SCC-rFMQ Learning in Cooperative Markov Games with Continuous Actions

Extended Abstract

Chengwei Zhang
School of Computer Science and Technology, Tianjin university
chenvy@tju.edu.cn

Xiaohong Li
School of Computer Science and Technology, Tianjin university
xiaohongli@tju.edu.cn

Jianye Hao*
School of Computer Software, Tianjin university
jianye.hao@tju.edu.cn

Siqi Chen
Southwest university
Chongqing, China
siqichen@swu.edu.cn

Karl Tuyls
University of Liverpool
Liverpool, UK
k.tuyls@liverpool.ac.uk

Zhiyong Feng
School of Computer Software, Tianjin university
zhiyongfeng@tju.edu.cn

ABSTRACT

Although many reinforcement learning methods have been proposed for learning the optimal solutions in single-agent continuous action domains, multiagent coordination domains with continuous action have received relatively few investigations. In this paper, we propose an independent learner hierarchical method, named Sample Continuous Coordination with recursive Frequency Maximum Q-Value (SCC-rFMQ), which divides the coordination problem into two layers. The first layer samples a finite set of actions from the continuous action spaces by a sampling mechanism with variable exploratory rates, and the second layer evaluates the actions in the sampled action set and updates the policy using a multiagent reinforcement learning coordination method. By constructing coordination mechanisms at both levels, SCC-rFMQ can handle coordination problems in continuous action cooperative Markov games effectively. Experimental results show that SCC-rFMQ outperforms other reinforcement learning algorithms.

KEYWORDS

Learning agent-to-agent interactions (negotiation, trust, coordination); Multiagent learning

ACM Reference Format:
Chengwei Zhang, Xiaohong Li, Jianye Hao, Siqi Chen, Karl Tuyls, and Zhiyong Feng. 2018. SCC-rFMQ Learning in Cooperative Markov Games with Continuous Actions. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

A large number of multiagent coordination domains involve continuous action spaces [7, 13], where agents not only need to coordinate with other agents towards desirable outcomes efficiently [1, 5, 6, 8, 9, 20],but also have to deal with infinitely large action spaces [3, 4, 10–12, 15, 16, 18].

We propose a reinforcement learning method for multiagent continuous-actions coordination problems named Sample Continuous Coordination with recursive Frequency Maximum Q-Value(SCC-rFMQ), which divides the problem into two layers: (1) the action set modification layer; (2) the evaluation and policy updating layer. The first layer extracts a discrete action set from the original continuous action spaces by a variation of re-sampling mechanism inspired by SMC [3]. The new re-sampling mechanism preserves the best action of each agent and uses a variable exploratory rate to control the resample distribution and the convergence of the re-sampling mechanism. The variable exploratory rate is adjusted by a strategy named the Win or Learn More (WoLM) principle, to handle non-stationarity and stochasticity problems in cooperative MASs during the action set modification period. In the evaluation and policy updating layer, we extend the PHC algorithm [2] with the idea of the rFMQ methods [6] such that it can handle coordination problems in multiagent multi-state environments. Experimental results show that SCC-rFMQ outperforms other algorithms regarding coordination efficiency and effectiveness.

2 SCC-RFMQ

The SCC-rFMQ divides the learning process into two layers: the action set modification layer, and the evaluation and policy updating layer. To handle coordination problems, we add a coordination strategy for each layer.

In SCC-rFMQ, the initial sample set $A_i(s)$ is a subset of the continuous action set $\mathbf{A}_i(s)$ with $|A_i(s)| = n$, where all elements $a \in A_i(s)$ are randomly selected from $\mathbf{A}_i(s)$. Each round in SCC-rFMQ consists of two steps, i.e., Action set modification (Line 5) and Evaluation & Policy updating (Line 6). First, whenever the resample condition is satisfied, the sample set $A_i(s)$ is updated by the Coordination Resample strategy. The condition we used here is quite simple: every c rounds for each state. Next, it moves to the valuation and policy updating, which evaluates action values in $A_i(s)$

*Corresponding author

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

Algorithm 1 SCC-rFMQ for agent i with n samples

For all state $s \in S$, initialize the available action set $A_i(s)$ by drawing n samples from $\mathbf{A}_i(s)$
repeat
  $s \leftarrow$ initial state
  repeat
    Action set modification:
    if resample condition is satisfied **then**
     Resample $A_i(s)$ using the SCC strategy
    Evaluation & Policy updating:
    for all $a \in A_i(s)$ **do**
     Update $\pi_i(s,a)$ and $Q_i(s,a)$ by multi-state rFMQ
    Update state: $s \leftarrow s'$
  until s is an absorbing state
until the repeated game ends

and updates the policy using the multi-state rFMQ learning strategy.

