We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

22

A Novel Credit Assignment to a Rule with Probabilistic State Transition

Wataru Uemura
Ryukoku University
Japan

1. Introduction

In this chapter, we introduce profit sharing method (Grefenstette, 1988) (Miyazaki et al., 1994a) which is a reinforcement learning method. Profit sharing can work well on the partially observable Markov decision process (POMDP) where a learning agent cannot distinguish an observation between states which need another action, because it is a typical non-bootstrap method, and its Q-value is usually handled accumulatively. So we study profit sharing as the next generation reinforcement learning system. First we discuss how to assign the credit to a rule on POMDP. The conventional reinforcement function of profit sharing does not consider POMDP. So we propose a novel credit assignment which considers the condition of the reward distribution on POMDP. Secondly, we discuss the probabilistic state transition on MDP. Profit sharing does not work well on the probabilistic state transition. We propose a novel learning method which considers the probabilistic state transition. It is similar to the Monte Carlo method. We therefore discuss the Q-values of our proposed method. In an environment with deterministic state transitions, we show the same performance for both conventional profit sharing and the proposed method. We also show the good performance of the proposed method against the conventional profit sharing.
In this chapter, we discuss the learning in POMDP and the probabilistic state transition. We show the advantages and disadvantages of the profit sharing method. We propose a novel learning method which has the same advantages and solves the disadvantages.
We propose how to handle the Q-values in an action-selection. Section 2 introduces the conventional reinforcement learning methods and profit sharing method. We propose the novel learning method in Section 3. Section 4 shows the results and finally Section 5 concludes this chapter.

2. Reinforcement learning system

In a reinforcement learning system (Sutton, 1990), a learning agent gets a reward if and only if it reaches the goal state. An agent learns a better policy by repeated trial and error. We just describe the goal condition, so an agent must learn how to go from the start state to the goal state by the interaction between an agent and an environment. At time t, an agent observes the **observation** o_t at the **state** s_t, and selects an **action** a_t by the **policy**. After selecting the action a_t, the environment will change from the state s_t to the next state s_{t+1}. When the next

state s_{t+1} is the goal state, the agent gets the reward r_{t+1}, and if the next state s_{t+1} is not the goal state, the reward r_{t+1} will be equal to 0, or less than 0 which means the penalty.
In Markov decision process (**MDP**) (Sutton, 1990), the probability $P_{st,st+1}$, which is the state transition probability from the state s_t to the state s_{t+1} by the action a_t, is decided by only the state s_t and the action a_t. If an agent cannot get the all of the state, then some other states are observed with the same observation. We call this a partially observable Markov decision process (**POMDP**) (Miyazaki et. al, 1998) (Whitehead & Balland, 1990). In a POMDP environment, an agent must select two or more actions at the same observation.

2.1 Q-learning

We introduce Q-learning (Watkins & Dayan, 1992) which estimate the rule's value as a Q-value. The Q-value means the expected return which is updated as follow:

$$\begin{aligned} Q(s_t,a_t) &\leftarrow Q(s_t,a_t)+\alpha\left[r_{t+1}+\gamma \max_{a_{t+1}\in A(s_{t+1})} Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right], \\ &= (1-\alpha)Q(s_t,a_t)+\alpha\left[r_{t+1}+\gamma \max_{a_{t+1}\in A(s_{t+1})} Q(s_{t+1},a_{t+1})\right] \end{aligned} \tag{1}$$

where α is the learning rate, and γ is the discount rate. After many trials, Q-value will reach the estimate value of its rule. Thus, an agent selects the rule which has the highest Q-value of its state in order to get the optimal policy. Using Q-learning, an agent can update Q-value per action-selection without the reward. We call this **on-line updating**. Thus we can set the any value as initial Q-value. We call this **optimistic initial value**.
In a POMDP environment the combination of $Q(o_t,a_t)$ and $Q(o_{t+1},a_{t+1})$ is effected by alias problems, where the observation o_t and o_{t+1} means the observation at the state s_t and s_{t+1} respectively, so Q-value cannot reach the optimal value. For example, $Q(o_1,a_1)$ has the high-value at the state s_1, on the other hand $Q(o_1,a_1)$ has the low-value at the state s_2, then $Q(o_1,a_1)$ has no aim.

