SPOKEN DIALOGUE SYSTEM FOR PRODUCT RECOMMENDATION USING HIERARCHICAL POMDP

Yoji Yamada *Tetsuya Takiguchi* *Yasuo Ariki*

School of System Informatics, Kobe University
Nada-ku, Kobe, 657-8501, Japan
`y.yamada@me.cs.scitec.kobe-u.ac.jp`

ABSTRACT

We present a spoken dialogue system developed for product recommendation, which uses partially observable Markov decision processes (POMDPs) hierarchically (H-POMDPs). Aiming at the efficient dialogue system, user utterance log is utilized and the global POMDP (G-POMDP) is newly introduced to control the entire H-POMDP. We developed the system as a smartphone application and evaluated it by real users.

1. INTRODUCTION

Various kinds of spoken dialogue systems have been developed in recent years. A robot with a chat type dialogue ability and a car navigation system for searching the destination [1] are the typical examples. Such spoken dialogue systems can be divided into two types: task-oriented or non task-oriented. When constructing a task-oriented system, reinforcement learning such as a POMDP is often used [2][3][4]. However, a reinforcement learning algorithm for spoken dialogue system generates a large number of states. Inevitably, it takes amount of time for the computation. Therefore, researchers have been proposed several methods concerning with the POMDP's state [5][6]. Among them, we especially focus on H-POMDP [7][8][9] which divides the large task into several subtasks to reduce the number of states and the computational complexity.

In this paper, we apply H-POMDP to a product recommendation system. Various kinds of products (e.g. foods, books and electrical appliances) exist in the real world. For example, we go to a bookstore when we want to read some novels. However, it is sometimes difficult to decide which ones to buy because there are too many choices. In this case, we will ask a clerk about recommended books. This is our motivation to develop a book recommendation system such as the clerk using H-POMDP.

H-POMDP divides a large number of POMDP's states, actions and observations. However, if observations are divided, the types of user utterance accepted by one of the divided small POMDPs are limited and this leads to the dialogue inflexibility. To solve this problem, we propose two functions. The first is utilization of the user utterance log in which all user utterances are recorded as observations. The second is the G-POMDP which can understand every H-POMDP's observation. They both enable a user to talk with the system more freely and contribute to develop the efficient dialogue system.

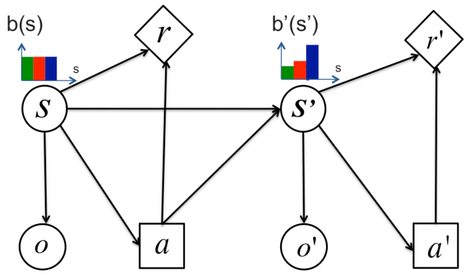

Fig. 1. Structure of POMDP

2. HIERARCHICAL POMDP

An overview of the POMDP is depicted in Figure 1. The system takes an action a in a state s, gets the reward r, and transits to the next state s'. It observes the user utterance o', and updates the belief state $b(s)$ to the $b'(s)$ [10]. However, if the number of the states is increased, it will take a heavy processing time. Therefore, as shown in Figure 2, we construct POMDP in a hierarchical way called as H-POMDP. The task is divided into several subtasks, each of which is modeled by local POMDP, and local policy is optimized by the Point Based Value Iteration in the local POMDP [11].

The system starts from the POMDP1 locating at the top of H-POMDP. The POMDP1 updates the belief state, achieves the POMDP1's goal state which is synonymous with the subtask's goal, and transits to the next POMDP2 or POMDP3. The system continues the transition from POMDP to POMDP and achieves the most significant POMDP's goal state which is synonymous with the task's goal. In this way, a more complex task can be accomplished by combining subtasks in a

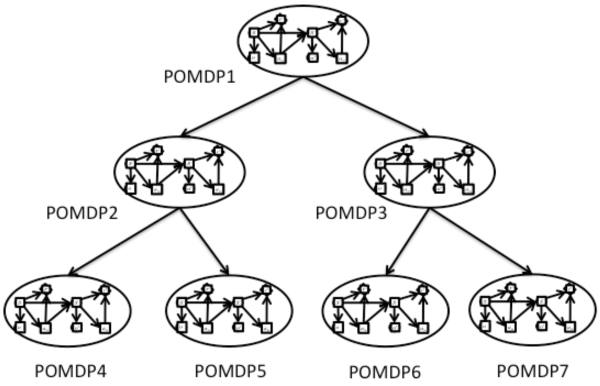

Fig. 2. Structure of H-POMDP

hierarchical way.

