Non-independent Intelligent Creatures Reinforcement Learning Mechanism Research Based on I-XCS

PengJianXi[1, a], Jianxiong Tan[1,b]

[1]FOSHAN POLYTECHNIC, Foshan,Guangdong 528137, China

[a]nhpjx@163.com, [b]jianxiong_tan@163.com

Keywords: artificial intelligence; Non-independent intelligent creatures;I-XCS;gradient descent; reinforcement learning

Abstract. In order to solve many problems of reinforcement learning of Non-independent intelligent creatures in artificial intelligence, such as the single MDP environment and narrow learning space. This paper designed an Non-independent intelligent creatures reinforcement learning mechanism based on the Improved XCS classifier. This learning mechanism based on the original XCS classification capabilities and online knowledge, it constructs a high-stability, low-dimensional approximation method by using the gradient descent related technologies. This method has low-storage ability and enhances the inductive learning ability of intelligent creatures. Simulation experiment results show that the I-XCS classification learning algorithm not only can efficiently solve MDP environment issues such as single, narrow space, but also to a certain extent improved the analysis of non-independent intelligent creatures in reinforcement learning performance.

Introduction

Approximation method is one of the core learning tools in the field of artificial intelligence, it does not construct any structure model for studying environmental and engineering analysis, it is that merits attention, often applied to reinforcement learning mechanism of intelligent creatures。 However, in the study existing mechanism of intelligent life , the approximation method to show the way the theory is generally expressed by the approximate data in tabular form, such form consists of two key factors, one of which is a collection of intelligent biological state, and the second is the state of the corresponding action command collection. Therefore, approaching the core of data tables is that the interaction between two key factors of key-value pairs. However, in the field of non-independent intelligent creatures, with the gradual increase of the corresponding data between their state and action, the requirement of learning storage space are also increasing. Meanwhile, learning convergence, iterative cycle accordingly been extended, so that the approximation method for the main deficiency is not satisfied for the multi-independent intelligent creatures learning mechanism.

Approximation methods in view of the above deficiencies, in this paper,based on the improved XCS learning classifier (I-XCS), we design a non-independent intelligent creatures learning mechanisms .On the basis of the traditional XCS classifier and online learning knowledge, the I-XCS learning classifier constructs a new mapping formula relations. Such relations are mainly corresponding relationship between low dimensions and action key value associated with information and learning environment returning parameter. Through gradient descent related technologies to construct one more stable approximation method, it can maintain lower dimension level between learning environment returning parameters and information of the approximation data tables. At the same time, the new approximation method requires a lower reinforcement learning environment and space, and can enrich learning summary strategies library of the non-independent intelligent creatures, helps to optimize reinforcement learning ability of non-independent intelligent creatures. According to simulation experiment results, show that the I-XCS classification algorithm can efficiently solve MDP environment issues such as single, small space, and to a certain extent improved the analysis of non-independent intelligent creatures in reinforcement learning performance.

A Study on Reinforcement Learning Mechanism

Non-independent intelligent creatures reinforcement learning mechanism is a process that relies mainly on the perception and operation of each other between around a simulated environment and proper error handling strategy to get the learning return parameters, also the parameters apply to the optimal learning strategies mechanism of online learning theory. Assume at the time point t, intelligent creatures state designate for s^t, the learning environment of intelligent creature is different, use one operation instruction a_t in the action instruction set of state corresponding A, to get the return parameter r_{t+1}, through such a learning process, intelligent creatures state changes $s_t \to s_{t+1}$. We know that, at the time point k,we can get the most optimized return parameter of intelligent creatures is

$$Q(s,\alpha) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{t+1+k}\right] \tag{1}$$

which γ is return discount factor， and $0 \le \gamma \le 1$， explains the importance of the return parameter, then turn the value into numerical. The size of the γ value has proportional relationship with the return parameter.

