Journal of Korea Multimedia Society Vol. 13, No. 12, December 2010(pp. 1820-1831)

Multiple Behaviors Learning and Prediction in Unknown Environment

Wei Song†, Kyungeun Cho††, Kyhyun Um†††

ABSTRACT

When interacting with unknown environments, an autonomous agent needs to decide which action or action order can result in a good state and determine the transition probability based on the current state and the action taken. The traditional multiple sequential learning model requires predefined probability of the states' transition. This paper proposes a multiple sequential learning and prediction system with definition of autonomous states to enhance the automatic performance of existing AI algorithms. In sequence learning process, the sensed states are classified into several group by a set of proposed motivation filters to reduce the learning computation. In prediction process, the learning agent makes a decision based on the estimation of each state's cost to get a high payoff from the given environment. The proposed learning and prediction algorithms heightens the automatic planning of the autonomous agent for interacting with the dynamic unknown environment. This model was tested in a virtual library.

Key words: Sequence learning, Autonomous agent, Virtual reality, Artificial Intelligence, Markov Decision Processing

1. INTRODUCTION

An autonomous agent is a system autonomously senses and acts in an environment over time with special assignment, purpose, or agenda [1]. One of the most important features of autonomous agent is adaptability for the situated environment.

In traditional game AI design or artificial life generation, autonomous agent is often created in a predefined environment with a priori knowledge [2]. The AI architecture always follows that the situated environment determines the agent's behavior range. However the agent is limited to specific environment and the predefined AI models or frameworks are difficult to implant into unknown environment.

To create an adaptive autonomous agent, such as robot and Non-player character (NPC), we need to solve the problem how to optimally react in unknown environment without a priori knowledge. Just like a baby, agents cannot create a well adaptive knowledge in a short period of time and the training and learning process must be taught by itself, partners or interactive environment. The learning approach allows agent to achieve a skilled performance in situated environment [3].

In this paper, we focus on how to improve sequence learning and prediction algorithms to help autonomous agent to adapt for unknown environment over time. We propose a coherent system for multiple sequential behavior generation. The system includes sequential behavior learning, prediction, and planning processes.

To finish a task, the agent needs to decide an adaptive actions order, instead of single action. Markov decision model is widely applied in plan-

※ Corresponding Author: Kyungeun Cho, Address: (100-715) 26, Pil-dong 3-ga jung-gu, Seoul, Korea , TEL : +82-2-2260-3834, FAX: +82-2-2260-3766, E-mail: cke@dongguk.edu
Receipt date: Nov. 29, 2010, Revision date: Dec. 13, 2010
Approval date: Dec. 28, 2010
† Dept. of Multimedia, Graduate School of Digital Image & Contents, Dongguk University
(E-mail: songwei@dongguk.edu)
†† Dept. of Game & Multimedia, Dongguk University
††† Dept. of Game & Multimedia, Dongguk University
(E-mail: khum@dongguk.edu)

ning, learning and prediction process of sequential behavior generation [4]. When the agent interacts with unknown environment, the number of sensed states keeps on increasing. It will cause a huge computation of sequence learning. We propose a motivation filter [5] to group the sensed states into several sequences so that the computation is smaller and the agent can make planning autonomously according to motivation.

In order to select adaptive action, we propose a sequence prediction algorithm to provide a reliable evaluation for the future state. Integrating the current state evaluation and the mean value of the future state payoff, the agent predicts each state's worth.

In this paper, we focus on how to improve sequence learning and prediction algorithms to help autonomous agent to adapt for unknown environment over time. We propose a coherent system for multiple sequential behavior generation. The system includes sequential behavior learning, prediction, and planning processes. Our proposed multiple sequential behavior learning algorithm can be applied in interactive games, virtual reality, autonomous robot and other AI environment.

The organization of this paper is as follows: In chapter 2 we explain related works on sequential learning algorithms and motivation-based learning models. In chapter 3 we propose a motivation filter to group sensed states. In chapter 4 we describe sequential behavior learning algorithm. In chapter 5, we derive prediction algorithms from learning result. In chapter 6, we detail the experiment of the proposed system in a virtual library environment. In chapter 7, we offer a conclusion and future work.

2. RELATED WORKS

When adapting to an unknown environment, agents always solves the problem of learning in a dynamic environment through trial-and-error interaction [6]. When making planning, instantaneous rewards may result bad situation in future. T.M. Gureckis [7] examine Softmax model, Eligibility trace model and Q-learning model to maximize long-term well being for people's strategies.

Meanwhile the existing learning and planning algorithms always depend on predefined state transition probability, which is not given in unknown environment. Ron Sun [8] presents a two-stage, bottom-up process for planning without a priori knowledge. He suggests a plan extraction algorithm to determine the relationship between generated Q-values and the probabilities of attaining the goals.