2.2 Profit sharing

In this section, we introduce profit sharing (Grefenstette, 1988) (Miyazaki et. al, 1994a) (Miyazaki et. al, 1994b) which is a reinforcement learning method. A profit sharing method has some advantages over other learning methods which mean that it can learn in MDP and also POMDP environments. In profit sharing, the agent distributes the reward to the selected rules (called an **episode**) when it reaches the goal state. The distributed function $f(x)$ is called a **reinforcement function**, and in MDP (Miyazaki et. al, 1994a) (Miyazaki et. al, 1994b) it should be formed by

$$C\sum_{j=i}^{W} f(j) < f(i-1) \qquad (i=1,\ldots,W), \tag{2}$$

where C is the maximum number of conflicting effective rules, and W is the maximum length of episodes. We usually use the reinforcement function that implements Equation 2 as follow:

$$f(x) = 1/L^x, \tag{3}$$

where x is the number of steps from the goal state, and L is the number of actions at each state.

3. Novel profit sharing

3.1 Reward distribution in a POMDP

Profit sharing uses the estimate value of rules in selecting rules. The estimate value does not be correct value when an aliasing observation confuses the agent observation capability. The conventional reinforcement function of profit sharing has this problem. The reinforcement of profit sharing is expressed by

$$\omega(o_x, a_x) \leftarrow \omega(o_x, a_x) + r \times f(x), \tag{4}$$

where o_x is the observation from the state s_x. In Equation 4, there is no problem because profit sharing does not use the relationship between observations. Profit sharing does not correctly estimate rules if and only if a rule (o_x,a_x) is equal to a rule $(o_{x'},a_{x'})$, a state s_x is not equal to a state $s_{x'}$, and an action a_x is not equal to an action $a_{x'}$. We discuss this case.

The case of the problem in profit sharing is that an agent confuses between a reinforcement rule and a non-reinforcement one. For example, at Figure 1a an agent has to suppress the rule (s_{t1},a_j) than the rule (s_{t1},a_i). At Figure 1b an agent can not distinguish the state s_{t1} and the state s_{t2} from observation o $(=o_{t1}=o_{t2})$. If the agent suppresses the rule (o_{t1},a_j) than the rule (o_{t1},a_i) at the state s_{t1}, its suppression will reinforce the rule (o_{t2},a_i) to make a loop at the state s_{t2}. Both the rule (o,a_i) and the rule (o,a_j) at Figure 1b are needed to receive a reward and must not be suppressed. None of needed rules for a reward must be suppressed. On MDP it is needless for an agent to think of the rule suppression because there is not aliasing state (like Figure 1b). On POMDP it is need for an agent to think of the rule suppression. All rule for a reward should be reinforced equally. All rule in an episode should be reinforced equally at each state, because an agent can see no difference between Figure 1a and Figure 1b with one episode.

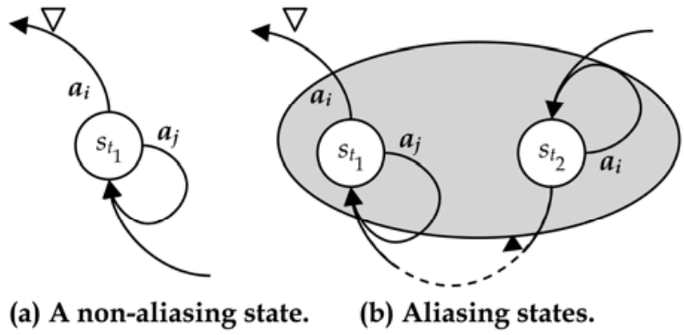
Thorem 1:

On POMDP the condition to distribute correctly the reward is

$$f(x) = \begin{cases} \alpha_{o_x} & \text{first reinforcement of rule x} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where rule x is reinforced by the function $f(x)$. α_{o_x} has to take the constant value at each observation o_x.

