Journal of Korea Multimedia Society Vol. 12, No. 12, DECEMBER 2009(pp. 1819-1826)

Motivation based Behavior Sequence Learning for an Autonomous Agent in Virtual Reality

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

ABSTRACT

To enhance the automatic performance of existing predicting and planning algorithms that require a predefined probability of the states' transition, this paper proposes a multiple sequence generation system. When interacting with unknown environments, a virtual 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. We describe a sequential behavior generation method motivated from the change in the agent's state in order to help the virtual agent learn how to adapt to unknown environments. In a sequence learning process, the sensed states are grouped by a set of proposed motivation filters in order to reduce the learning computation of the large state space. In order to accomplish a goal with a high payoff, the learning agent makes a decision based on the observation of states' transitions. The proposed multiple sequence behaviors generation system increases the complexity and heightens the automatic planning of the virtual agent for interacting with the dynamic unknown environment. This model was tested in a virtual library to elucidate the process of the system.

Key words: Sequence learning, Autonomous agent, Virtual reality, Game AI

1. INTRODUCTION

When a virtual agent attempts to explore an unknown environment, he does not know how to optimally interact with the environment. Given the selective action interfaces, the agent needs to decide which action or action order can lead to a new state with high rewards and update the transition probability according to the result of the action taken.

Sequential behavior generation involves arranging the action order to accomplish the goal state or a good state. Recently, many researches have indicated that creating an autonomous agent is necessary to approach planning without a priori knowledge. The aim of this paper is to help the agent decide how to maintain balance among the internal variables without any predefined states and probabilistic transition. Given the selective action interfaces, the agent needs to decide which action order can result in a good state and estimate the transition probability based on the current state and the action taken.

After the autonomous agent attempts some fresh actions in an unknown environment, the dynamic states and probabilistic transitions are updated over time. If the number of perceptive states increases, the computation of sequence learning will involve a considerably high n-step sequence prediction.

To reduce the calculations, this study proposes

※ Corresponding Author : Kyungeun Cho, Address : (100-715) 26, Pildong 3-ga, Jung-gu, Seoul, Korea, TEL : +82-2-2260-3834, FAX : +82-2-2260-3766, E-mail : cke@dongguk.edu

Receipt date : Nov. 30, 2009, Revision date : Dec. 18, 2009
Approval date : Dec. 19, 2009

[†] 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)

※ This research was supported by MEST(Ministry of Education, Science and technology) (S-2009-A0004-00017)

a filter to group the sensed states into several subsequences by specific motivations. Then, the state space in the sequence learning becomes smaller, and the computational cost is lower.

Some approaches provide the solution to predict and learn about the next states from the previous states. Typically, an agent always selects an action or transition that may lead to a reasonable state with a higher reward.

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.

In section 2, we explain related works on sequential behavior learning and prediction. In section 3, we describe the generating system for multiple sequential behaviors. In section 4, we provide details on the experiment of the proposed system in a virtual restaurant environment. In section 5, we present the conclusion.

2. RELATED WORKS

When adapting to an unknown environment, agents must perceive their virtual world and learn how to make better decisions to achieve a good state. There have been broad theories and game AI researches on sequence behavior learning and planning.

The existing learning and planning algorithms always depend on predefined state transition probability, which is not given in unknown environment. Ron Sun [1] presents a two-stage, bottom-up process for planning without a priori knowledge. He suggests a plan extraction algorithm is 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 [2] proposes an abstraction method for the state partition of the nondeterministic MDPs which is used to make choices for two players. Siddiqi [3] introduces a Dense-Mostly-Constant (DMC) transition matrix to enhance speedups for learning an optimal state transition sequence for the observations.

When making planning, instantaneous rewards may result bad situation in future. T.M. Gureckis [4] discusses relationship between short-term and long-term rewards in dynamic planning by testing some learning model.

Without state grouping method, the computation of Ron Sun's work will be very complicated. We propose a state grouping method to reduce computation of sequence planning and learning process. Different 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 an n-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. MULTIPLE SEQUENCE BEHAVIOR

In this chapter, we propose a coherent system for multiple sequential behavior generation. The system includes sequential behavior learning, prediction, and planning processes. In the learning process, a motivation filter system to group the sensed states into a motivation-oriented sequence is adopted in order to reduce the computational cost. We suggest an algorithm to evaluate the dynamic probability of the states' transition from the generated Q-learning values. According to perceptive states transition probability, the agent can plan to achieve a state with high cost as well as maintain a balance within the internal states.