Because the agent continues to observe new states and perceive more sensed states, the computation of learning and prediction processes will be larger. Even when a game is simple, the agent's state space is very large due to the dynamic perceptions. To reduce the computational cost, Kwiatkowska [9] proposes an abstraction method for the state partition of the nondeterministic MDPs which is used to make choices for two players. Siddiqi [10] introduces a Dense-Mostly-Constant (DMC) transition matrix to enhance speedups for learning an optimal state transition sequence for the observations. Timmer [11] speeds up the learning process by dynamic programming and heuristic search. They propose a coarse partition of the state space which is possible to help the agent to learn very good policies. Lihong Li [12] investigatesthe idea of classification-based policy-search reinforcement learning methods where the policy is represented as a classifier mapping states to actions. Zengchang Qin [13] proposes a decision tree learning algorithm used for classification. Compared to back propagation Neural Networks, their algorithm has equivalent classification accuracy and better transparency.

According to neuroscience theory, the autonomous agent can make decision from intrinsic motivation. Sevin [14] have proposed a motivational model of action selection and hierarchical classifier system starting from motivation level. Singh [15] present a study of intrinsically motivated reinforcement learning to generate skill hierarchies in a form of Markov decision process. Merrick [16] extends reinforcement learning to multi-task application where motivation performs as a parameter of reward estimation.

Without state grouping method, the computation of learning and planning process will be very complicated. We propose a state grouping method to reduce computation of sequence planning and learning process. Differing from Kwiatkowska's research of limited state grouping method, we design a motivation filter to classify dynamic observed state into parallel subsequences.

According to Gureckis's testing result of Q-learning model, we propose a probability estimation algorithm that can provide a next-step evaluation based on generated transition probability distribution in sequence learning. The main focus is to help the agent make motivation-oriented plans from undefined states and probabilistic transition.

3. MOTIVATION FILTER USING AGENT'S CHARACTERISTICS

When continuously interacting with virtual environment, an agent may perceive a huge number of states transition that need to be stored in the memory. To reduce the computations, we design a filter to group states in learning process.

Filters that classify the sensed states by a set of reasonable features help the learning process by allowing fewer states to be computed. The imbalance of the agent's internal variables provides the agent with a motive to take action. This study proposes motivation as the filter of the states classifier. The proposed motivation filter groups sensed states and taken transitions into different behavior sequences.

The data flow of the motivation filter [5] is shown in Fig.1. The agent is affected by the stimulation (S) when interacting with the virtual environment. The input of the filter is the change in internal variables ($I \cdot S$), and the output is the change in motivation (ΔM). First, the agent is affected by the stimulation (S) due to interaction with the virtual world, which may change the agent's internal variables. According to the tendency (I) of the agent's characteristics, this filter enables the agent to determine which motivation is stimulated. From motivation alteration (ΔM), a reward is estimated for the learning process. Finally, the agent groups observed the state into sequence memory with a learned probability of the states' transition.

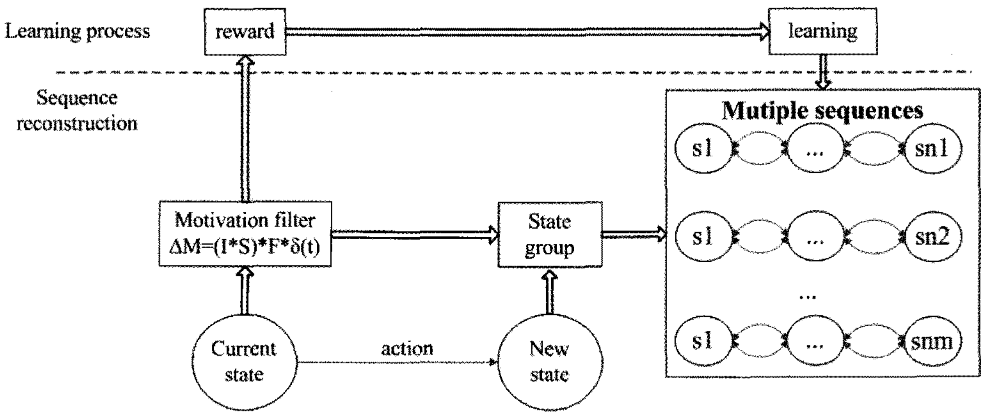


Fig. 1. Data flow chart of motivation filter.

4. SEQUENTIAL BEHAVIOR LEARNING ALGORITHM

When perceiving an environment without a priori knowledge, a virtual agent has many action choices and observes many new states and transitions that result in a dynamic update to the behavior sequence. A dynamic sequence extension is necessary to adapt to the environment.

The sensed information comprises the current state, possible actions, and a new sensed state. When updating the sequence, the agent estimates the stochastic transition probability using the Q-learning result. If the action results in a new state, the stimulated sequence will be updated with the new transition and state.

4.1 Integrated state definition

In learning process, the autonomous agent's states can be expressed as integration of some binary sub-states. We define the integrated states in table 1, where the value of element ssji can be 0, 1 and null (*). Null value means that there is no relation between state j with sub-state i. Then we can get the integrated state in form of a set of binary number. In other situation, we can define sub-states with integer value.

If the number of sensed sub-states becomes higher, the size of learning table will be larger. Therefore we propose a states combining method shown in fig.2. If the value of sub-state for states m1 and m2 are 0 and 1, and this value does not change in the result states n1 and n2 after taking action k, then states m1 and m2 will be combined to state m and states n1 and n2 will be combined to state n.

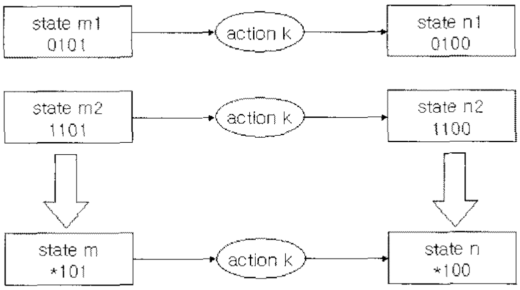


Fig. 2. states combination for learning table.

4.2 Integrated state definition

A sequence is defined as a tuple $M=(S,A,P_a(s,s'))$, where S denotes a set of states; A, a set of selective actions; and $P_a(s,s')$, the probability that a transition is occurring in state s and leading to state s'.

In unknown environment, it's not possible that all of the transitions between any two states. Therefore agent needs to determine which transitions can happen and initialize probability distribution of sequence. We propose a transition feasibility matrix to represent which transition agent has implemented. The transition feasibility matrix (T) is shown in Equation (1),

$$T=\begin{bmatrix} t_{00} & t_{01} & \cdots & t_{0n} \\ t_{10} & t_{11} & \cdots & t_{1n} \\ & & \cdots & \\ t_{n0} & t_{n1} & \cdots & t_{nn} \end{bmatrix} \quad where\ t_{ij} \in \{0,1\} \tag{1}$$

where n is the number of sensed states. We define t_{ij} as transition feasibility parameter. If t_{ij} =1,

Table 1. integrated states definition

	sub-state 1	sub-state 2	…	sub-state i	…
state 1	ss_{11}	ss_{12}	…	ss_{1i}	…
state 2	ss_{21}	ss_{22}	…	ss_{2i}	…
…	…	…	…	…	…
state j	ss_{j1}	ss_{j2}	…	ss_{ji}	…
…	…	…	…	…	…

agent can take more than one action to implement transition from state i to state j. If t_{ij} =0, this transition can not be achieved.

According to the perceptive transition feasibility matrix, the transition probability of the sequence is initialized evenly by equation (2). The element p_{ij} represents the transition probability from state i to state j.

$$P = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0n} \\ p_{10} & p_{11} & \cdots & p_{1n} \\ & & \cdots & \\ p_{n0} & p_{n1} & \cdots & p_{nn} \end{bmatrix} \qquad p_{ij} = \frac{t_{ij}}{\sum_k t_{ik}} \tag{2}$$

In game AI programming, the agent always randomly selects an action from the probability distribution of the states' transition. According to the learning theory, it is necessary for the agent to make an optimal decision with more rewards in the next state. We propose a transition probability evaluating algorithm by integrating behavior planning with a learning algorithm.

The Q-learning algorithm (3) is applied to evaluate the taken action's contribution to motivation satisfaction. For state s_t and action taken a from action set A, we can calculate an action cost using the following expression [17]:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha[R(s_t,a_t) + \gamma \max Q(s_{t+1},a_t) - Q(s_t,a_t)] \tag{3}$$

where $R(s_t,a)$ denotes an observed reward, s_t denotes the state at time t, and a denotes the action taken at state s_t. The coefficient α denotes the learning rate $(0 \le \alpha \le 1)$, and γ, the discount factor $(0 \le \gamma \le 1)$. The generated Q-value supports the learning agent to update the transition probability to an optimal value. If the action taken improves the agent's situation, the probability of this action occurring is higher in state s_t. Therefore, we suggest the following probability updating algorithm [5]:

$$p_a'(s_t,s_{t+1}) = \beta k Q'(s_t,a) p_a(s_t,s_{t+1}) + (1-\beta) p_a(s_t,s_{t+1}) \tag{4}$$

After agent takes action $\hat{a}$ from current state s_i, the probability of transition to sate s_j is $p_{\hat{a}}(s_i,s_j)$. To get sum of the transition probabilities of all optional actions, we can get the transition probability from current state s_i to next state s_j, denoted as $p(s_i,s_j)$. In another word, $\sum_{\hat{a}\in A} p_{\hat{a}}(s_i,s_j) = p(s_i,s_j)$ where A is actions set. Based on quality of Markov Chains, $\sum_{j\in S} p(s_i,s_j) = 1$, where S is a set of sensed states. Therefore

$$\sum_{j\in S}\sum_{\hat{a}\in A} p_{\hat{a}}(s_i,s_j) = 1 \tag{5}$$

Let's take equation (4) into equation (5), we can get

$$k = \frac{1}{\sum_{j\in S}\sum_{\hat{a}\in A} Q'(s_i,a) p_{\hat{a}}(s_i,s_j)} \tag{6}$$

Taking derived k into (6), we can derive the transition probability from s_t to s_{t+1} as:

$$p_a'(s_t,s_{t+1}) = \beta \frac{Q'(s_t,a) p_a(s_t,s_{t+1})}{\sum_{j\in S}\sum_{\hat{a}\in A} Q'(s_t,\hat{a}) p_{\hat{a}}(s_t,s_j)} + (1-\beta) p_a(s_t,s_{t+1}) \tag{7}$$

5. STATES PREDICTION AND BEHAVIOR PLANNING

In the game strategy, we always encounter a problem such that an instant payoff may not provide more rewards after several steps. We interpret a state's result at an n-step forward stage as the state's "cost." Using the evaluation of probabilistic transition and the agent's internal variable, we propose a state prediction algorithm to estimate each state's "cost."