We propose the Episode-based Profit Sharing (EPS) that fills the need for the correct distribution on POMDP. The reinforcement function of EPS is



(a) A non-aliasing state. **(b) Aliasing states.**

Fig. 1. Aliasing states and a non-aliasing state.

$$f(x) = \begin{cases} 1/L^w & \text{first reinforcement of rule } x \\ 0 & \textit{otherwise,} \end{cases} \quad (6)$$

where L is the number of non-detour rules at a state, then the number of rule-1 is sufficient for L. We show that EPS can suppress the reinforcement of rules that make a loop.
If the environment has aliasing states, then the reinforcement function to distribute correctly rewards needs *Theorem* 1. The perceptual aliasing problem does not affect EPS because EPS can fill the needs from *Theorem* 1. So we have no need to think about the affectable of the aliasing states. We show the two case, one is that only one state makes a loop, and the other case is that multiple states make a loop. Next we propose the sub-episode method that reinforces rules with part of an episode. When part of an episode can be used always, the reinforcement function matches a geometrical decreasing function, that is the conventional function.

(a) a loop consisting of single state

Now we discuss the case that one observation makes a loop. The reinforcement value is written as Δ. The difference of reinforcement values between a non-detour rule and a detour rule is

$$\Delta(o, \text{non-detoure rule}) > \Delta(o, \text{detour rule}). \quad (7)$$

So EPS can suppress the reinforcement of rules that make a loop in the case of single state.

(b) a loop consisting of multiple states

Now we discuss the case that has two or more observations in order to make a loop. The difference of reinforcement values between a non-detour rule and a detour rule is

$$\Delta(o_l, \text{non-detoure rule}) > \Delta(o_l, \text{detour rule}). \quad (8)$$

EPS can suppress the reinforcement of rules that make a loop in the case of multiple states. So we can show the suppression proof of EPS.

(c) using part of an episode

We discuss about the sub-episodes (o_i,a_i), (o_{i+1},a_{i+1}), ..., (o_t,a_t) (i=1,2,...,t-1) which are the parts of an episode (o_1,a_1), (o_2,a_2), ..., (o_t,a_t). An agent can learn from the sub-episodes which start at the time i. To use sub-episodes has to fill the needs for *Theorem* 1 in order to distribute correctly rewards on POMDP. When an agent can see no difference between the observation o_{k1} and the observation o_{k2} affected by perceptual aliasing, there may be some difference between the state s_{k1} and the state s_{k2}. In this case, the agent can not use the sub-episode which has the rule (o_k,a_k) is the start rule in order to fill the needs for *Theorem* 1 ($k_1 < k \leq k_2$). That is to say that the agent can use the sub-episode starting at the rule (o_k,a_k) ($k \leq k_1$ or $k_2 < k$). It is the same when two or many observations are affected by perceptual aliasing. The rules between the observation o_{k1} and the observation o_{k2} are defined as rules *on an observation loop*. The flag to mean whether the rule (o_k,a_k) is on an observation loop or not is d_k which is defined as

$$d_k = \begin{cases} 0 & \text{ok is on an observation loop.} \\ 1 & \text{otherwise,} \end{cases} \quad (9)$$

An agent can reinforce rules using the length t-i+1 of the sub-episode (o_i,a_i), (o_{i+1},a_{i+1}), ..., (o_t,a_t). Now the amount $f(x)$ of reinforcement for rule (o_x,a_x) is

$$f(x) = \sum_{k=x}^{W} \frac{1}{L^k} d_k \cdot \tag{10}$$

Figure 2 shows this reuse sub-episodes. So the reinforcement function of EPS with sub-episodes is

$$f(x) = \begin{cases} \sum_{k=x}^{W} \frac{1}{L^k} d_k & \text{first reinforcement of rule } x \\ 0 & \textit{otherwise.} \end{cases} \tag{11}$$

The reinforcement function on MDP that is d_k=1 ($\forall k = 1,2,\ldots,W$) and has no same rules in an episode is

$$f(x) = \sum_{k=x}^{W} \frac{1}{L^k} \cdot \tag{12}$$

Given $W \to \infty$, the reinforcement function $f(x)$ becomes the geometrical decreasing function with a common ratio $1/L$. This function matches the conventional function.