3.1 Sequence learning using motivation filter

Due to the continuous sensing of the environment, an agent may perceive a huge number of states that need to be stored in the long-term memory. To reduce the computations, we design a motivation filter to group states for multiple sequential learning due to stimulated motivation. The sensed states are grouped into different sequences by the proposed motivation filter. Each abstracted sequence can provide special motivation so as to maintain a balance between the agent's internal states. The data flow of the motivation filter is illustrated in Fig. 1.

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 his internal variables. According to the tendency (I) of his 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.

We define the characteristics tendency matrix I with the elements denoting what an agent likes or what type of estimation can lead him to a better situation. Equation (1) is an $n*n$ square matrix with all off-diagonal elements equal to zero, where n denotes the number of the agent's characteristics. The parameter i_j denotes whether the agent's state is good when the jth internal variable is high. When i_j is equal to 1, the agent has a tendency to be satisfied when the jth internal variable increases. When equal to -1, the agent has a tendency to be unsatisfied when the jth internal variable increases. The situation that i_j is equal to 0 implies that this internal variable does not influence the agent.

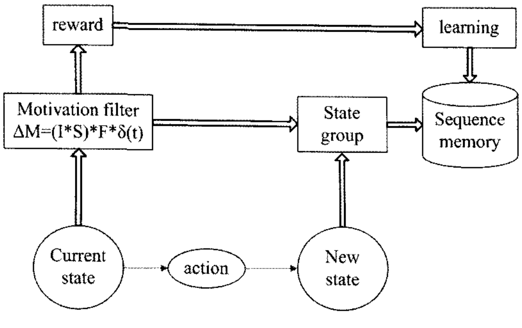


Fig. 1. Data flow chart of the motivation filter

$$I=\begin{bmatrix} i_1 & 0 & \cdots & 0 \\ 0 & i_2 & \cdots & 0 \\ \cdots & \cdots & i_j & \cdots \\ 0 & 0 & \cdots & i_n \end{bmatrix} \quad i_j \in \{1,0,-1\} \tag{1}$$

We define the stimulus from the environment as a one-column vector in eqüation (2), which represents the change in the internal variables. The parameters of S denote the changes in the internal variables after the agent acts. The element s_n affects the nth internal variable defined in equation (1).

$$S=[s_1, s_2, \ldots, s_n]' \tag{2}$$

We design the motivation filter as the $m*n$ matrix F in equation (3). In the motivation filter matrix, m denotes the number of motivation categories, and n, the number of internal variables given in equation (1). The elements equal to -1 in the n row denote that this internal variable has a relationship with the mth motivation. Agents can obtain a decreasing motivation value from the production of the input and the filter matrix. If f_{ij} equals to 0, the nth internal variable does not affect the mth motivation and will be filtered out.

$$F=\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \cdots & \cdots & f_{ij} & \cdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix} \quad f_{mn} \in \{0,-1\} \tag{3}$$

The output of the motivation filter in equation

(4), the product of $F \bullet (I \bullet S)$, generates which motivations are stimulated from the agent's changing internal variables. If the mth element in the output is negative, the resultant state will be inserted into the mth sequences, because this action can reduce the filtered mth motivation. Because the action taken can affect more than one motivation, the new state may insert more than one sequence according to the filtered result.

$$F \cdot (I \cdot S) = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \cdots & \cdots & f_{ij} & \cdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix} \begin{bmatrix} i_1 \cdot s_1 \\ i_2 \cdot s_2 \\ \cdots \\ i_n \cdot s_n \end{bmatrix} = \begin{bmatrix} \sum_{t=1}^{n} f_{1t} \cdot i_t \cdot s_t \\ \sum_{t=1}^{n} f_{2t} \cdot i_t \cdot s_t \\ \cdots \\ \sum_{t=1}^{n} f_{mt} \cdot i_t \cdot s_t \end{bmatrix} \quad (4)$$

Equation (5) represents the characteristics tendency matrix of a virtual agent, which is an instance of equation (1). All the diagram elements denote the internal variables of anger, thirst, hunger, sleepiness, relaxation, and safety. We know that when the degree of hunger is low, the situation of the agent is better, and the relevant parameter is -1. The internal variables of anger and thirst also follow this rule. We define that if the internal variable for relaxation increases, the situation of the guest improves and the relaxation parameter is 1. We define that the agent never cares about sleepiness and safety in the restaurant, and these elements are equal to 0.

$$I_{guest} = \begin{array}{c} \begin{bmatrix} -1 & & & & & 0 \\ & -1 & & & & \cdots \\ & & -1 & & & \\ & & & 0 & & \\ & \cdots & & & 1 & \\ 0 & & & & & 0 \end{bmatrix} \end{array} \begin{array}{c} \textit{anger} \\ \textit{thirst} \\ \textit{hunger} \\ \textit{sleepiness} \\ \textit{relaxation} \\ \textit{safety} \end{array} \quad (5)$$