3. SPOKEN DIALOGUE SYSTEM FOR PRODUCT RECOMMENDATION

3.1. Concept of the product recommendation

We apply the H-POMDP to a spoken dialogue system for product recommendation. The product means articles such as books, CDs, and groceries. In this paper, we set a task of recommending books. As shown in Figure 3, books are classified into categories such as novels, magazines and textbooks. Furthermore, novels are classified into categories such as mystery novels and history novels. Additionally, it is often the case that the popular books depend on the reader's age. An user answer to the system's questions can narrow down such branching conditions. The system can estimate the user intention and eventually recommend one book.

3.2. Example of dialogue

An example of the dialogue is shown below.

Sys : What kind of book do you want? (POMDP1)
Usr : I want to read a novel.
Sys : What genre of novel do you want?(POMDP2)
Usr : Mystery.
Sys : How old are you? (POMDP5)
Usr : Twenty two.
Sys : I beg your pardon? (POMDP5)
Usr : Twenty two.
Sys : So, I recommend this book.

Table 1. List of POMDPs

Depth 0	POMDP1 s : novel, magazine, textbook a : confirm(ask the kinds of book), transit to POMDP2 or POMDP3 or POMDP4 o : novel, magazine, textbook
Depth 1	POMDP2 s : mystery, history a : confirm(ask the genre of novel)、 transit to POMDP5 or POMDP6 o : mystery, history POMDP3 s : fashion, travel a : confirm(ask the genre of magazine), transit to POMDP7 or POMDP8 o : fashion, travel POMDP4 s : English、 mathematics a : confirm(ask the subjects), transit to POMDP9 or POMDP10 o : English, mathematics
Depth 2	POMDP5 s : teen, twenties, thirties a : confirm(ask the age), recommend a book(teen or twenties or thirties) o : 10, 20, 30 … POMDP10 s : middle school, high school, university a : confirm(ask the grade), recommend a book(middle or high or university) o : middle, high, university

3.3. Application of H-POMDP

We apply H-POMDP to this task. First, as described in POMDP1's structure shown in Table 1, the states are {"novel", "magazine", "textbook"}, the observations are {"novel", "magazine", "textbook"}, the actions are {"confirm", "transit to POMDP2", "transit to POMDP3", "transit to POMDP4"}. The action "confirm" means the system utterance that requests the user utterance. For example, the system asks the user the type of books, and observes the user utterance "I want to read a novel". The POMDP1 updates the belief state and chooses the action "transit to the POMDP2". The system asks the user the genre of novel as an action in POMDP2.

Thus, applying the tree structure shown in Figure 3 to the H-POMDP, 10 POMDPs are created as shown in Table 1. If

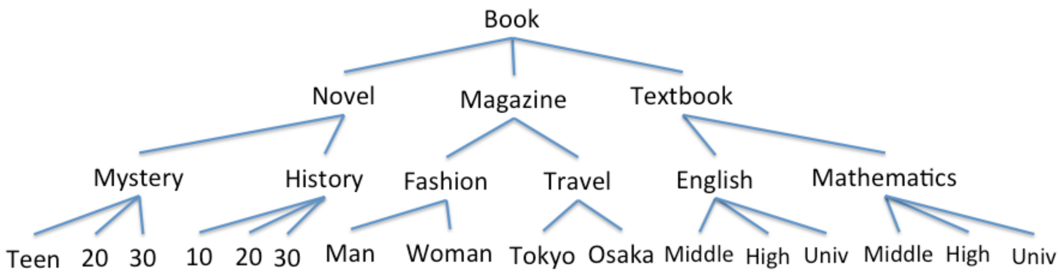

Fig. 3. Classification of books

the action "recommend a book" is selected in a POMDP at the bottom depth, the system recommends the user a book and finishes the task.

4. FLEXIBILITY OF THE USER UTTERANCE

There are problems in the H-POMDP. First, if the user says "I want to read mystery novel" in the POMDP1, the system ignores the observation "mystery" because it is not included in the POMDP1's observation. Second, once a POMDP transited to a lower POMDP, it can not transit to other POMDP at the same depth. To solve these problems, we propose the user utterance log function and the G-POMDP that can control the entire H-POMDP. They can both improve the user utterance flexibility and construct the efficient spoken dialogue system.