3.2 Online updating

Usually softmax action selection is used for profit sharing because its Q-value means the accumulation of past rule values, for example, roulette distribution, Boltzmann distribution, and Gibbs distribution. In a POMDP environment, in some states, the agent cannot recognize that the observation there is not similar to the observation of another state. In other words, it gets the same observation in the other states. This problem is called an **alias problem** (Whitehead & Ballland, 1990).

Profit sharing is robust in a POMDP environment for two reasons. One is that updating the Q-value is non-bootstrapping. Non-bootstrapping means that the agent does not use Q-values which are in other states in order to estimate the Q-value. Updating the equation for profit sharing is as follows:

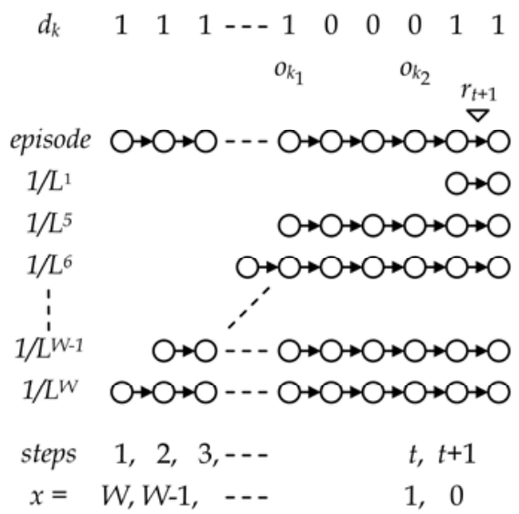


Fig. 2. Reuse sub-episodes (when $o_{k1} = o_{k2}$).

$$\omega(s_x, a_x) \leftarrow \omega(s_x, a_x) + f(x), \quad \text{For all value of } x \text{ in the episode.} \tag{13}$$

where $\omega(s_x, a_x)$ is the Q-value of the rule (s_x, a_x). Equation 13 was proposed by Miyazaki (Miyazaki et. al, 1994a) (Miyazaki et. al, 1994b). This updating euqation does not require the Q-value of another rule. So profit sharing is a non-bootstrapping method.

The second reason is that the action selection of profit sharing is a softmax action selection. In order to solve the alias problem, the agent must not select one action always often one observation because due to the alias problem the agent must select two or more actions. For example, in the state s_{t1} the agent gets the observation o (=o_{t1}), and the action which brings the agent near to the goal state is action a_i (shown at Figure 1b). On the other hand, in the state s_{t2} the agent gets the same observation o (=o_{t2}), however the action which bring the agent near to the next state is action a_j. Thus, the agent should not select one action for the one observation o. The agent must select both two actions, a_i and a_j, at the one observation o.

The conventional reinforcement learning methods (Watkins & Dayan, 1992) uses greedy action selection. When the action selection is greedy action selection, the agent can select the rule which has the highest Q-value of its state. Using this select method, a rule which has a secondary high Q-value is never selected. Thus the conventional reinforcement learning method does not work well in a POMDP environment. In an MDP environment, there is no aliasing states (shown in Figure 1a). So greedy action selection can work well. Using Equation 14 proposed by Miyazaki (Miyazaki et. al, 1994a) (Miyazaki et. al, 1994b) (called **accumulative profit sharing**), the agent can select two or more actions at the same observation. So accumulative profit sharing is robust in a POMDP environment.

Accumulative profit sharing, however, does not consider the probability of the state transition (Uemura et. al, 2007). For example, it distributes the same rewards whatever the state transition probability is. The expected value means $R \times P$, where R is the reward and P is the transition probability. So we should make the distributed reward nearly equal to its expected value.

A reinforcement function cannot know the state transition probability because many trials are needed to find it. Thus it is too difficult to estimate the rule-transition probability using only one episode. Some conventional reinforcement learning methods work per action selection, where the agent can update Q-values.

We propose a novel credit assignment method which considers the probabilistic state transition. Accumulative profit sharing does not consider the number of selection in the same rule. This method, therefore, distributes the same credit assignment to the rules which got the same rewards but have a different probabilistic state transition.