After the guest makes a call, if the waitress serves him, he will receive an internal changing signal $S_{calling}$ denoted by equation (6). This stimulus signal indicates that the values of anger and relaxation will decrease by -5. Because the guest orders dinner during the serving event, the thirst and hunger variables will decrease by -1. However, this action will cause a slight increase in the sleepiness variable.

$$S_{calling} = \begin{bmatrix} -5 \\ -1 \\ -1 \\ 1 \\ -5 \\ 0 \end{bmatrix} \begin{array}{c} \textit{anger} \\ \textit{thirst} \\ \textit{hunger} \\ \textit{sleepiness} \\ \textit{relaxation} \\ \textit{safety} \end{array} \quad (6)$$

Equation (7) denotes the guest's motivation filter matrix, wherein the filtered motivation choices are $M[1]$ = serving need, $M[2]$ = food need, and $M[3]$ = rest need. The matrix indicates that the changes in the anger, thirst, and hunger variables affect the service need, thirst and hunger variables affect the food need, and sleep and relaxation variables affect the rest need.

$$F = \begin{bmatrix} -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 \end{bmatrix} \begin{array}{c} \textit{servingneed} \\ \textit{foodneed} \\ \textit{restneed} \end{array} \quad (7)$$

Equation (8) denotes the filtered result of the change in each guest's motivation after receiving service on call. The product result implies that the calling action can maintain a balance in the motivations of the serving and food needs, but aggravate the motivation for the rest need. Thus, the calling action and the new observed state are added into the sequences linked with the motivation of the serving and food needs.

$$F \bullet I_{guest} \bullet S_{calling} = [-7, -2, 5]' \quad (8)$$

3.2 Sequence updating method

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.

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'.

The transition probability of the sequence is initialized evenly by equation (9). The element $p_a(i,j)$ represents the transition probability from state i to state j.

$$p_a(i,j)=\frac{T_a(i,j)}{\sum_k\sum_a T_a(i,k)} \tag{9}$$

where $T_a(i,k)=1$, if the transition from state i to state k happens by action a.

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.

After taking an action, the agent may acquire a signal for changing his motivation. By calculating the action's contribution to the stimulated motivation, the reward is estimated as follows:

$$R(s_t,a)=f(\Delta M) \tag{10}$$

where s_t denotes the state at time t, and a, the action taken at state s_t. The reward $R(s_t,a)$ is evaluated according to the motivation's variation ΔM, derived from equation (4).

With the calculated reward, the Q-learning algorithm (11) 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 [5]:

$$Q(s_t,a_t)=Q(s_t,a_t)+\alpha[R(s_t,a_t)+\gamma\max Q(s_{t+1},a_t)-Q(s_t,a_t)] \tag{11}$$

The coefficient α denotes the learning rate $(0\le\alpha\le1)$, and γ, the discount factor $(0\le\gamma\le1)$. 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:

$$p_a'(s_t,s_{t+1})=\beta kQ(s_t,a)p_a(s_t,s_{t+1})+(1-\beta)p_a(s_t,s_{t+1}) \tag{12}$$

where β denotes the discount factor of the Q-value influence. In equation (12), $\beta kQ(s_t,a)p_a(s_t,s_{t+1})$ denotes the discount value of the learned probability and $(1-\beta)p_a(s_t,s_{t+1})$, the part for the previous perceived probability. The transition probability of the next state is updated with these two parts. Because the agent cannot determine the relationship between the probability and the Q-value at the beginning of the learning process, we suggest a coefficient k as the ratio of the Q-value to the probability.

Based on the quality of Markov Chains, the transition probability from s_t to s_{t+1} can be finally calculated by equation (13) as follows:

$$p_a'(s_t,s_{t+1})=\beta\frac{Q'(s_t,a)p_a(s_t,s_{t+1})}{\sum_{j\in S}\sum_{\hat{a}\in A}Q'(s_t,\hat{a})p_{\hat{a}}(s_t,s_j)}+(1-\beta)p_a(s_t,s_{t+1}) \tag{13}$$

According to the calculations for the Q-values of the optional actions and the current stage's probability distribution, the agent can estimate the transition probability $p_a'(s_t,s_{t+1})$ for the next stage.

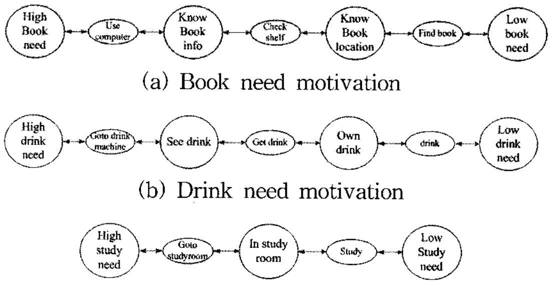
In the behavior planning process, we propose that the agent selects an action randomly according to calculated probability distribution $p_a'(s_t,s_{t+1})$ to achieve a good state with the highest cost.

4. SYSTEM EXPERIMENT AND ANALYSIS

To elucidate the mechanism of the stochastic process learning system using a 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 thirst, hunger, and a tendency to study.

Fig. 2 illustrates some examples of possible transitions in the virtual library simulation. Each planning sequence begins with a special motivation. (a) illustrates the sequence of the motivation for book need. In order to decrease the motivation for book need, the virtual agent can take actions such as using a computer, checking the shelf, and finding a book. After taking these actions, the agent may complete the possible transition and receive a reward '*r*' from the environment. (b) and (c) illustrate the sequences of drink and study need motivations separately.

Fig. 3 shows the training result of the action selection probabilities from the knowing-book-information state to the knowing-book-location state in the book need sequence, which are denoted by $p_{checkshelf}(2,3)$, $p_{usecomputer}(2,3)$, and $p_{findbook}(2,3)$, respectively. From Fig. 3, which illustrates the training curves for 0~400 times, we can note that $p_{checkshelf}(2,3)$ is higher than any other transition probability. It conforms to the reality that if a virtual agent has information about the book, he will first try to find the book on a particular shelf rather than start finding the book randomly.



(a) Book need motivation

(b) Drink need motivation

(c) Study need motivation

Fig. 2. Possible actions and states for different motivations

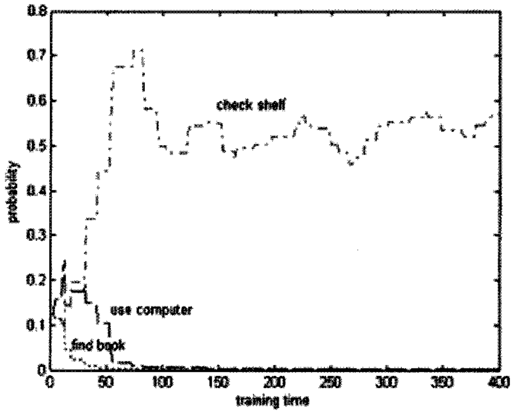


Fig. 3. Probability p(2,3) of the training performance in the virtual library

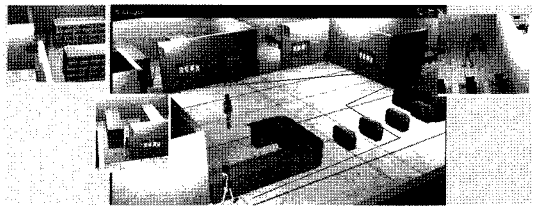


Fig. 4. Simulation screenshot for the learned sequence of book need

Fig. 4 illustrates the demo for long-term learning action sequence for book need motivation., the agent attempts to check information on the book in the computer room. Then, she goes to the place where the bookshelves are located and searches for the book there.

5. CONCLUSION

This paper proposes a multiple sequence behavior generation system for an autonomous agent to learn, predict, and plan without a priori knowledge. When interacting with an unknown environment, the observed state space is very large due to the dynamic perception. To reduce the computation for the learning and prediction processes, motivation as a states' filter is used to classify the sensed states into several subsequences. After receiving a reward from the environment, the virtual agent evaluates the action taken using a Q-learning algorithm. The probability of each transition in the sequence is updated in proportion to the learning result. According to the perception result, a virtual agent can make an optimal decision with a high payoff. The proposed system increases the complexity and heightens the automatic planning of the virtual agent.

In case that the sensed state presents a result combined with the several subsense states, we will enhance this learning system with detailed definitions of the states in the future.

REFERENCES

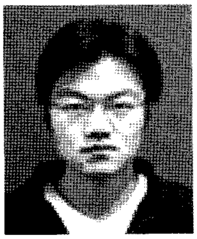
[1] R. Sun and C. Sessions, "Learning plans without a priori knowledge," Journal of Adaptive Behavior, Vol.8, No.3, pp. 225-253, 2000.

[2] 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.

[3] 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.

[4] T.M. Gureckis and B.C. Love, "Short-term gains, long-term pains: How cues about state aid learning in dynamic environments," Cognition, Vol.113, No.3, pp. 293-313, 2009.

[5] 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 Game & 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 Game and 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