4.1. User utterance log function

The system maintains all user utterances $\{o_1, o_2, ..., o_n\}$. In n-th turn of the dialogue, the system checks utterances $\{o_1, o_2, ..., o_{n-1}\}$ before it performs the action a_n. If an expected user's response is included in the utterance log, the system chooses it as the observation o_n and updates the belief state without listening the n-th user utterance.

An example of this system is as follow:

Sys : What kind of book do you want? (POMDP1)
Usr : I want to read a mystery novel.
Sys : How old are you? (POMDP5)
Usr : Twenty two.
Sys : So, I recommend this book.

Even if the user has spoken the both words "mystery" and "novel" in POMDP1, an information "mystery" is not ignored but recorded in the utterance log, and used in POMDP2. The question "ask the genre" in POMDP2 is omitted and then the number of dialogues is reduced.

4.2. Global POMDP

To transit to a higher or lower POMDP more freely, we construct the special POMDP called G-POMDP to control the entire H-POMDP. The random variables of the G-POMDP are as follows:

$s \in S_g$: {"POMDP1", "POMDP2", ... ,"POMDP10"}
$a \in A_g$: { "Confirm", "Transit to POMDP1", ... , "Transit to POMDP10"}
$o \in O_g$: {"Novel", "Magazine", ... , "University"}

The S_g includes every POMDP in H-POMDP, and the O_g includes all observations in H-POMDP. A G-POMDP's policy (global policy) is learned from these random variables.

A global policy selects an action (a_g) at each system's turn, and a local policy of H-POMDP also selects an action (a_h). Therefore, these actions need to be compared to select more optimal one. To estimate the value of actions, we set the evaluation functions as below.

the value of a_g : $w * k(n, a_g)$
the value of a_h : $(1 - w) * k(n, a_h)$

w is a weight to adjust which policy needs to be focused ($0 < w < 1$). When policy selects the action a at the present POMDPn, the evaluated value $k(n, a)$ is given as shown in Table 2. In Table 2, the actions a_g and a_h are both aligned at the first row. Recommending a book in a POMDP at the bottom depth gets value 10. Transiting to the one lower level POMDP gets value 5. Transiting to other POMDP in the same level or two lower level POMDP gets value 1. Confirming gets value -1, and other actions gets value -10. Comparing these two evaluation functions for a_g and a_h, the action which gives a larger value is selected.

An example of this system is shown below.

Sys : What kind of book do you want? (POMDP1)
Usr : I want to read a mystery.
Sys : How old are you? (POMDP5)
Usr : Twenty two.
Sys : So, I recommend this book.

In first turn of the above dialogue, $w = 0.7$, $n = 1$, a_g is transition to the POMDP5 ($k(1, a_g) = 1$) and a_h is confirmation ($k(1, a_h) = -1$). The value of a_g (0.7) is larger than the value of a_h (-0.3), and then a_g is selected. Therefore, even if the user does not utter the POMDP1's observation "novel", the system can transit to the lower POMDP.

Table 2. Evaluated values to each policy

Current POMDP n	Action a							
	Transit to POMDP1	POM.2	...	POM.5	...	POM.10	Confirm	Recommend
POMDP1	-10	5	...	1	...	1	-1	-10
POMDP2	-10	-10	...	5	...	1	-1	-10
:	:	:	:	:	:	:	:	:
POMDP9	-10	-10	...	-10	...	1	-1	10
POMDP10	-10	-10	...	-10	...	-10	-1	10

5. EVALUATION OF SYSTEMS

5.1. Details of experiments

We implemented the proposed spoken dialogue systems as smartphone applications and evaluated it by 13 real users. The implemented systems are following four types:

Sys1:H-POMDP
Sys2:H-POMDP + user utterance log
Sys3:H-POMDP + G-POMDP($w = 0.3$)
Sys4:H-POMDP + G-POMDP($w = 0.7$)

For comparison, the following three evaluations were carried out: task achievement rate TA, averaged number of answers in a dialogue AD, and system performance SP. SP is defined as follows:

$$SP = \frac{TA}{AD}$$

Experimental conditions are described below.