So we must count the number of selections in the same action, and discount the Q-value. The novel Q-value is as follows:

$$Q(s,a) \leftarrow N_r(s,a)/N_a(s,a) \times \omega(s,a), \tag{14}$$

where $N_r(s,a)$ is the number of rewards by the rule (s,a), and $N_a(s,a)$ is the number of selections of the rule (s,a).

If the state transition of rule (s,a) is always deterministic, then the number of rewards obtained $N_r(s,a)$ is almost equal to the number of selections of the rule $N_a(s,a)$. If and only if

the episode has a loop, $N_a(s,a)$ becomes larger than $N_r(s,a)$. If the rule (s,a) has the probabilistic state transition, $N_r(s,a)$ / $N_a(s,a)$ means an estimated value. In other words, $N_r(s,a)$ / $N_r(s,a)$ means the experiential rule transitional probability under its learning procedure.
For example, the conventional Monte Carlo method uses the average estimate value. Its estimating function is as follows:

$$Q(s,a) \leftarrow N_r(s,a)/N_a(s,a), \tag{15}$$

This equation brings the $Q(s,a)$ to the average of rewards. This, however, is not accumulative. Thus the Monte Carlo method requires greedy action selection. Our proposed method accumulates the rewards. Thus it requires softmax action selection. It is also robust for the POMDP environment. We call our proposed method the **accumulative Monte Carlo method**.

4. Experiment

4.1 Reward distribution in a POMDP

An agent cannot know how many states affected perceptual aliasing on POMDP. So we prepare the experimental environment which has aliasing states by half of all (Figure 3). Agent can select one action from four actions (up, down, left and right) at each state. If the direction of the selected action is equal to one of the arrow in the figure, then the agent moves to the next state. The observation o_1 is observed at the state s_1, s_2, and s_3. The agent should select randomly one action from three actions except for left action because the agent must select right, down, and up at each state. At the state s_4, s_5, and s_6, the agent has to learn the action moving to the next state because the observations are equal to the states. The performance means received rewards per number of the selected actions, and the performance by the optimum policy is 10/12 = 0.833.

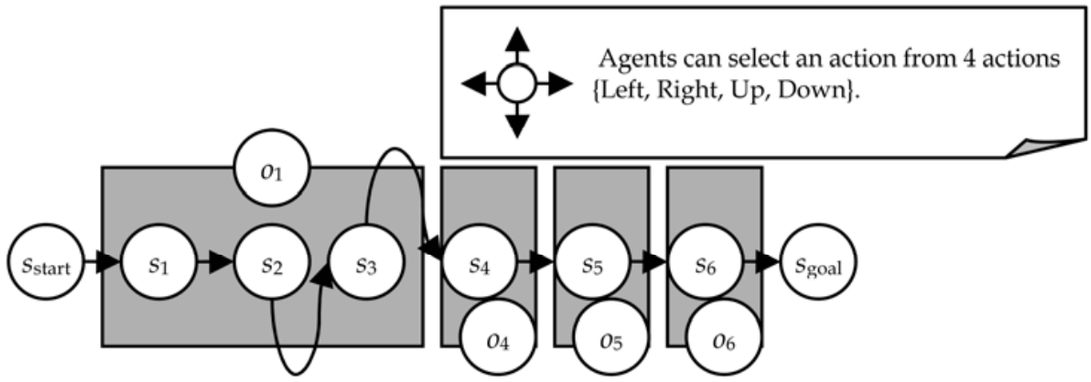


Fig. 3. The Experimental Environment in a POMDP.

Figure 4 shows the result. The action selection of Q-Learning is ε-greedy which selects the maximum Q-value in 90% probability and random actions in 10% probability. The conventional profit sharing with the geometric decreasing function is written as *PS (Decrease)*. The performance of PS (Decrease) becomes worse but the proposed profit sharing, EPS, can learn more policy.