We define $e(s_i,I)$ as an evaluation for the agent's current situation in state s_i, where I denotes an internal variable. With the calculated transition probability in the sequence, the agent can predict each state's "cost" $g(s_i)$ using equation (8). λ denotes a discount factor $(0 \le \lambda \le 1)$.

$$g(s_i) = \lambda e(s_i,I) + (1-\lambda)\sum_{j\in S}\sum_{\hat{a}\in A} p_{\hat{a}}(s_i,s_j) g(s_j) \tag{8}$$

In the next stage, the learning agent earns a reward $R(s_t,a)$ and the change in the internal variable is denoted by I. Then, the agent needs to correct the state's cost $g(s_i)$ to make it closer to the real observation $R(s_i,a)+\mu e(s_{i+1},I)$, where μ denotes the discount factor of the achieving state $(0 \le \mu \le 1)$.

To reduce the difference between $g(s_i)$ and observation, we modify parameter λ to satisfy $R(s_i,a)+\mu e(s_{i+1},I)-g(s_i)<\epsilon$, where ϵ is smaller than any positive number.

The solution of λ is derived as:

$$\lambda \approx \frac{R(s_i,a)+\mu e(s_{i+1},I)-\sum_{j\in S}\sum_{\hat{a}\in A} p_{\hat{a}}(s_i,s_j)g(s_j)}{e(s_i,I)-\sum_{j\in S}\sum_{\hat{a}\in A} p_{\hat{a}}(s_i,s_j)g(s_j)} \tag{9}$$

In the behavior planning process, we propose that the agent first selects a goal state with a higher cost and predicts an action with more rewards in the following states and then the agents chooses the action. According to the predicted result, indicating which state can result in the highest cost in future, the agent randomly selects an action from the generated probability distribution to achieve a good state with the highest cost.

6. SYSTEM EXPERIMENT AND ANALYSIS

To elucidate the mechanism of the multiple sequential behavior learning system using motivation filter, we tested the proposed algorithm in a virtual library, where the learning agent is a virtual student who wishes to borrow books with several internal variables, such as anger, thirst, relaxation, and a tendency to study.

When the agent comes into an unknown library, the agent tries several actions to percept and learns to interact with this environment. Using our proposed learning algorithm, the virtual agent can learn to borrow book without priori knowledge. Because the one of most important features of virtual reality and games is real-time processing, we need to simulate the proposed learning algorithm with low calculating consummation.

6.1 Simulation system design

The main network is designed as the framework in Figure 3. Firstly, a virtual agent receives environment signal which affects motivation and up-

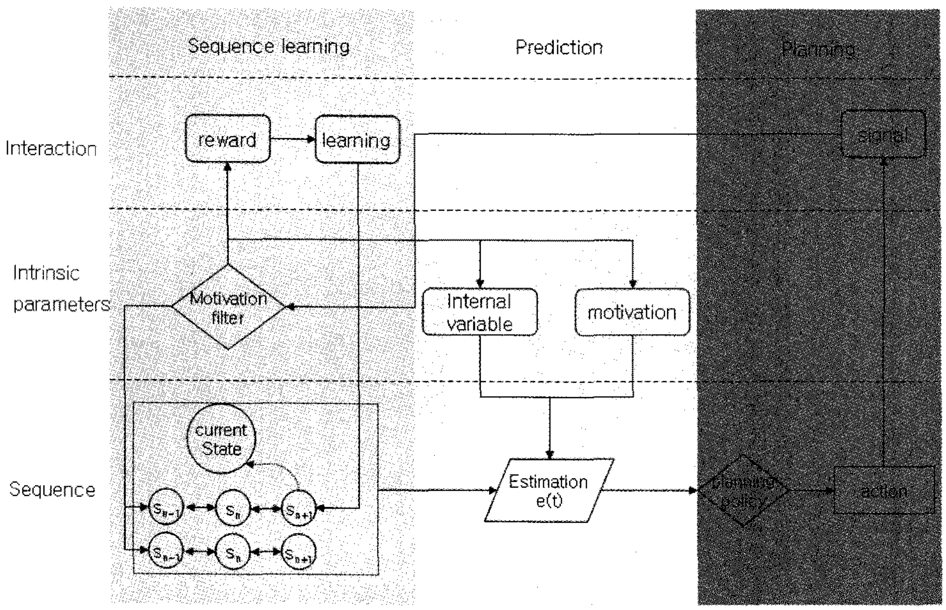


Fig. 3. Multiple sequential behavior generation framework.

dates sequence learning result. The agent then predicts the cost of each state based on internal variable and learning result after taking action. According to prediction result and perceived transition probability distribution, the agent selects an action to achieve a good state. After sending the product of internal variable changing signal to motivation filter, NPC can 'feel' which motivation is stimulated. If the sensed state is new, agent will group the new state into the behavior sequence linked with filtered motivation. Then agent calculates reward from the increment of filtered motivation. Finally the probability distribution updates directly with learning result.

6.2 Virtual library simulation environment

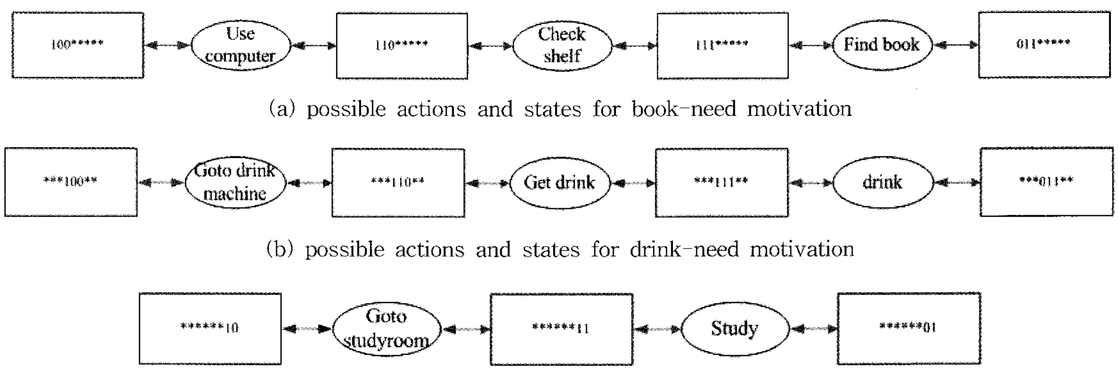
Because the sensed states in unknown environment is very complex in a form of several substate combination. We propose a method in section 4.1 helps the agent to define the sensed state autonomously. Firstly, we define several possible sensed sub-states as a set:

a. BOOK_NEED
b. HAVING_BOOK_INFORMATION
c. KNOWN_BOOK_LOCATION
d. DRINK_NEED
e. DRINK_IN_VIEW
f. OWN_DRINK
g. STUDY_NEED
h. IN_STUDY_ROOM

Fig. 4 illustrates some examples of possible transitions in the virtual library simulation after long time training. Each behavior sequence begins with a special motivation. In order to satisfy book-need motivation, after training to satisfy the motivation for book need, the virtual student selects behaviors such as using a computer, checking the shelf, and finding a book. After taking these actions, the agent's sub-state 'BOOK_NEED' is changed from 1 to 0 and receives a reward for each transition. Also the sub-states 'DRINK_NEED' and 'STUDY_NEED' decrease from 1 to 0 in learned sequences for drink-need motivation and study-need motivation separately.

Fig. 5 shows the sequence probability updating result for the book-need motivation without motivation filter. The agent senses more than ten states in this simulation, where we extract transitions of Fig. 4 (a) to discuss the performance of multiple sequential learning algorithm.

In the simulation, the agent selects action according to prediction result $g(s_i)$ calculated from equation (9). We can note that the transition probabilities p(1,2), p(2,3) and p(3,4) are higher than any other transition probability in each row. Because the agent achieves goal of satisfying book-need motivation in state 4 (011*****), which means the sub-state BOOK_NEED= 0 and satisfying book-



(a) possible actions and states for book-need motivation

(b) possible actions and states for drink-need motivation

(c) possible actions and states for study-need motivation

Fig. 4. Possible learned sequences.

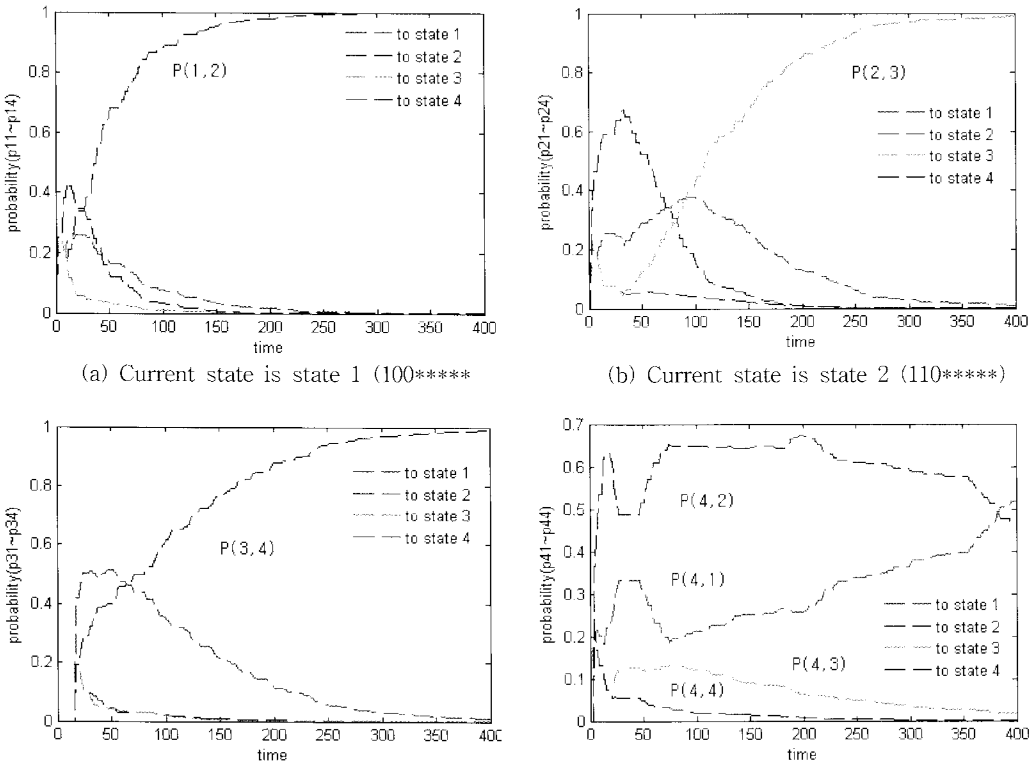
need motivation, the agent will randomly select action randomly so that the learning result is inconstant shown in fig. 5 (d).

The agent will first try to find the book's detail using a computer, and then go through all the books on a particular shelf. After going through all the books, the agent will be able to find the book.

Using our proposed prediction algorithm, the agent selects action according to prediction result $g(s_i)$. Fig. 6 shows sequence probability updating result for the book need motivation with motivation filter and prediction algorithm. We can note that the learned probabilities p(1,2), p(2,3) and p(3,4) are nearly one of after 150 times training. This method is faster than Fig. 5, which need training around 300 times.

Fig. 7 shows a comparison of results of satisfying book-need, drink need and study need motivations. We can see that with proposed motivation filter, the learning speed is faster than that without motivation filter. Meanwhile the training speed of drink-need is faster than that of book-need and study-need with motivation filter, because the number of states in drink-need sequence is smaller. But the training speed of drink-need is lower than that of book-need and study-need without motivation filter and prediction. It illustrates that the ratio of necessary states to all sensed states is smaller of drink-need motivation, so the agent takes several useless actions to try to solve this motivation without motivation filter.

Fig. 8 illustrates the demo for long-term learning action sequence for book need motivation. When the student (autonomous agent) enters library (a), the student's book-need motivation is higher than other motivations and sub-state BOOK_NEED= 1. Therefore she extracts the book-need sequence from knowledge base. Firstly



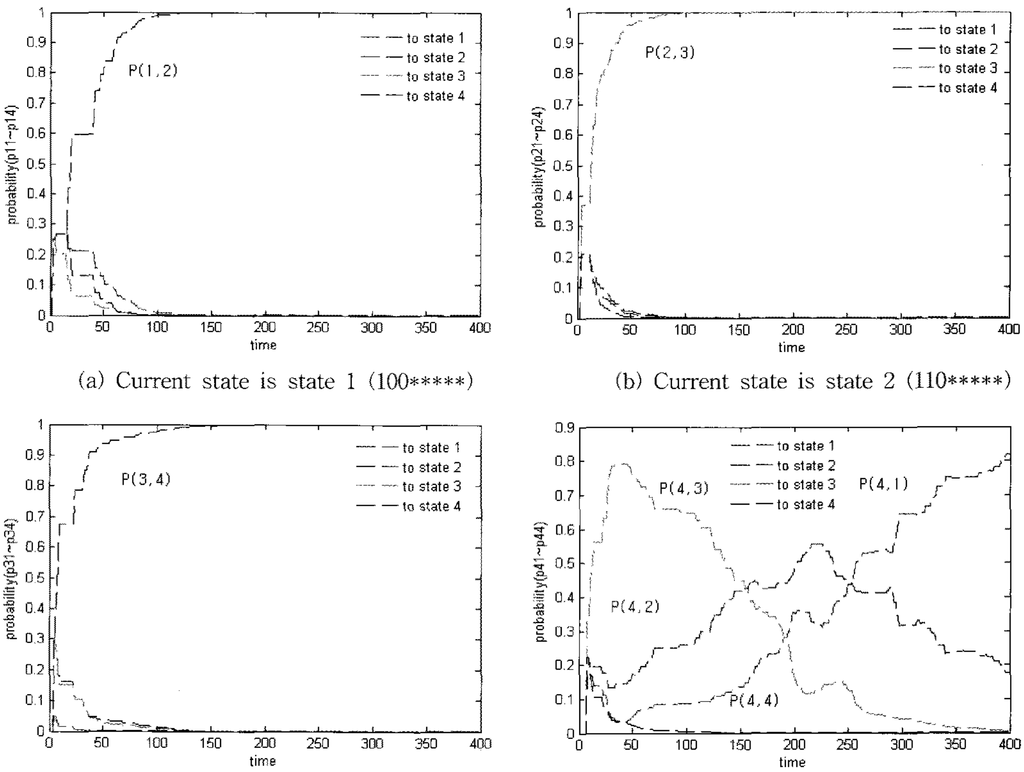
(a) Current state is state 1 (100*****

(b) Current state is state 2 (110*****)

(c) Current state is state 3 (111*****)

(d) Current state is state 4 (011*****)

Fig. 5. Sequence probability updating result without motivation filter.



(a) Current state is state 1 (100*****)

(b) Current state is state 2 (110*****)

(c) Current state is state 3 (111*****)

(d) Current state is state 4 (011*****)

Fig. 6. Sequence probability updating result with motivation filter and prediction.

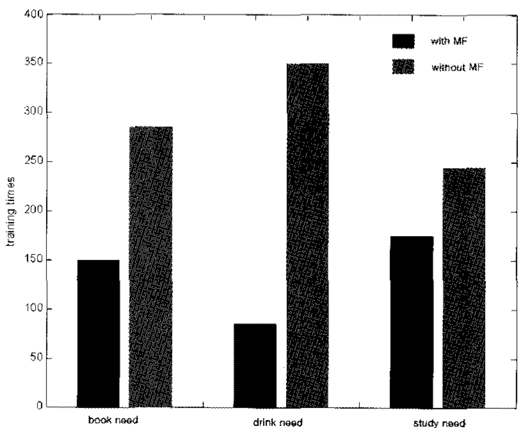


Fig. 7. training times for different condition.

she enters into the computer room and checking book information using computer (b). Then she has book's information and sub-state HAVING_BOOK_INFORMATION =1. Then, she goes to the place where the bookshelves are located and searches for the book there (c) until sub-state KNOWN_BOOK_LOCATION = 1. Finally, she goes to studying room with the book (d). By taking such sequence, she achieve the goal of satisfying book-need motivation, because the sub-state BOOK_NEED = 0.

7. CONCLUSION

This paper proposes a multiple sequence learning and prediction system for an autonomous agent to interact in unknown environment without a priori knowledge. To reduce the learning processes from large observed state space, motivation as a states' filter is used to classify the sensed states into several groups. In case that the sensed state presents a result combined with the several sub-sensed states, we suggests a state definition method. The virtual agent updates the probability

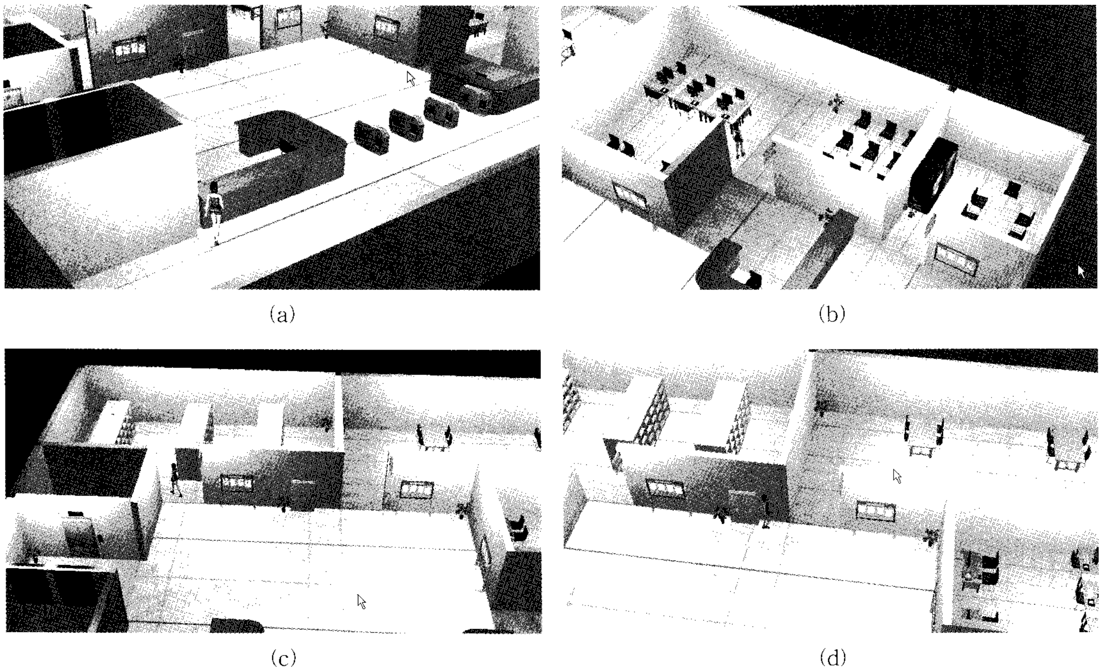


Fig. 8. Simulation screen shot for learned sequence of book need.
(a) Entering into library with book-need motivation, (b) Entering into computer room to check book information
(c) Entering into books room and finding book, (d) Taking book and going to study room

of each transition in the sequence in proportion to the learning result. We propose a prediction algorithm to estimate each state's cost, which helps a virtual agent to make an optimal decision with a high payoff.

This study can be applied to adaptive agent generation in an unknown environment, prediction in a real-time strategy game, dynamic learning for a virtual agent with a large state space, and so on.

REFERENCE

[1] F. Stan and G. Art, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents," Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, pp.21-35, 1996.

[2] I. Szita, M. Ponsen, and P. Spronck, "Effective and Diverse Adaptive Game AI," *IEEE Transactions on Computational Intelligence and AI in Games*, March 2009, pp.16-27, 2009.

[3] N.F. Xiao and S. Nahavandi, "A reinforcement learning approach for robot control in an unknown environment," IEEE International Conference on Industrial Technology, Vol.2, pp.1096-1099, 2002.

[4] S. Proper and P. Tadepalli, "Solving multiagent assignment Markov decision processes," Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, pp.681-689, 2009.

[5] W. Song, K. Cho, and K. Um, "Motivated based Multiple Behaviors Learning in Unknown Environment," *J. of Korea Multimedia Society*, Vol.12, No.12, pp.1819-1824, 2009.

[6] L.P. Kaelbling, L.M. Littman, and A.W. Moore, "Reinforcement learning: a survey," *J. of Artificial Intelligence Research*, Vol. 4, No.11, pp.237-285, 1996.

[7] T.M. Gureckis and B.C. Love, "Short-term gains, long-term pains: How cues about state aid learning in dynamic environments," *J. of*

Cognition, Vol.113, No.3, pp.293-313, 2009.

[8] R. Sun and C. Sessions, "Learning plans without a priori knowledge," *J. of Adaptive Behavior*, Vol.8, No.3/4, pp.225-253, 2000.

[9] M. Kwiatkowska, G. Norman, and D. Parker, "Game-based Abstraction for Markov Decision Processes," Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems, pp.157-166, 2006.

[10] S. Siddiqi and A. Moore, "Fast Inference and Learning in Large-State-Space HMMs," Proceedings of the 22nd International Conference on Machine Learning, Vol. 119, pp.800-807, 2005.

[11] S. Timmer and M. Riedmiller, "Learning Policies for Abstract State Spaces," 2005 IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, pp.3179-3184, 2005.

[12] L.H. Li, V. Bulitko, and R. Greiner, "Focus of attention in reinforcement learning," *J. of Universal Computer Science*, Vol. 13, No. 9, pp.1246-1269, 2007.

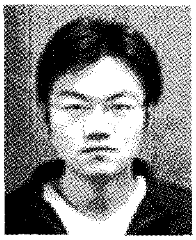
[13] Z.C. Qin and J. Lawry, "Decision tree learning with fuzzy labels," *Information Sciences*, Vol. 172, pp.91-129, 2005.

[14] E.D. Sevin and D. Thalmann, "A Motivational Model of Action Selection for Virtual Humans," *Computer Graphics International (CGI), IEEE Computer Society Press*, pp.213-220, 2005.

[15] S. Singh, A.G. Barto, and N. Chentanez, "Intrinsically Motivated Reinforcement Learning," Proceedings of Advances in Neural Information Processing Systems 17 (NIPS), 2005.

[16] K. Merrick, "Modelling Motivation for Experience-Based Attention Focus in Reinforcement Learning," PhD Thesis, University of Sydney, 2007.

[17] R.S. Sutton and A.G. Barto, "Reinforcement Learning: An Introduction," *MIT Press, A Bradford Book*, Cambridge, MA, 1998.



Song, Wei

2001. 9~2005. 7 Software College of Northeastern University, China (BS)
2006. 3~2008. 2 Dept. of Multimedia, Graduate School of Digital Image & Contents, Dongguk University.
2008. 3~present Ph.D Student, Dept. of Multimedia, Graduate School of Digital Image & Contents, Dongguk University.
Research Interests: Artificial Intelligence for Games, Game Algorithm

Cho, Kyungeun

1989. 3~1993. 2 Computer Science, Dongguk University (BS)
1993. 3~1995. 2 Computer Engineering, Dongguk University (MS)
1995. 3~2001. 8 Computer Engineering, Dongguk University (Ph.D)
2002. 3~2003. 2 Dept. of Digital Media, Fulltime Lecturer, Anyang University
2003. 3~2003. 9 Dept. of Game Engineering, Fulltime Lecturer, Youngsan University
2003. 9~2005. 8 Dept. of Computer Multimedia Engineering, Fulltime Lecturer, Dongguk University
2005. 9~2009. 8 Dept. of Game & Multimedia Engineering, Assistant Professor, Dongguk University.
2009. 9~present Dept. of Multimedia Engineering, Associate Professor, Dongguk University.
Research Interests: Artificial Intelligence for Games, Game Algorithm, Computer Vision.

Um, Kyhyun

1971. 3~1975. 2 Dept. of Applied Mathematics, Engineering College, Seoul National University (BS)
1975. 3~1977. 2 Dept. of Computer Science, Korea Advanced Institute of Science and Technology (MS)
1986. 3~1994. 2 Dept. of Computer Engineering, Graduate School, Seoul National University (Ph.D)
1978. 3~2006. 6 Dept. of Computer and Multimedia Engineering, Full Professor, Dongguk University
2006. 7~present Dept. of Multimedia Engineering, Full Professor, Dongguk University
2001. 3~2003. 2 College of Information and Industrial Engineering, Dean, Dongguk University
1995. 3~1999. 2 Information Management Institute, chief director, Dongguk University
1998. 8~2000. 7 Korea Information Science Society, SIGDB Chair
1999. 4~2005. 4 Int. Conf. on Database Systems for Advanced Applications (DASFAA) Steering Committee member
2007. 1~present Korean Multimedia Society, President
2004. 1~present Korean Game Society, Consulting member
Research Interests: Game Systems, Multimedia Applications