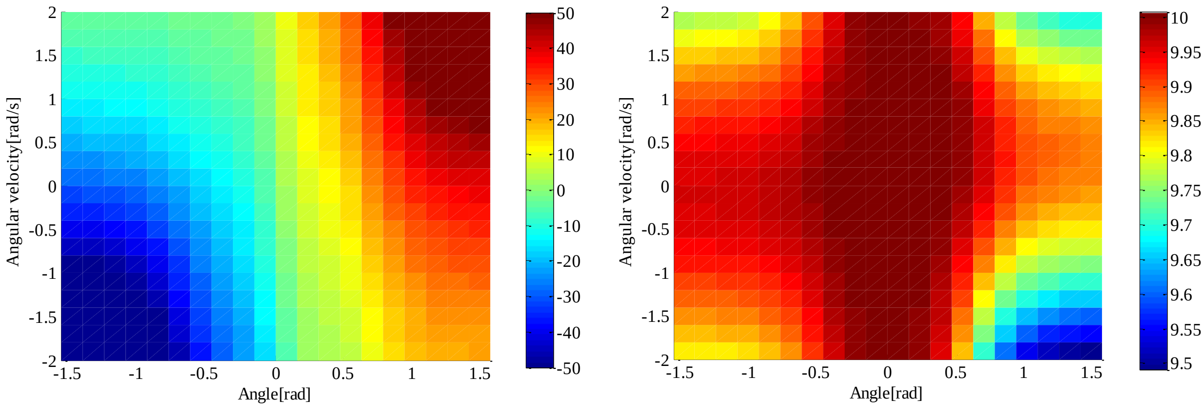


Figure 2. Comparisons of different piecewise models



(a) Final policy obtained after the training is over

(b) Final value function obtained after the training is over

Figure 3. Comparisons of the learned policy and optimal value function

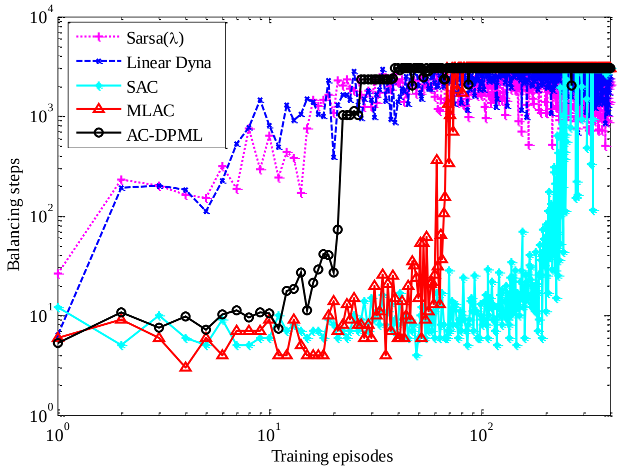


Figure 4. Comparisons of the balancing steps

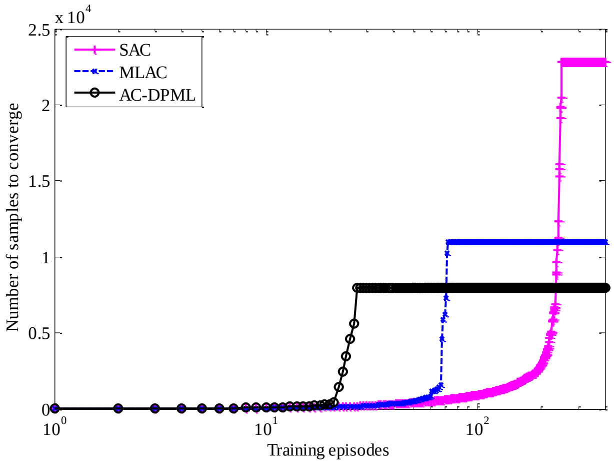


Figure 5. Comparisons of the sample efficiency

Conclusions.

This paper proposes an improved AC algorithm based on two piecewise models, the state-based piecewise model and the action-based piecewise model, to improve the sample efficiency and convergence rate for the problems with continuous state and action spaces. The empirical results show that the two models can cooperate well, additionally, the performance becomes more stable after introducing two piecewise models. In comparison to the discrete action algorithms Sarsa (λ) and linear Dyna as well as the continuous action algorithms SAC and MLAC, AC-DPML behaves well not only in convergence rate but also in sample efficiency. The performances of the discrete action algorithms Sarsa(λ) and linear Dyna do not look as well as those of the compared continuous algorithms. The comparison results between the method with model learning and the one without model learning, e.g., the discrete methods linear Dyna versus Sarsa(λ) or the continuous methods MLAC versus SAC, seem to demonstrate that model learning can improve the performance to a certain extent.

Since the introduction of the piecewise models can really improve the model accuracy, the sample efficiency and the convergence from the experimental results, it would be interesting to apply the two kinds of models to more complex domains, e.g., the inputs are figures or videos, so as to improve the performances for these domains.

.

.

References

[1] S. Adam, L. Bu_soniu, and R. Babu_ska. Experience replay for real-time reinforcement learning control. Machine Learning, 2(42):201{212, 2012.

[2] H. Berenji and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. IEEE Transactions on Neural Networks, 5(3):724{740, 1992.

[3] J. Boyan. Technical update: Least-square temporal difference learning. Machine Learning, 49(2-3):233{246, 2002.

[4] S. Bradtke and A. Barto. Linear least-squares algorithms for temporal difference learning. Machine learning, 22(1-3):33{57, 1996.

[5] L. Bu_soniu, R. Babu_ska, B. Schutter, and D. Ernst. Reinforcement learning and Dynamic Programming Using Function Approximators. CRC Press, New York, USA, 2010.

[6] I. Grondman, M. Vaandrager, L. Bu_soniu, R. Babu_sska, and S. E. E_cient model learningmethods for actor-critic control. IEEE Transactions on Systems, Man, and Cyberneticss, 3(42):591{602, 2012.

[7] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine. Continuous deep Q-learning with model-based acceleration. In ICML, 2016.

[8] L. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. Machine Learning, 8(3):293{321, 1992.

[9] A. Moore and C. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. Machine Learning, 1(13):103{130, 1993.

[10] J. Peng and R. Williams. E_cient learning and planning within the Dyna framework. Adaptive Behavior, 4(1):437{454, 1993.
[11] M. Santos, J. Mar_tin H., V. L_opez, and B. G. Dyna-H: a heuristic planning reinforcement learning algorithm applied to role-playing game strategy decision systems. Knowledge Based Systems, 32(1):28{36, 2012.
[12] J. Sorg and S. Singh. Linear options. In AAMAS, pages 31-38, 2010.
[13] R. Sutton. Integrated architecture for learning, planning and reacting based on approximating dynamic programming. In ICML, pages 216{224, 1990.
[14] R. Sutton and A. Barto. Reinforcement learning: An introduction. MIT Press, Cambridge, Massachusset, USA, 1998.
[15] R. Sutton, C. Szepesv_ari, A. Geramifard, and M. Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. In UAI, pages 528{536, 2008.
[16] M. Tagorti and B. Scherer. On the rate of the convergence and error bournds for LSTD(_). In ICML, pages 528{536, 2015.
[17] H. Van Seijen and R. Sutton. A deeper look at planning as learning from replay. In ICML, pages 692-700, 2015.
[18] C. Watkins and P. Dayan. Q-learning. Machine Learning, 3-4(8):279{292, 1992.
[19] H. Yao and C. Szepesv_ari. Approximate policy iteration with linear action models. In AAAI, 2012.

.

Acknowledgements

This paper was partially supported by Innovation Center of Novel Software Technology and Industrialization, National Natural Science Foundation of China (61472262, 61502323, 61502329, 61272005, 61303108, 61373094), Natural Science Foundation of Jiangsu (BK2012616), Provincial Natural Science Foundation of Jiangsu (BK20151260), High School Natural Foundation of Jiangsu (13KJB520020, 16KJB520041), Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172014K04), Suzhou Industrial application of basic research program part (SYG201422)