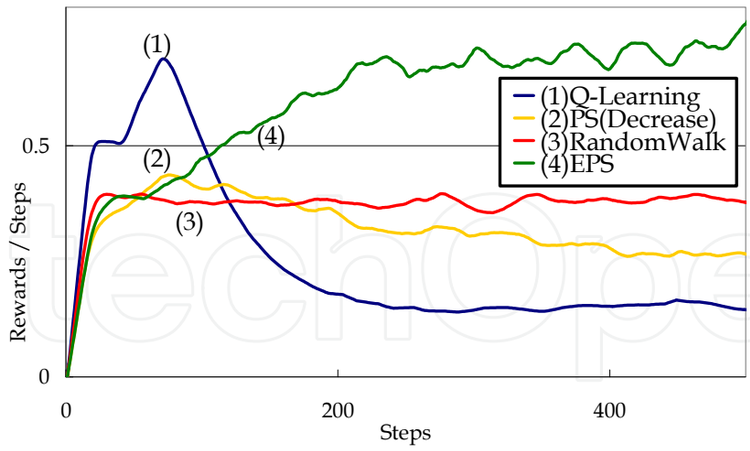


Fig. 4. The learning performances at POMDP.

4.2 Online updating

We carried out experiments in a maze (Sutton, 1998) (Figure 5). An agent starts at state S and selects one action from 4 actions (up, down, left and right). When the agent reaches the goal state G, the reward R = 10 is received, and the agent restarts at the start state S. The performance is how many rewards to get per step. All actions have the same probabilistic state transition. The agent goes to the selected state by the probability P = 0.8, and goes to the neighbour state by the probability P = 0.2.

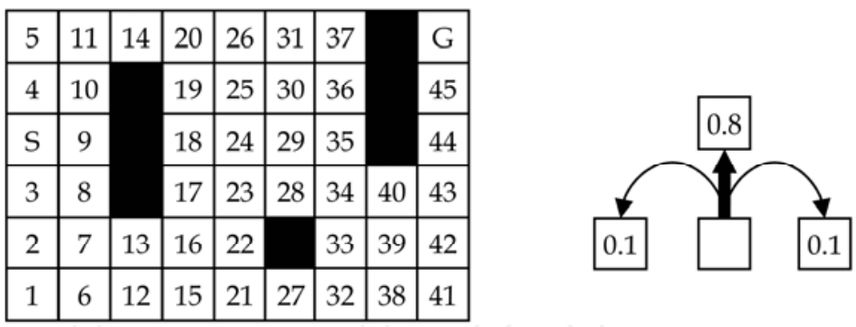


Fig. 5. The maze of Sutton with probabilistic state transitions.

The proposed method has almost the same performance as the conventional method in the non-probabilistic state transition. There is a difference if and only if the agent makes a loop in the early stage of learning. So in the first learning steps, the proposed method distributes slightly less rewards than conventional profit sharing. The performance for the probabilistic state transition is shown in Figure 6. The proposed method has better performance than the conventional method.

5. Conclusion

In this chapter, we have proposed a novel credit assignment method similar to profit sharing which considers the aliasing problem and the probabilistic state transition. We show that the condition to learn in a POMDP is to distribute equal rewards to rules at the same

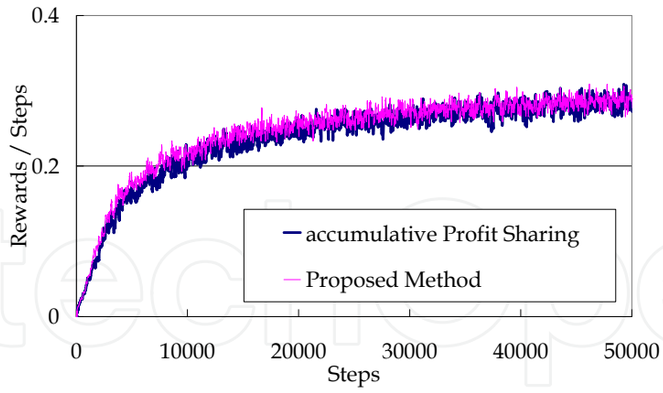


Fig. 6. The performance between conventional method and the proposed method.

state in an episode. We proposed a novel reward distribution method, called EPS, which considers this condition. Next, we consider the probabilistic state transition. If the agent experiences the same rule as the previous episode, the current episode has a loop rule, that is, its rule has a probabilistic state transition. So its rule value should be less than the previous reward. The equation $R \times P$, where R is the reward and P is the transition probability, shows the expected value. Thus the temporary rule variable should be divided by the number of its rule selection. Finally the temporary rule variable reaches to its expected value. We have proposed how to decrease the estimated values of rules per action selection.