Approximation Learning Method.Approximation method of learning is a kind of reinforcement learning algorithm， known as MDK .This algorithm is mainly based on a different learning environment, takes different learning explore action instructions,and obtains different learning environments with corresponding returns correct and error message, as return parameter. At the same time in the maintenance of reinforcement learning convergence, iterative, the return parameter will be applied to reinforcement learning strategy mechanism of intelligent creatures. Algorithm is described in detail as follows:

Pre-set the learning status set $s = (s_1, s_2, \cdots, s_n)$,for any point in time t has $s_t \in s$.

Reference to status characteristic s_t for any point in time t of intelligent creatures to set up the corresponding action instruction set $A(\mathrm{s_t})$， and one action instruction $a_t \in A(\mathrm{s_t})$.

Reinforcement learning state transition method $T(s,a,s')$,that represents the mapping relationship between the corresponding key/value pair information $< s,a >$ and status after transform s'.

Return parameters method $R(\mathrm{s,a,s'})$,describes that the intelligent creature process the action to get the return parameter after operates the $(\mathrm{s,s'})$， that means the intelligent creature gets the return parameter is r_t at time point t.

Approximation method for the core algorithms and optimization strategy of convergence mechanism is as follows：

$$Q(\mathrm{s_t},\alpha_t) = \mathrm{Q}(\mathrm{s_t},\alpha_t) + \beta(\mathrm{r_t} + \gamma \max \mathrm{Q}(\mathrm{s_{t+1}},\alpha_{t+1}) - \mathrm{Q}(\mathrm{s_t},\alpha_t)) \tag{2}$$

$$\pi(\mathrm{s_t},\alpha_t) = \mathrm{argmaxQ}(\mathrm{s_{t+1}},\alpha) \tag{3}$$

and $\beta(0 \le \beta \le 1)$ is Learning efficiency factor, $\gamma(0 \le \gamma \le 1)$， $\alpha \in A$.

Gradient Descent Learning Method. In order to Maximum optimizes the deviation between the return parameter which obtained by $r_{t+1} + \lambda \max Q(\mathrm{s_t},\alpha)$ and the real return parameter which got by $Q(\mathrm{s_{t+1}},\alpha_{t+1})$,Gradient descent related technologies can effectively reduce the deviation parameter value，its Weight set W changes the approximation method into numerical. At any time point $t+1$， any one weight parameter W, and its variable is ΔW, known that：

$$\Delta w = \beta(r_{t+1} + \gamma \max Q(s_{t+1}, \alpha) - Q(s_t, \alpha_t)) \cdot \frac{\partial Q(s_t, \alpha_t)}{\partial w} \quad (4)$$

Gradient descent algorithm can be used for reinforcement learning environment mechanism in the field of intelligent creatures. This algorithm has many advantages, such as maintain learning convergence ,low dimensional stability, etc. By applying the gradient descent algorithm into the approximation learning method, can low dimensional stability, also is able to optimize the learning strategies problems.

$$\Delta w = \beta(r_{t+1} + \gamma \max Q(s_{t+1}, \alpha) - Q(s_t, \alpha_t)) \cdot [\frac{\partial Q(s_t, \alpha_t)}{\partial w} - \gamma \frac{\partial(\max Q(s_{t+1}, \alpha))}{\partial w}] \quad (5)$$

$$\Delta w = \beta(r_{t+1} + \gamma \max Q(s_{t+1}, \alpha) - Q(s_t, \alpha_t)) \cdot [\frac{\partial Q(s_t, \alpha_t)}{\partial w} - \phi\gamma \frac{\partial(\max Q(s_{t+1}, \alpha))}{\partial w}] \quad (6)$$

XCS Related Technologies

Learning classifier generally consists of three important values, which are the determination conditions, actions instruction, and the return parameters.determination conditions used to specify relevant learning environment and state parameters according to different situations; Action introductions which are the processing operations of operation instruction set in the reinforcement learning mechanism *XCS* . Return parameter P , a correct or wrong reference value is got after operation command issued by *XCS* processing operation. ξ is used to represent the deviation value of return parameter， F indicates the optimized accuracy of P .

Execution Policy. The learning strategy is randomly acquired in *XCS* ， its data information are key-value pairs of data mainly come from status sets and operation command sets. First, based on the different input parameters of the learning environment; secondly, reference F and P ， output speculation form and action instruction sets of *XCS* learning mechanism; finally, operation distribution probability values where get from different environment in the processing action instructions, in order to achieve the aim of optimization for original learning strategies in *XCS* . As:

$$P(\alpha_i) = \frac{\sum cl_k \in |M| \alpha_i P_k \times F_k}{\sum cl_k \in |M| a_i F_k} \quad (7)$$

Reinforcement Learning Mechanism. *XCS* in the implementation of learning strategies，for any action command α in different environment, *XCS* gets the return parameter R ,then P and F to do the optimization and upgrading Respectively, we know that：

$$P \leftarrow R + \gamma \max_{\alpha} P(\alpha) \quad (8)$$

$$P_j \leftarrow P_j + \beta(P - P_j) \quad (9)$$

$$\xi_j \leftarrow \xi_j + \beta(|P - \xi_j| - \xi_j) \quad (10)$$

$$k_j = \begin{Bmatrix} 1, if\ \xi_j \le \xi_0 \\ \alpha(\xi_j / \xi_0)^{-\nu}, otherwise \end{Bmatrix} \quad (11)$$

$$k_j' = \frac{(k_j \times num_j)}{\sum cl_k \in |A|_{-1} (k_j \times num_j)} \quad (12)$$

For $\xi_0(\xi_0 > 0)$ is a speculation redundancy deviation ratio of return parameter; $\nu(\nu > 0)$ and $\alpha(0 < \alpha < 1)$ are the parameter constants. While if ξ_0 over the normal parameter value, we will

change k into status k' in A, then F which in the learning strategy of XCS also refer to parameter values k to make appropriate changes：

$$F_k \leftarrow F_j + \beta(\mathrm{k_j'} - \mathrm{F_j}) \tag{13}$$

Rule Set Secondary Treatment. If there is no data matching reference value between XCS and corresponding learning environment in the rule sets, the the rule sets will be mine again to produce covering algorithm, thus appear arbitrary XCS. For its rule data information which excluded from rule sets will be placed into the relevant sets.

I-XCS Research and Design

Determine the Mapping Relationship.

$XCS \leftarrow Q(\mathrm{s_{t+1}}, \alpha_{t+1})$

We use the gradient descent related technologies into learning mechanism of XCS， construct its mapping relationship between XCS and $Q(\mathrm{s_{t+1}}, \alpha_{t+1})$. It shows below：

$$Q(\mathrm{s_{t+1}}, \alpha_{t+1}) = \frac{\sum cl_j \in [\mathrm{A}]_{-1} \mathrm{P_j} \times \mathrm{F_j}}{\sum cl_j \in [\mathrm{A}]_{-1} \mathrm{F_j}} \tag{14}$$

$XCS \leftarrow \Delta w$

The approximation parameter value is typically produced by the weight parameter value w of weight set W interact with approximation method. In XCS, P_j as approximation method key factor, so in the improved learning classifier, which is $I-XCS$, uses reference P_j into the calculation $\frac{\partial Q(\mathrm{s_{t+1}}, \alpha_{t+1})}{\partial w}$, in order to get gradient descent related parameter value from $cl_j \in [\mathrm{A}]_{-1}$.
As shows below：

$$\frac{\partial Q(\mathrm{s_{t+1}}, \alpha_{t+1})}{\partial w} = \frac{\partial}{\partial P_k}[\frac{\sum cl_j \in [\mathrm{A}]_{-1} \mathrm{P_j} \times F_j}{\sum cl_j \in [\mathrm{A}]_{-1} \mathrm{F_j}}] = \frac{1}{\sum cl_j \in [\mathrm{A}]_{-1} \mathrm{F_j}} \frac{\partial}{\partial P_k}[\sum_{d_j \in [\mathrm{A}]_{-1}} P_j \times P_j] = \frac{F_j}{\sum cl_j \in [\mathrm{A}]_{-1} \mathrm{F_j}} \tag{15}$$

Improved Reinforcement Learning Mechanism. On the basis of the traditional XCS, reinforcement learning mechanism of $I-XCS$ algorithm has been further optimized in the classification guessing ability. If $[\mathrm{A}]_{-1}$ is changed, the parameters of the corresponding devices as follows:

$$F_{[\mathrm{A}]_{-1}} = \sum_{d_j \in [\mathrm{A}]_{-1}} F_j \tag{16}$$

P_k Corresponding changes as follows:

$$P_k \leftarrow P_k + \beta(\mathrm{r} + \gamma \max \mathrm{P}(\alpha) - \mathrm{P_k}) \frac{F_k}{F_{[\mathrm{A}]_{-1}}} \tag{17}$$

Intelligent Creatures Coding Scheme. Intelligent creatures enhanced learning environment through a certain pattern (in box) to express， the number "X" which inside the box indicates here has been blocked, destination using the "M"。Intelligent creatures can be put in any free squares and continuous free squares.Intelligent creatures are equipped with behavioral processing sensing device, it capable of measuring and identification of grid status information. Sensing device using a dual digital coding, namely to block label: 10, the destination label: 11, vacant squares location label: 00, etc. All input values using hexadecimal. As shown in table 1.

S0	S1	S2	S3	
S5	S6	S7	S8	S9
X	X	M	S10	S11
X	X	X	S12	S13
X	X	X	S14	S15

S0	S1	S2	S3	S4
S5	S6	S7	S8	S9
X	X	M	S10	S11
X	X	X	S12	S13
X	X	X	S14	S15

Table 1 intelligent creatures status relationship table

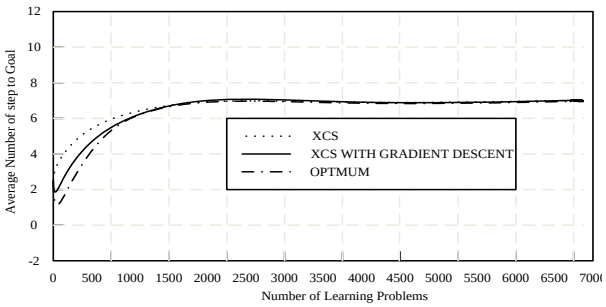

Fig.1 A smaller range learning environment of intelligent creatures reinforcement learning （N=2000）

Simulation Experiment

The purpose of simulation experiment is aimed at non-independent intelligent creatures use the improved classification learning algorithm XCS, also known as algorithm $I-XCS$, from the random Vacant square area in accordance with the relevant State and action instructions, movement to the target area. The initial location of these non-independent intelligent creatures are arbitrarily selected, and after the purpose of the experiment is completed, the simulation phase ends. Those intelligent creatures which have completed the experimental task to implement certain return incentive mechanism. This simulation is divided into multiple scenes for distribution to test the optimal results that provide from its $I-XCS$ algorithms. Specific detailed experimental set-up is as follows:

(1)Scene one: 25 squares distributed learning environment

This scene consisting of 25 squares in distributed environment， and a total of eight block grid, sixteen vacant squares, only one destination point.Participate in testing experimental classification equipment amounted $N=200$, as shown in figure1.

Shows that when the initial period in a simulating experiment, the improved algorithm $I-XCS$ is more optimized and similar to effects while compare to the traditional algorithm XCS .Shows that when the initial period in a simulating experiment, the improved algorithm is more optimized and similar to effects while compare to the traditional algorithm. Therefore, in the smaller classifier learning environment , the improvement algorithm is not dominant, but it can easily get better optimization effect of reinforcement learning.

Scene two:A “5” Labyrinth-type distribution environment

Using a “5” labyrinth-type distribution environment as example, shown in figure 3. We known that,when N reaches to 3000, both *XCS* and $I-XCS$ can efficiently complete the study optimization work. However, as N increases, the traditional *XCS* shows a gradual decline in reinforcement learning optimization capabilities. Correspondingly, the improved $I-XCS$ is not affected in any way, it still be able to effectively complete optimization tasks and its learning convergence, iterative stability, achieved good effect of reinforcement learning. Experiment results show in figure 4-5.

Scene three： “6”Labyrinth-type distribution environment

Using a “6” labyrinth-type distribution environment as example, shown in figure 6. We compare the labyrinth-type distributed learning environment above and know that the target destination M is more difficult to recognize, this increases the difficulty in reinforcement learning mechanism of intelligent creatures. It shows at figure 7, the advantages of $I-XCS$ are obvious， but the traditional *XCS* is unable to adapt to such environment, thus, in this kind of environment, $I-XCS$ not only has the stable learning convergence, and reinforcement learning optimization effectiveness is maximized. Compared with these two simulation scenarios, we know that the improved algorithm $I-XCS$ achieves a good global optimization in a single learning classifier.

x	x	x	x	x	x	x	x	x
x				M				x
x	x		x		x	x		x
x		x			x			x
x				x			x	x
x		x						x
x		x						x
x						x		x
x	x	x	x	x	x	x	x	x

Fig3 Labyrinth-type distribution environment

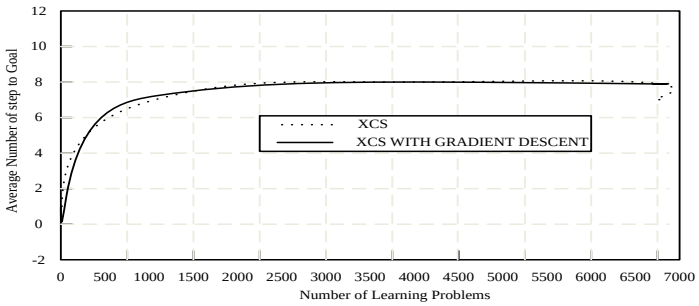

Fig.4 reinforcement learning of intelligent creatures in “5” labyrinth-type distribution learning environment （N=3000）

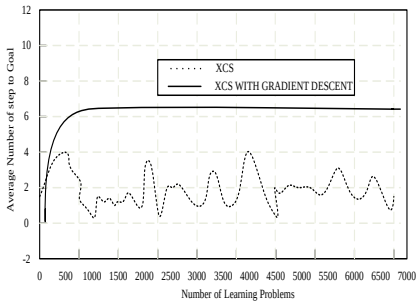

Fig.5 reinforcement learning of intelligent creaturesin “5” labyrinth-type distribution learning environment（N=3500） learning environment（N=3000）

x	x	x	x	x	x	x	x	x
x								x
x	x		x			x		x
x		x			x			x
x				x			x	x
x		x						x
x	x	x						x
x	M					x		x
x	x	x	x	x	x	x	x	x

Fig 6 “6”Labyrinth-type distribution environment

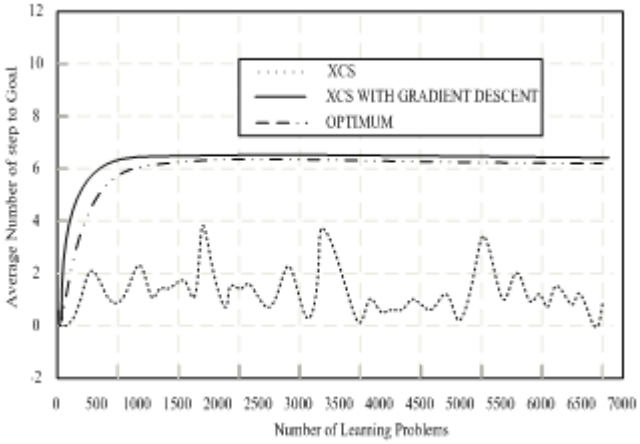

Fig.7 reinforcement learning of intelligent creatures in “6” labyrinth-type distribution

Conclusion

For the problems that the field of artificial intelligence,the non-independent intelligent creatures appear in reinforcement learning MDP environment issues such as single, narrow learning space, this paper uses improved algorithm $XCS(I-XCS)$ solves the problem of such intelligent creatures reinforcement learning. The main principle of $I-XCS$ algorithm is that base on the original XCS classification capabilities, and online knowledge. Using gradient descent related technology to construct a approximation method , which has the characteristics of high-stability and low-dimensional, and it is applied to the reinforcement learning mechanism of non-independent intelligent creatures under a relevant environment. The simulation results show that， based on the traditional XCS and gradient descent technique integrate into $I-XCS$, and it can be applied to the reinforcement learning mechanism of non-independent intelligent creatures, and its effect is superb.

References

[1] Dixon P W, Corne D W, Oates M J. Apreliminary investigation of modified XCS as a generic data mining tool[C]// Lanzi P L, Stolzmann W, Wilson S W, eds. LNAI, Advances in Learning Classifier Systems. vol. 2321, Berlin, Germany: Springer-Verlag, 2002: 133-150.

[2] Wiering M. Multi-agent reinforcement learning for traffic light control[C]// Proc. 17th Int. Conf. Mach. Learn. (ICML-00). Stanford Univ. Stanford, CA, 2009: 1151-1158.

[3] Butz M V, Lanzi P L, Wilson S W. Function approximation with XCS: Hyperellipsoidal conditions, recursive least squares, and compaction[J]. IEEE Trans. Evol. Comput, 2008, 12(3): 355-376.

[4] Hung K-T, Liu J-S, Chang Y-Z. Smooth path planning for a mobile robot by evolutionary multiobjective optimization[C]// IEEE Int. Symposium on Computational Intelligence in Robotics and Automation. Jacksonvile, Florida, Jun 2007.

[5] Zang Peng, Zhou Peng, Minnen D, et al. Discovering options from example trajectories[C]// Proc of the 26th Annual Int Conf on Machine Learning. New York: ACM, 2009: 1217-1224.

[6] Panait L, Luke S. Cooperative Multi-Agent Learning: The State of the Art[J]. Autonomous Agents and Multi-Agent Systems, 2005, 3(11): 383-434.

[7] Shi Chuan, Shi Zhongzhi, Wang Maoguang. Online hierarchical reinforcement learning based on path-matching[J]. Journal of Computer Research and Development, 2008, 45(9): 1470-1476.

[8] Yu Kai, Tresp V, Schwaighofer A. Learning Gaussian processes from multiple tasks[C]// Proc of the 22nd Annual Int Conf on Machine Learning. New York: ACM, 2005: 1017-1024.

[9] Tamei T, Shibata T. Fast reinforcement learning for three-dimensional kinetic human-robot cooperation with EMG-to-activation model[J]. Advanced Robotics, 2011, 25(5): 563-580.

[10] Bhatnagar S, Sutton R S, Ghabamzadeh M, et al. Natural actor-critic algorithms[J]. Automatica, 2009, 45(11): 2471-2482.

[11] Hachiya H, Peters J, Sugiyama M. Reward-weighted regression with sample reuse for direct policy search in reinforcement learning[J]. Neural Computation, 2011, 23(11): 2798-2832.

[12] Peters J, Schaal S. Natural actor-critic[J]. Neurocomputing, 2008, 71(7/8/9): 1180-1190.

[13] Chu B, Park J, Hong D. Tunnel ventilation controller design using an RLS-based natural actor-critic algorithm[J]. International Journal of Precision Engineering and Manufacturing, 2010, 11(6): 829-838.

[14] Han Y K, Kimura H. Motions obtaining of multi-degree-freedom underwater robot by using reinforcement learning algorithm[C]// Proceedings of TENCON IEEE Region 10 Conference. Fukuoka: IEEE, 2010: 1498-1502.