1. The user answers the system's question.
2. The user choses words described in Fig. 3 ("I want to read novel", "English book")
3. If the system could not recommend a book within 5 answers in a dialogue, the task failed.

5.2. Results of experiments

Experimental results are shown in Table 3. It indicates that the Sys4 performs well because the G-POMDP includes many kinds of observations, and it can deal with various user utterances. For example, some user firstly say "I want to go travel" or "I want to study English", but it does not include the POMDP1's observations such as "magazine" or "textbook". Therefore, the Sys4 can only observe the utterances such as "travel" or "English".

Table 3. Evaluation results of systems

	Sys1	Sys2	Sys3	Sys4
TA	0.85	0.92	0.85	0.96
AD	3.73	3.27	3.85	3.15
SP	0.23	0.28	0.22	0.30

6. CONCLUSIONS

We proposed a spoken dialoge system for product recommendation by using H-POMDP. This system can make user's vague desire clean through dialogue, and recommend a product the user really wants. We showed the user utterance log and the G-POMDP can deal with the various observations and improve the system performance.

However, we have additional tasks to indicate the effectiveness of our method. First, clarifying differences between our proposed system and hidden information state model which uses partition of user state[5]. Second, comparing the computing cost between the H-POMDP and normal POMDP. We plan to conduct these experiment as a future work.

Besides, we plan to set a large number of POMDPs in parallel, and optimize the transitions from POMDP to POMDP by the book data [12].

7. REFERENCES

[1] Tetsuya Takiguchi Yasuhide Kishimoto and Yasuo Ariki, "Spoken dialogue manager in car navigation system using partially observable markov decision processes with hierarchical reinforcement learning," in *Institute of Electronics Information and Communication Engineers*, 2010, vol. 110, pp. 49–54.

[2] Yasuhiro Minami, "Dialogue control using partially observable markov decision process," in *Acoustical Society of Japan*, 2011, vol. 67, pp. 482–487.

[3] Jason D. Williams and Steve Young, "Partially observable markov decision processes for spoken dialog systems," in *Computer Speech and Language*, 2007, vol. 21, pp. 393–422.

[4] Koichiro Yoshino and Tatsuya Kawahara, "Information navigation system based on pomdp that tracks user focus," in *SIGdial*, 2014, pp. 32–40.

[5] Simon Keizer Francois Mairesse Jost Schatzmann Blaise Thomson Steve Young, Milica Gasic and Kai

Yu, "The hidden information state model: A practical framework for pomdp-based spoken dialogue maganement," in *Computer Speech and Language*, 2010, vol. 24, pp. 150–174.

[6] Jason D. Williams and Steve Young, "Scaling up pomdps for dialogue management: The "summary pomdp"method," in *IEEE workshop on Automatic Speech Recognition and Understanding*, 2005, pp. 177–182.

[7] Tetsuya Takiguchi Yasuhide Kishimoto and Yasuo Ariki, "Spoken dialogue manager using partially observable markov decision processes with hierarchical reinforcement learning," in *Institute of Electronics Information and Communication Engineers*, 2010, vol. 110, pp. 121–126.

[8] Nicholas Roy Joelle Pineau and Sebastian Thrun, "A hierarchical approach to pomdp planning and execution," in *ICML Workshop on Hierarchy and Memory in Reinforcement Learning*, 2001, vol. 110, pp. 121–126.

[9] I. Kruijff-Korbayova H. Cuayahuitl and N. Dethlefs, "Non-strict hierarchical reinforcement learning for interactive systems and robots," in *ACM Transactions on Interactive Intelligent Systems*, 2014, vol. 4.

[10] Blaise Thomson Steve Young, Milica Gasic and Jason D. Williams, "Pomdp-based statistical spoken dialog systems: A review," in *IEEE*, 2013, vol. 101, pp. 1160–1179.

[11] Geoffrey J. Gordon Joelle Pineau and Sebastian Thrun, "Point-based value iteration: An anytime algorithm for pomdps," in *International Joint Conferences on Artificial Intelligence*, 2003, pp. 1025–1032.

[12] Pirros Tsiakoulis Milica Gaŝić Matthew Henderson Dongho Kim, Catherine Breslin and Steve Young, "Inverse reinforcement learning for micro-turn management," in *INTERSPEECH*, 2014, pp. 328–332.