In an environment with a deterministic state transition, we show the same performance for both conventional profit sharing and the proposed profit sharing. And we show the good performance of proposed profit sharing against the conventional profit sharing with a probabilistic state transition.

6. References

Arai, T. & Kragic, D. (1999). Name of paper, In: *Name of Book in Italics,* Name(s) of Editor(s), (Ed.), page numbers (first-last), Publisher, ISBN, Place of publication

Grefenstette, J. (1988). Credit assignment in rule discovery systems based on genetic algorithms. *Machine Learning*, Vol. 3, pp. 225 – 243.

Miyazaki, K., Yamamura, M. and Kobayashi, S. (1994). A Theory of Profit Sharing in Reinforcement Learning. *Journal of Japanese Society for Artificial Intelligence*, Vol. 9, No. 4, pp. 104 – 111.

Miyazaki, K., Yamamura, M. and Kobayashi, S. (1994). On the Rationality of Profit Sharing in Reinforcement Learning. *Proceedings of the 3rd International Conference on Fuzzy Logic*, Neural Nets and Soft Computing, pp. 285 – 288.

Miyazaki, K. and Kobayashi, S. (1998). Learning Deterministic Policies in Partially Observable Markov Decision Processes. *Proceedings of International Conference on Intelligent Autonomous System 5*, pp. 250 – 257.

Sutton, R. (1990). Integrated Architecture for learning, planning, and reacting based on approximating dynamic programming. *Proceedings of the 7th International Conference on Machine Learning*, pp. 216 – 224.

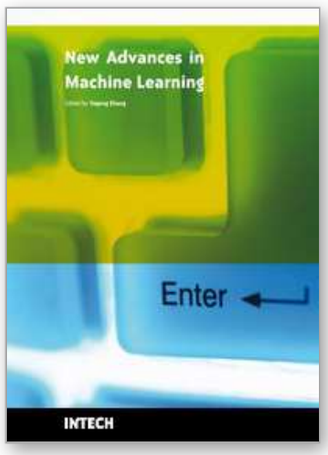
Sutton, R. (1998). Reinforcement learning – an introduction – , *the MIT Press*.

Uemura, W., Ueno A. and Tatsumi, S. (2004). The exploitation reinforcement learning method on POMDPs. *in Joint 2nd International Conference on Soft Computing and Intelligent Systems*, TUE – 1 – 3.

Uemura, W. (2007). About distributing rewards to a rule with probabilistic state transition, *in the SICE Annual Conference 2007, International Conference on Instrumentation, Control and Information Technology*, pp. 2762 – 2765.

Watkins, C. and Dayan, P. (1992). Technical note: Q-Learning, *Machine Learning*, Vol. 8, pp.279 – 292.

Whitehead, S. and Balland, D. (1990). Active perception and reinforcement learning, *Proceedings of the 7th International Conference on Machine Learning*, pp.162 – 169.



New Advances in Machine Learning

Edited by Yagang Zhang

ISBN 978-953-307-034-6
Hard cover, 366 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

The purpose of this book is to provide an up-to-date and systematical introduction to the principles and algorithms of machine learning. The definition of learning is broad enough to include most tasks that we commonly call “learning” tasks, as we use the word in daily life. It is also broad enough to encompass computers that improve from experience in quite straightforward ways. The book will be of interest to industrial engineers and scientists as well as academics who wish to pursue machine learning. The book is intended for both graduate and postgraduate students in fields such as computer science, cybernetics, system sciences, engineering, statistics, and social sciences, and as a reference for software professionals and practitioners. The wide scope of the book provides a good introduction to many approaches of machine learning, and it is also the source of useful bibliographical information.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen