Acta Technica *62* No. 1B/2017, 337–348 © 2017 Institute of Thermomechanics CAS, v.v.i.

Practical online learning model based on big data balance[1]

Qu Jie[2]

Abstract. In view of the problem that the number of parameters is huge in Markov online learning, the convergence rate is slow and the online learning can't be implemented, a practical Markov online learning method based on the big data balance was proposed. Firstly, the learning parameters were represented in a practical manner to reduce the number of the learning parameters; and then, according to the a priori knowledge and observation data, the Markov method was used to learn and optimize the exploration and utilization of the balance relationship between the two; finally, Markov online learning method based on the big data balance was used to realize the rapid convergence of the learning process, so as to achieve the purpose of the online learning. The simulation results show that the algorithm can meet the requirements of real-time system performance.

Key words. Markov decision process, Markov online learning, big data balance.

1. Introduction

In the real dynamic system, the state transition function is usually unknown and dynamic. If the steady state model is used to describe the dynamic system, it will cause the distortion of the dynamic system modeling, so that the obtaining of the real approximation optimal value function and the optimal strategy cannot be guaranteed in theory. In view of this problem, it is necessary for the agent to learn in the interaction with the dynamic uncertain environment. Online learning is an effective optimal control learning method, which can be used under the conditions of complex model or uncertainty, etc. to realize the system multi stage optimization of the learning control based on the drive of the data [1–2].

The classical enhanced learning algorithms can be divided into two categories

[1]This work was support by the 13th Five-Year Research Project "Study on Cultivating Normal Students: Ability of the Information Technology Application Under the Background of Internet Plus Era", Supported by Education Science in Shaanxi Province (No. SGH16H169). The Key research Subject "Research on University Students' MOOC Study Status and Related Factors" Supported by Baoji University of Art and Sciences (No.ZK16073).

[2]College of Education of Baoji University of Art and Sciences, Baoji, Shaanxi, 721016, China

according to whether it is based on the big data, namely, the algorithms based on the big data balance and the algorithms based on the model-free data: the algorithms based on the big data balance and the ones based on the model-free. The algorithms based on the big data balance include TD learning, Q learning, SARSA [3] and other algorithms. The algorithms based on the model-free include DYNA-Q, priority sweep and other algorithms. The above classical enhanced learning algorithms have proved the convergence of the algorithms in theory. In the practical application areas, the number of learning parameters is huge. This is a typical NP difficult problem, which makes it hard to optimize the exploration and utilization of the balance of the two [4—5].

Markov Online Learning (referred to as MOL for short) conducts the modeling for the unknown model parameters by using the model priori knowledge, then makes the update to the posterior distribution of the unknown model parameters based on the observed data, and finally, conducts the planning according to the posterior distribution, thus obtaining the maximized expected reward value. Essentially, MOL transforms the learning problem into the planning problem. Since MOL can conduct the modeling for the unknown parameters and unknown models by using the priori distribution of the states, it has provided a perfect solution to optimize the balance of the optimized exploration and utilization, at the same time, it has gained extensive attention from the scholars at home and broad, and has become the hot spot in the research of the online learning field at present [6—7]. But there are two difficult problems existing in MOL: Firstly, the number of learning parameters is huge, and the scale increases exponentially [8]; secondly, the solution of the planning problem in the space of all the posterior belief states will encounter the "Curse of dimensionality" [9]. The above two difficult problems lead to the result that the existing MOL algorithms can only solve small scale problems, but cannot realize the online learning for large scale problems.

In this paper, a practical Markov online learning method based on the big data balance was put forward to solve the above problems, and the practical representation method was used to reduce the scale of the learning parameters; in the condition that the DBNs (Dynamic Markov Networks) structure (independent relationship between the variables) was unknown, Markov method was adopted to learn the unknown structure and parameter; finally, the Pointed-Based MOL (referred to as PBMOL for short) based on the big data balance was used to make the action choice in the posterior belief space, thus realizing the online planning and learning. The experimental and simulation results show that the proposed algorithm can effectively reduce the number of parameters and realize the online learning on the dynamic system.

2. Markov online learning modeling

The Markov Decision Processes (referred to as MDPs for short) can be described by a quaternion $< S, A, T, R >$. The state set $S = \{s_1, s_2, \cdots, s_n\}$ contains all the possible states of the agent; the action set $A = \{a_1, a_2, \cdots, a_n\}$ contains all the possible actions of the agent; state transition function; the state transition function

$T\left(s,a,s^{'}\right)=P\left(s^{'}|s,a\right)$, taking the probability of the transfer from the action a to the state $s^{'}$ when the agent is in the state$s$; the reward function $R\left(s,a,s^{'}\right)$, taking the reward value obtained from the transfer of the action a to the state $s^{'}$ in the state s.

In the online learning, the state transition function $T\left(s,a,s^{'}\right)$ is unknown learning parameter $\theta^{s,a,s^{'}}$. According to the literature [10], the Markov online learning based on the big data balance is defined as the Partially Observable Markov Decision Processes (referred to as POMDPs for short), which is described using the sextet-set $< S_{\mathrm{P}}, A_{\mathrm{P}}, Z_{\mathrm{P}}, T_{\mathrm{P}}, O_{\mathrm{P}}, R_{\mathrm{P}} >$. Here, S_{P} stands for the cross product of the discrete state S and the continuous unknown parameter $\theta^{s,a,s^{'}}$; the action set A_{P} is the same as the action set A; $Z_{\mathrm{P}}=S$. The state transition function $T_{\mathrm{P}}\left(s,\theta,a,s^{'},\theta^{'}\right)=P\left(s^{'},\theta^{'}|s,\theta,a\right)$ can be decomposed into the product of the two conditional distributions as the following:

$$\begin{aligned} T_{\mathrm{P}}\left(s,\theta,a,s^{'},\theta^{'}\right) &= P\left(s^{'},\theta^{'}|s,\theta,a\right) \\ &= P\left(s^{'}\left|s,\theta,a,\theta^{'}\right.\right)P\left(\theta^{'}|s,\theta,a\right) \\ &= \theta^{sas^{'}}\delta^{\theta\theta^{'}}. \end{aligned} \tag{1}$$

Here, $\delta^{\theta\theta^{'}}$ is the Kronaike function, which meets the following

$$\delta^{\theta\theta^{'}}=\left\{\begin{array}{ll} 1, & \theta^{'}=\theta \\ 0, & \text{otherwise} \end{array}\right\}. \tag{2}$$

The observation function $O_{\mathrm{P}}\left(s^{'},\theta^{'},a,z\right)=P\left(z\left|s^{'},\theta^{'},a\right.\right)$ stands for the probability that when the agent executes the action a and the state and parameters are transferred to $s^{'}$ and $\theta^{'}$, the observation is z. Since the reward function does not depend on θ or $\theta^{'}$, the reward function $R_{\mathrm{P}}\left(s,\theta,a,s^{'},\theta^{'}\right)=R\left(s,a,s^{'}\right)$ is the same as the reward function of MDPs.

According to the definition of Markov online learning, the MDPs problem is transformed into the POMDPs problem. In POMDPs, as the state is unknown, the probability distribution $b\left(s\right)$ of the state S is introduced, which is known as the belief. Through the introduction of the concept of belief, θ learning can be performed by using the belief monitoring method [11]. Making use of the Markov update rule, the belief is updated as the following:

$$b^{s,a,s^{'}}\left(\theta\right)=\eta b\left(\theta\right)P\left(s^{'}|\theta,s,a\right)=\eta b\left(\theta\right)\theta^{s,a,s^{'}}, \tag{3}$$

where η is the normalization factor.

The belief monitoring method is effective only when the priori and posterior distribution of beliefs are in the same distribution family, and Markov online learning adopts the Dirichlet distribution to represent the priori and posterior belief. As-

suming that the priori belief is $b(\theta) = \prod_{s,a} D(\theta^{sa}; n^{sa})$, n^{sa} is the vector of the super-parameter $n^{s,a,s'}$, then its posterior distribution is given as

$$\begin{aligned} b^{s,a,s'}(\theta) &= \eta\theta^{sas'} \prod_{s,a} D\left(\theta^{sa}, n^{sas'}\right) \\ &= \prod_{s,a} D\left(\theta^{sa}, n^{sa} + \delta_{\tilde{s},\tilde{a},\tilde{s}}, \left(s, a, s'\right)\right). \end{aligned} \tag{4}$$

Here, δ is still the Kronaike function, when $s = \tilde{s}$,$a = \tilde{a}$ and $s = \tilde{s}$, it is 1, and in other conditions, it is 0.

The goal of the online learning is to find the optimal strategy to optimize the balance exploration and utilization so as to obtain the maximum long-term reward value according to the posterior state of the current model under the condition that the state transition function is unknown. In the POMDPs, the strategy π is the mapping from the belief b to a, that is, $\pi(b) \to a$. The optimal strategy π^* is the strategy corresponding to the optimal value function V^*.

3. Practical Markov online learning based on the big data balance

In order to ensure that a good model is still available in the case of uncertainty, it is necessary to collect large amount of data during the learning process, which results in the exponential explosive growth of the learning parameters, thereby causing the failure of MOL to achieve rapid convergence. The practical representation method is an effective method to solve the "Curse of dimensionality" problem of the learning parameters [12]. In the practical representation methods, if the independent relationship between the variables in the DBNs is known, the size of the learning parameters can be easily compressed. However, in the practical application field, the structure of the DBNs is unknown. Therefore, the structure and parameters of the DBNs need to be learned at the same time.

3.1. Practical learning representation

In most real-world models, through the analysis of the internal structure of the state variables that, it can be found that the state variables can be represented by a set of random variable sets, which are called practical properties. This internal characteristic is known as the practical characteristic. The practical state is represented by a random finite variable set $X = \{X_1, X_2, \cdots, X_n\}$, in which each X_i stands for a characteristic of the state variable, X_i stands for the value set of each variable in the set X. One state can be represented as $s = \{X_1 = x_1, \cdots, X_n = x_n\}$, where $x_i \in X_i$, which can also be represented by $s = \{x_i\}_{i=1}^n$ for short. The state variable space is $|S| = \prod_{i=1}^n |X_i|$. After making the state variables practical, all the state transition function, the observation function, and the reward function can be represented using DBNs in compression [12].

Quantity $G(a)$ is defined as a two-layer directed acyclic graph, in which, $a \in A$,

the node is $X = \left\{X_1, \cdots, X_n, X_1^{'}, \cdots, X_n^{'}\right\}$. Symbol $\theta_{G(a)}$ is defined as a conditional probability table, then the state transition function T can be expressed by the $G(a)$ and $\theta_{G(a)}$. Quantity X_i is defined as the ith characteristic variable under the current state, $X_i^{'}$ is defined as the ith characteristic variable in the next moment. Finally $X_{\prec i}^{G(a)}$ stands for the value taken for the parent node when characteristic variable $X_i^{'} = x_i$. Then the state transition function is calculated as follows

$$T\left(s, a, s^{'}\right) = T\left(X, a, X^{'}\right) = \prod_{i=1}^{n} P\left(X_i^{'} \left| X_{\prec i}^{G(a)}\right.\right), \tag{5}$$

where $T\left(s, a, s^{'}\right)$ is the state transition function at the time when the decomposition representation is not performed, and $T\left(X, a, X^{'}\right)$ is the state transition function at the time of the practical representation.

In the FMDPs (Factored MDPs), since the state transition function, the observation function and the reward function can all be expressed by the conditional probability table of the DBNs, the above unknown models can be learned at the same time by repeating the calculation of the belief.

3.2. Belief posterior update

The practical Markov online learning can obtain the observation data, learning of unknown parameters and unknown structures through the interaction of the agents with the environment, so as to establish the state transition model and the reward model. In a deterministic environment, the initial belief $b(s)$ is given, and $b_{a,z'}\left(s^{'}\right)$ of its belief is calculated as

$$b_{a,z'}\left(s^{'}\right) = \eta\delta\left(\left[s^{'}\right]_{Z'} = z^{'}\right)\sum_{s} b\left(s\right) P\left(s^{'} | s, a\right), \tag{6}$$

where, η is the normalized constant; $\left[s^{'}\right]_{Z'}$ is the subset of the state variable values corresponding to the observed set of variables $Z^{'}$; δ is the Kronecker function, which returns the value 1 when $\left[s^{'}\right]_{Z'} = z^{'}$ is true and returns value 0 when it is false. Since the model and the structure are unknown, according to the knowledge of the previous section, it can be known that the update process of the belief state is as the following:

$$b\left(X^{'}, \theta_{G(a)}\right) = \eta\delta\sum_{X} P\left(X^{'} \left| X, a, \theta_{G(a)}\right.\right) b\left(X, \theta_{G(a)}\right), \tag{7}$$

Here, X and $X^{'}$ are the practical representation of the variable characteristics, a is the action, Z is the observation data set, z is the subset in Z, $\theta_{G(a)}$ is the unknown parameter, and δ is the Kronecker function.

However, since the belief state update requires the historical information, it is necessary to traverse all the historical observations and actions, resulting in the failure of convergence of Equation (7). According to literature [9], the belief state update is close-looped on the Dirichlet mixture product. Therefore, Dirichlet mixture product can be used to represent the state of the belief. And the representation form of the Dirichlet mixture product of the belief priori probability is as

$$b\left(X,\theta_{G(a)}\right)=\sum_{i}c_{i,X}\prod D_{i,X}\left(\theta_{G(a)}^{X_{\prec i}^{G}}\right), \tag{8}$$

where $c_{i,X}$ is the Dirichlet coefficient, D is the Dirichlet distribution function, $\theta_{G(a)}^{X_{\prec i}^{G}}=P\left(X^{'}\mid \text{ parents }\left(X^{'}\right)\right)$. Hence, the posterior belief after the update of the state of the belief is

$$b_{a,z'}\left(X,\theta_{G(a)}\right)=\sum_{j}c_{j,X'}\prod D_{j,X'}\left(\theta_{G(a)}^{X_{\prec i}^{G}}\right). \tag{9}$$

3.3. Value function parameterization

According to the above knowledge, it can be known that the Markov online learning in the FMDPs field can be modeled by DBNs with the model variable $\theta_{G(a)}$. If the unknown model variable $\theta_{G(a)}$ is used as the hidden variable of FMDPs, the FMDPs with unknown parameters can be transformed into the FPOMDPs (Factored POMDPs, POMDPs). According to the above conclusions, the existing FPOMDPs planning algorithms can be used to solve the MOL problem.

Porta and others put forward a BEETLE algorithm for iterative online learning for big data balance [9], which is only applicable to the MDPs field. On this basis, Porta et al. [10] proposed an improved BEETLE to process the continuous BEETLE algorithm and its improved algorithms take full advantage of the fact that the optimal value functions are parametric forms of the interface on the function set $\alpha-$, and the function $\alpha-$ is a multivariate polynomial. However, they are on the basis of MDPs or POMDPs, and cannot be generalized to the field of FPOMDPs [4]. This section draws on BEETLE's ideas, and extends the value function parameters to the FPOMDPs field.

The optimal value function of the discrete POMDPs has the characteristic of the piecewise linear convexity, that is, the optimal value function can be expressed by the upper interface of the linear segment set Γ (known as the vector $\alpha-$). The formula description is as the following:

$$V^{*}(b)=\max\,\alpha(b). \tag{10}$$

Each α is the linear combination of the probability value of each characteristic variable, that is $\alpha(b)=\sum_{X}c_{X}b(s)$. For a discrete state space, the number of states is bounded, and α can be expressed as the Dirichlet coefficient vector, that is $\alpha(X)=c_{X}$. For the continuous state POMDPs, the optimal value function is the upper envelope of the linear function (function $\alpha-$) set, and the equation description

is

$$\alpha(b) = \int_X c_X b(X)\, \mathrm{d}X . \tag{11}$$

In the practical online learning, assuming that the optimal value function at time k is $V^k(b)$, the set of function $\alpha-$ isΓ^k, then the following can be obtained

$$V^k(b) = \max_{\alpha \in \Gamma^k} \alpha(b) . \tag{12}$$

According to Bellman's update equation, the optimal value function at the time $k+1$ is $V^{k+1}(b)$, the set of function $\alpha-$ is Γ^{k+1}. Due to the introduction of the function $\alpha-$, the Bellman update equation can be rewritten as

$$V^{k+1}(b) = \max_{\alpha \in A} \sum_X b(X) R\left(X, a, \theta_{G(a)}\right) +$$

$$+ \gamma \sum_{z'} P\left(z' \left| b, a, \theta_{G(a)}\right.\right) \max_{\alpha \in \Gamma^k} \alpha\left(b_{a,z'}\right) , \tag{13}$$

According to the proof in literature [7], the function $\alpha-$ is the linear combination of the Dirichlet product. In each Bellman backup, the number of Dirichlet products in the linear combination is equal to the size of the state space. Therefore, the linear combination size of the function will increase with the decision time exponentially in the scale.

4. Experimental results

For the Chain problem, the BEETLE algorithm in literature [9] and the MC-MOL algorithm in literature [6] put forward in recent years can represent the level of the current Markov online learning algorithm. In view of the Mountain climbing problem, comparison with the EFSL algorithm in literature [8] is conducted.

4.1. Experimental results

In the Chain problem, there are two actions $\{a, b\}$, five states $\{1, 2, 3, 4, 5\}$, and the transfer probability of the action $P = 0.2$. Once the state 5 is achieved , the reward value is 10. Chain has Chain_Tied, Chain_Semi, Chain_Full and other three versions. Chain_Full refers that both the state transition function $T\left(s, a, s'\right)$ and state transition structure G are unknown. Chain_Semi refers that the state transition structure is known, the state transition function is unknown and there is a dependency between actions. Chain_Tied refers that the dynamic system is known, the action transition probability is unknown, and the action and state are independent. Therefore, the Chain problem has a variety of uncertainties, which is the ideal platform for the evaluation of the online learning algorithm.

In this paper, the three versions of the Chain problem were tested in 500 experiments, and each experiment executed 1000 steps (the number of iterations); then

for the reward value, the experimental results of the average and standard deviation was taken. The greater the value of reward, the more superior the algorithm shall be. Table 1 shows the comparison of the reward values of different algorithms, in which, n. Symbol v stands for not available, Optimal stands for the optimal value under the ideal condition; BEETLE is the value iterative algorithm based on the big data balance. The algorithm uses DDNs (Dynamic Decision Networks) to decompose the state. MC-MOL is the Markov online learning algorithm based on Monte Carlo, Q-Learning is a kind of $\varepsilon-$ greedy strategy, and the value of ε ranges from 0 to 0.5, PB-MOL is the algorithm proposed in this paper. The experimental data is the sampling results of $K = 1000$.

Table 1. Comparison of the reward values for different algorithms

Problem	BEETLE	MC-MOL	Q-Learning	PB-MOL
Chain_Tied	3650 ± 41	3618 ± 29	1616 ± 24	3659 ± 20
Chain_Semi	3648 ± 41	*n.v.*	1616 ± 24	3661 ± 21
Chain_Full	1754 ± 42	1646 ± 32	1616 ± 24	2565 ± 23

As can be known from the experimental data in Table 1, the average value of PB-MOL and BEETLE and MC-MOL is not the same in the Chain_Tied and Chain_Semi problems with less uncertain factors, but the PB-MOL algorithm is closer to the true optimal value In the large-scale Chain_Full problem, the state transition function and state transition structure are unknown, the uncertain factors are more, PB-MOL average reward value is 2565, which is significantly higher than BEETLE and MC-MOL algorithm. Therefore, BEETLE has more Good performance. As the use of Monte Carlo sampling method can effectively reduce the scale of the problem, the exploration and use can be balanced better. As Q-Learning is a free online learning method of this model, it is related to state transition function and other models are independent, so its reward value in the three versions of Chain remains unchanged. BEETLE, MC-MOL and PB-MOL are three different types of Markov online learning methods. As can be seen from Table 1, Kofu online learning method makes full use of the priori knowledge, which can effectively enhance the learning effect and improve the reward value.

Figure 1 shows the situation of the change of the accumulated reward value of four algorithms BEETLE, MC-MOL, Q-Learning and PB-MOL with the number of iterations. The experiment is the iterative result of the first 1000 steps. As an be known from Figure 1, the accumulated reward value of the PB-MOL proposed in this paper is maximum, while the accumulated reward value of the Q-Learning algorithm is minimum. The results of the BEETLE algorithm and MC-MOL algorithm are close to those of the PB-MOL algorithm. As can be known from the comparative experiment of the accumulated reward value, Markov online learning has more superior performance than the Q learning. In the iteration of the first 1000 steps, the learning rate of the algorithm is constant.

Table 2. Comparison of algorithm calculation time (in ms)

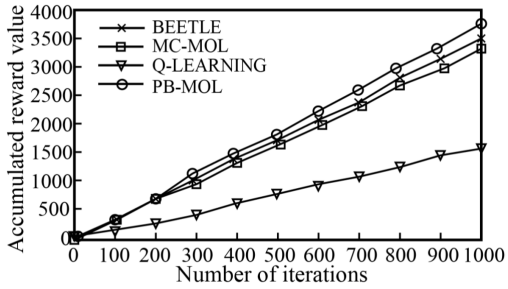

Fig. 1. Comparison of accumulated reward values

Problem	BEETLE		MC-MOL		PB-MOL	
	Offline	Online	Offline	Online	Offline	Online
Chain_Tied	400	1500	1.8e+6	32	400	18
Chain_Semi	1300	1300	*n.v.*	*n.v.*	1300	22
Chain_Full	14800	18000	*n.v.*	*n.v.*	14800	37

Table 2 shows the comparison of the calculation time for different Markov online learning algorithms, where *n.v.* stands for not available. From the data in the table, it can be seen that it is PB-MOL, where *n.v.* stands for not available. The data in the table shows that PB-MOL and MC-MOL on-line computations are less time consuming, which have higher real-time. However, from the value of the reward shown in Table 1, MC-MOL shows relatively big error in the large scale problem solving. PB-MOL off-line calculation method and BEETLE are the same, and there is the time-consuming problem. The offline pre-calculation will not affect the online real-time performance of the algorithm; at the same time, offline training can help obtain better priori knowledge, so as to get as large accumulated reward value as possible, which has properly solved the difficult problem of exploration and utilization in the online learning.

4.2. Simulation experiment of car climbing problem

In the literature on the related online learning, the car climbing learning control is usually used as a typical continuous state space online learning problem to verify the learning efficiency and generalization performance of the algorithm. The goal of the car is to climb to the top of the mountain. Before climbing the top of the mountain will not get positive feedback, hence the car has no knowledge on the environment that it is in.

Due to the lack of power, the car cannot climb directly to the top of the mountain. Therefore, it must first climb to the left to get sufficient kinetic energy, so as to reach the top of the mountain. The car's kinetic equation is as the following:

$$\begin{cases} x_{t+1} = x_t + v_{t+1}, \\ v_{t+1} = v_t + 0.001a_t - 0.0025\cos(3x_t), \end{cases} \tag{14}$$

where x stands for the location of the car, $x \in [-1.2, 0.5]$, v stands for the speed of the car, $v \in [-0.07, 0.07]$, a_t stands for the space of the action and $A(s) \in \{-1, 0, 1\}$. When $x_t = -1.2$, the speed of the car is 0; when $x = 0.5$, the goal of the car is achieved. And the beginning point of the car is $x = -0.5$, $v = 0.0$.

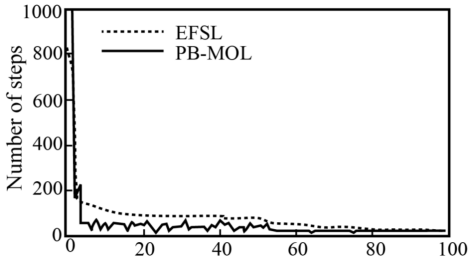

Fig. 2. Car climbing problem learning curve

The main evaluation index of the car climbing learning system is the number of steps of the car from the starting point position to the target position and the number of episodes required to achieve the steady state. The less the number of studies, the less the algorithm. In order to test the effectiveness of the proposed PBMOL learning algorithm, the algorithm is combined with EFSL (Enhanced Fuzzy Sarsa Learning). In this paper, the EFSL (Enhanced Fuzzy Sarsa Learning) and algorithm were compared to test the effectiveness of the PB-MOL learning algorithm proposed in this paper, and the experimental results were obtained, as shown in Fig. 2. In this experiment, the maximum number of steps was the 1000, the learning rate was 0.1, the discount factor was 0.9, the maximum and minimum of the temperature parameter were 0.1 and 0.001, respectively; and the sampling period was 0.02s. As can be seen from Fig. 2, the PB-MOL algorithm achieved the goal of climbing the car in 15 to 20 time steps after about 8 times of learning, while the EFSL needed about 100 time steps after about 10 times of learning, thus achieving the goal of the car climbing the mountain. If the EFSL needed to achieve the goal of the car within 20 time steps, it needed to learn at least 75 times. The experimental results show that, PB-MOL has better convergence and real-timeness than EFSL.

5. Conclusion

In view of the "curse of dimensionality" problem of the learning parameters in the Markov online learning based on the big data balance, the practical method was used to reduce the dimension of the unknown learning parameters in this paper. For the dynamic system of the model under the uncertain environment, the simultaneous learning of DBNs structure and the unknown parameters were used to effectively realize the real modeling of the dynamic system in the uncertain environment and solve the problem of the difficulty in the application modeling. The PRACTICAL

ONLINE LEARNING MODEL of the 11 unknown parameters were regarded as the hidden variables of MDPs, and the MDPs learning problem was transformed into the planning problem of POMDPs. All the existing POMDPs planning methods were applied to the MDPs learning through the transformation from MDPs to POMDPs, so as to solve the difficult problem of the online learning generalization. Finally, an online value iterative algorithm based on the big data balance was proposed to realize the online planning and learning.

References

[1] R. Costanza: *Science and ecological economics: Integrating of the study of humans and the rest of nature.* Bulletin of Science, Technology and Society *29* (2009), No. 5, 358–373.

[2] H. T. Liu, B. R. Hong, S. H. Piao, X. M. Wang: *Evolutionary algorithm based reinforcement learning in the uncertain environments.* Acta Electronica Sinica *34* (2006), No. 07, 1356–1360.

[3] Q. Liu, J. Li, Q. M. Fu, Y. C. Fu, Z. M. Cui: *A multiple-goal Sarsa algorithm based on lost reward of greatest mass.* Acta Electronica Sinica *41* (2013), No. 8, 1469–1473.

[4] S. Ross, J. Pineau, B. Chaib-Draa, P. Kreitmann: *A bayesian approach for learning and planning in partially observable Markov decision processes.* Journal of Machine Learning Research *12* (2011), No. 4, 1729–1770.

[5] Y. Gao, J. K. Hu, B. N. Wang, D. L. Wang: *Elevator group control using reinforcement learning with CMAC.* Acta Electronica Sinica *35* (2007), No. 2, 362–365.

[6] F. Doshi-Velez, J. Pineau, N. Roy: *Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs.* Artificial Intelligence *187–188* (2012), 115–132.

[7] H. Itoh, H. Fukumoto, H. Wakuya, T. Fukukawa: *Bottom-up learning of hierarchical models in a class of deterministic POMDP environments.* International Journal of Applied Mathematics and Computer Science *25* (2015), No. 3, 597–615.

[8] V. Na, H. Ngo, S. Lee, T. C. Chung: *Approximate planning for bayesian hierarchical reinforcement learning.* Applied Intelligence *41* (2014), No. 3, 808–819.

[9] J. Asmuth, L. Li, M. L. Littman, A. Nouri, D. Wintage: *A bayesian sampling approach to exploration in reinforcement learning.* Conference on Uncertainty in Artificial Intelligence (UAI), 18–21 June 2009, Montreal, Quebec, Canada, AUAI Press Arlington, Virginia, USA (2009), 19–26.

[10] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney, R. K. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J. M. Brady, P. M. Matthews: *Advances in functional and structural MR image analysis and implementation as FSL.* Neuroimage *23*, (2004), Suppl. No. 1, S208–219.

[11] M. Kearns, S. Singh: *Near-optimal reinforcement learning in polynomial time.* Machine Learning *49* (2002), Nos. 2–3, 209–232.

[12] H. L. S. Younes, M. L. Littman, D. Weissman, J. Asmuth J: *The first probabilistic track of the international planning competition.* Journal of Artificial Intelligence Research *24* (2005), No. 1, 851–887.

Received June 29, 2017