Sample Continuous Coordination (SCC) Strategy. To solve learning problems in continuous action cooperative multiagent games, the SCC strategy preserves the currently best action and resamples $n-1$ new actions according to a variable probability distribution, and use a variable exploratory rate to control the convergence of the sampling strategy by adjusting the sampling probability distribution adaptively. We propose WoLM principle to adjust the exploratory rate.

Specifically, we first update the exploratory rate $l_i(s)$ for agent i at state s following the WoLM principle. If the current average reward is no less than the accumulate average reward $V_i(s)$, $l_i(s)$ is decreased to $l_i(s)\delta_d$ ($\delta_d < 1$), otherwise, it is increased to $l_i(s)\delta_l$ ($\delta_l > 1$). The variable exploration rate $l_i(s)$ ensures that the sampling range can be adaptively changed in response to the changing environment, while also ensures that sampling can find optimal solutions efficiently. Then, we update the accumulate reward $V_i(s)$ using the learning rate α_s. Next, the available action set $A_i(s)$ is resampled by preserving the currently best action a_{max} and drawing $n-1$ new samples following uniform distribution $U\left[a_{max} - l_i(s), a_{max} + l_i(s)\right]$. In this way, when actions in $A_i(s)$ are close to the optimal action of the whole action space $\mathbf{A}_i(s)$, the exploratory rate $l_i(s)$) will be gradually decreased to a very small value and eventually reaches 0. Thus it ensures that our algorithm can converge to an optimal policy. Finally, $\pi_i(s,a)$ and $Q_i(s,a)$ are re-initialized to $1/n$ and 0, to ensure that each new sampled action has enough observations to obtain a relatively correct estimate Q given that the total number of observations are limited.

Multi-state rFMQ Strategy. Combining with the idea of rFMQ, we extend the PHC algorithm to multi-state multiagent games. Initially, if the set $A_i(s)$ has been resampled, $F_i(s,a)$, $Q_i^{max}(s,a)$ and $E_i(s,a)$ are initialized for all actions in $A_i(s)$, where $E_i(s,a)$ is a weighted average result of $Q_i(s,a)$ and the historical maximum of the state-action pair $Q_i^{max}(s,a)$, and $F_i(s,a)$ is the weight estimating the frequency of receiving the maximum reward when action a is executed. After initialization, agent i selects an action a based on policy $\pi_i(s,a)$ with certain degree of explorations, and receives reward r; then, the expected reward $Q_i(s,a)$ is updated by Q-learning strategy [19]. Next, the process of calculating $E_i(s,a)$ is a natural extension of the original single-state rFMQ [6] to multiple states. Specially, $F_i(s,a)$ is recursively computed using a learning rate α_f,

$$F \leftarrow \begin{cases} 1 & r + \gamma \max\limits_{a'} Q_i(s',a') > Q_{max}(s,a) \\ (1-\alpha_f)F + \alpha_f & r + \gamma \max\limits_{a'} Q_i(s',a') = Q_{max}(s,a) \\ (1-\alpha_f)F & r + \gamma \max\limits_{a'} Q_i(s',a') < Q_{max}(s,a) \end{cases}$$

and $E_i(s,a) \leftarrow F_i(s,a)Q_{max}(s,a) + (1 - F_i(s,a))Q_i(s,a)$. Finally, the policy $\pi_i(s,a)$ is updated according to $E_i(s,a)$ by PHC strategy [2]. Overall, the improved rFMQ strategy can be applied in multi-state games and can handle stochastic coordination problems.

3 EXPERIMENTAL EVALUATION

Using the bilinear interpolation techniques [14], we construct continuous action game models based on two classic matrix games shown in Figure 1(a), which usually be used to test the ability to solve non-stationarity and stochasticity problems.

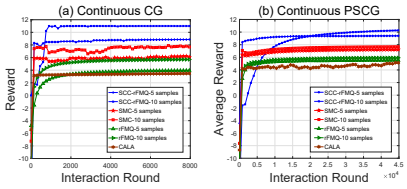
(a) Climbing Game

		Agent 2		
		A	B	C
Agent 1	A	11	-30	0
	B	-30	7	6
	C	0	0	5

(b)Partially stochastic Climbing Game

		Agent 2		
		A	B	C
Agent 1	A	11	-30	0
	B	-30	14/0	6
	C	0	0	5

(a) Game



(b) Performance

Figure 1: Performance comparison between SCC-rFMQ, SMC-learning and rFMQ

We compare our algorithm SCC-rFMQ with SMC-learning [3], rFMQ [6] and CALA [17], in continuous CG and continuous PSCG. Both SMC-learning and SCC-rFMQ are initialized with 5 and 10 samples. For rFMQ algorithm, we conduct two games with 5 and 10 evenly distributed actions from the continuous CG. Figure 1(b) shows the experimental results in continuous CG and continuous PSCG averaged over 50 runs. We can see that in all cases, our algorithm SCC-rFMQ significantly outperforms the other three algorithms, followed by SMC-learning, and CALA performs worst.

ACKNOWLEDGMENTS

This work has partially been sponsored by the National Science Foundation of China (No. 61572349, 61272106, 61702362)

REFERENCES

[1] Daan Bloembergen, Michael Kaisers, and Karl Tuyls. 2011. Empirical and theoretical support for lenient learning. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*. 1105–1106.

[2] M. Bowling and Veloso. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136, 2 (2002), 215–250.

[3] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. 2007. Reinforcement Learning in Continuous Action Spaces through Sequential Monte Carlo Methods. In *Conference on Neural Information Processing Systems*. 833–840.

[4] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[5] Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. 2007. Hysteretic q-learning :an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*. IEEE, 64–69.

[6] Laetitia Matignon, Guillaume j. Laurent, and Nadine Le fort piat. 2012. Review: Independent Reinforcement Learners in Cooperative Markov Games: A Survey Regarding Coordination Problems. *Knowl. Eng. Rev.* 27, 1 (2012), 1–31.

[7] Jingbo Meng, Dmitri Williams, and Cuihua Shen. 2015. Channels matter: Multimodal connectedness, types of co-players and social capital for Multiplayer Online Battle Arena gamers. *Computers in Human Behavior* 52 (2015), 190 – 199.

[8] Gregory Palmer, Karl Tuyls, Daan Bloembergen, and Rahul Savani. 2017. Lenient Multi-Agent Deep Reinforcement Learning. *CoRR* abs/1707.04402 (2017).

[9] Liviu Panait, Keith Sullivan, and Sean Luke. 2006. Lenient learners in cooperative multiagent systems. In *Proceedings of the 5th international joint conference on Autonomous agents and multiagent systems*. 801–803.

[10] Jason Pazis and Michail G. Lagoudakis. 2011. Reinforcement learning in multidimensional continuous action spaces. In *IEEE Symposium on Adaptive Dynamic Programming & Reinforcement Learning*. IEEE, 97–104.

[11] Jason Pazis and Ronald Parr. 2011. Generalized Value Functions for Large Action Sets. In *International Conference on Machine Learning, ICML 2011*. Bellevue, Washington, USA, 1185–1192.

[12] Jan Peters and Stefan Schaal. 2008. 2008 Special Issue: Reinforcement Learning of Motor Skills with Policy Gradients. *Neural Netw.* 21, 4 (2008).

[13] Martin Riedmiller, Thomas Gabel, Roland Hafner, and Sascha Lange. 2009. Reinforcement Learning for Robot Soccer. *Auton. Robots* 27, 1 (2009), 55–73.

[14] Santanu Saha Ray. 2016. *Numerical analysis with algorithms and programming*. CRC Press, Taylor & Francis Group, Boca Raton.

[15] Brian Sallans and Geoffrey E. Hinton. 2004. Reinforcement Learning with Factored States and Actions. *J. Mach. Learn. Res.* 5 (2004), 1063–1088.

[16] Richard S. Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, and Eric Wiewiora. 2009. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Danyluk Et*. 993–1000.

[17] M. A. L. Thathachar and P. S. Sastry. 2004. Networks of Learning Automata: Techniques for Online Stochastic Optimization. *Kluwer Academic Publishers* (2004).

[18] Hado van Hasselt. 2012. *Reinforcement Learning in Continuous State and Action Spaces*. Springer Berlin Heidelberg, Berlin, Heidelberg, 207–251.

[19] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q -learning. *Machine Learning* 8, 3-4 (1992), 279–292.

[20] Ermo Wei and Sean Luke. 2016. Lenient Learning in Independent-learner Stochastic Cooperative Games. *J. Mach. Learn. Res.* 17, 1 (2016), 2914–2955.