8

How to Recommend Preferable Solutions of a User in Interactive Reinforcement Learning?

Tomohiro Yamaguchi[1], Takuma Nishimura[2] and Kazuhiro Sato[3]
[1, 2]Nara National College of Technology,
[2]currently Kyoto University,
[3]FANUC Ltd
Japan

1. Introduction

In field of robot learning (Kaplan et al., 2002), interactive reinforcement learning method in that reward function denoting goal is given interactively has worked to establish the communication between a human and the pet robot AIBO. The main feature of this method is the interactive reward function setup which was fixed and build-in function in the main feature of previous reinforcement learning methods. So the user can sophisticate reinforcement learner's behavior sequences incrementally.
Shaping (Konidaris & Barto, 2006; Ng et al., 1999) is the theoretical framework of such interactive reinforcement learning methods. Shaping is to accelerate the learning of complex behavior sequences. It guides learning to the main goal by adding shaping reward functions as subgoals. Previous shaping methods (Marthi, 2007; Ng et al., 1999) have three assumptions on reward functions as following;

1. Main goal is given or known for the designer.
2. Subgoals are assumed as shaping rewards those are generated by potential function to the main goal (Marthi, 2007).
3. Shaping rewards are policy invariant (not affecting the optimal policy of the main goal) (Ng et al., 1999).

However, these assumptions will not be true on interactive reinforcement learning with an end-user. Main reason is that it is not easy to keep these assumptions while the end-user gives rewards for the reinforcement learning agent. It is that the reward function may not be fixed for the learner if an end-user changes his/ her mind or his/ her preference. However, most of previous reinforcement learning methods assumes that the reward function is fixed and the optimal solution is unique, so they will be useless in interactive reinforcement learning with an end-user.
To solve this, it is necessary for the learner to estimate the user's preference and to consider its changes. This paper proposes a new method how to match an end-user's preference solution with the learner's recommended solution. Our method consists of three ideas. First, we assume *every-visit-optimality* as the optimality criterion of preference for most of end-users. Including this, section 2 describes an overview of interactive reinforcement learning in our research. Second, to cover the end-user's preference changes after the reward function is given by the end-user, interactive LC-learning prepares *various policies* (Satoh &

Yamaguchi, 2006) by generating variations of the reward function under *every-visit-optimality*. It is described in section 3. Third, we propose *coarse to fine recommendation* strategy for guiding the end-user's current preference among *various policies* in section 4.
To examine these ideas, we perform the experiment with twenty subjects to evaluate the effectiveness of our method. As the experimental results, first, a majority of subjects prefer each *every-visit* plan (visiting all goals) than the *optimal* plan. Second, the majority of them prefer *shorter* plans, and the minority of them prefer *longer* plans. We discuss the reason why the end-users' preferences are divided into two groups. These are described in section 5. In section 6, the search ability of interactive LC-learning in a stochastic domain is evaluated. Section 7 describes relations between our proposed solutions and current research issues on recommendation systems. Finally, section 8 discusses our conclusions and future work.

2. Interactive reinforcement learning

This section describes the characteristics on interactive reinforcement learning in our research, and shows the overview of our system.

2.1 Interactive reinforcement learning with human

Table1 shows the characteristics on interactive reinforcement learning. In reinforcement learning, an optimal solution is decided by the reward function and the optimality criteria. In standard reinforcement learning, an optimal solution is fixed since both the reward function and the optimality criteria are fixed. On the other hand, in interactive reinforcement learning, an optimal solution may change according to the interactive reward function. Furthermore, in interactive reinforcement learning with human, various optimal solutions will occur since the optimality criteria depend on human's preference.
Then the objective of this research is to recommend preferable solutions of each user. The main problem is how to guide to estimate the user's preference? Our solution consists of two ideas. One is to prepare various solutions by *every-visit-optimality* (Satoh & Yamaguchi, 2006), another is the *coarse to fine recommendation* strategy (Yamaguchi & Nishimura, 2008).

Type of reinforcement learning	an optimal solution	reward function	optimality criteria
standard	fixed	fixed	fixed
interactive	may change	interactive	fixed
interactive with human	various optimal	may change	human's preference

Table 1. Characteristics on interactive reinforcement learning

2.2 Overview of the plan recommendation system

Fig. 1 shows an overview of the plan recommendation system. When a user input several goals to visit constantly as his/ her preference goals, they are converted to the set of rewards in the plan recommendation block for the input of interactive LC-learning (Satoh & Yamaguchi, 2006) block. After *various policies* are prepared, each policy is output as a round plan for recommendation to the user. The user comes into focus on his/ her preference criteria through the interactive recommendation process. The interactive recommendation will finish after the user decides the preference plan. Next section, interactive LC-Learning block is described.

Fig. 1. The plan recommendation system

2.3 Interactive LC-Learning block

Fig. 2 shows an overview of interactive LC-Learning (Satoh & Yamaguchi, 2006) block that is extended model-based reinforcement learning. In Fig. 2, our learning agent consists of three blocks those are model identification block, optimality criterion block and policy search block. The details of these blocks are described in following section. The novelty of our method lies in optimality criterion as *every-visit-optimality* and the method of policy search collecting *various policies*.

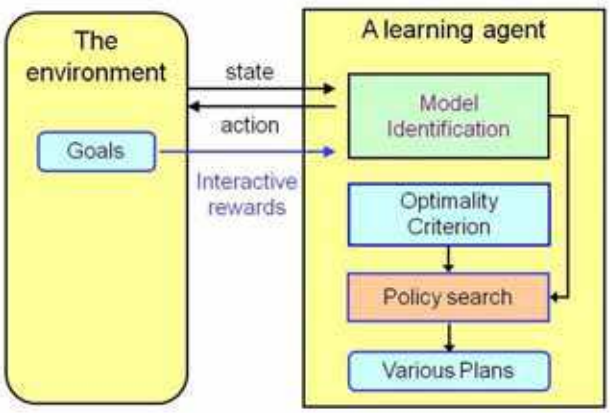


Fig. 2. Interactive LC-Learning block

2.3.1 Model identification

In model identification block, the state transition probabilities $P(s' \mid s,a)$ and reward function $R(s,a)$ are estimated incrementally by observing a sequence of (s,a,r). Note that s is an observed state, a is an executed action, and Rw is an acquired reward. This estimated model is generally assumed Markov Decision Processes (MDP) (Puterman, 2006). MDP model is defined by following four elements.

1. Set of states: $S = \{s_0, s_1, s_2, \cdots, s_n\}$
2. Set of actions: $A = \{a_0, a_1, a_2, \cdots, a_m\}$
3. State transition probabilities: $P(s' \mid s,a)$ probability of occurring state s' when execute action a at state s.
4. Reward function: $R(s,a)$ acquired reward when execute action a at state s.

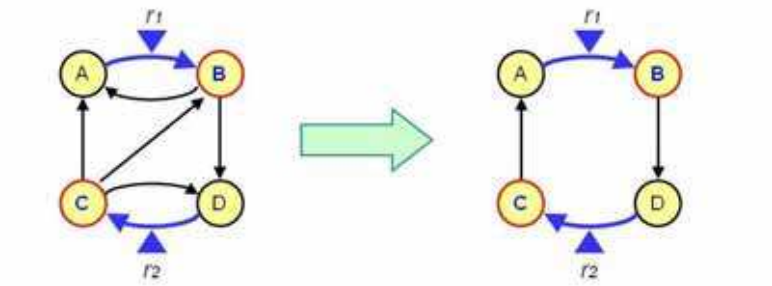
2.3.2 Optimality criterion

Optimality criterion block defines the optimality of the learning policy. In this research, a policy which maximizes average reward is defined as an *optimal* policy. Eq. (1) shows the definition of average reward.

$$g^{\pi}(s) \equiv \lim_{N \to \infty} E\left(\frac{1}{N} \sum_{t=0}^{N-1} r_t^{\pi}(s) \right) \tag{1}$$

where N is the number of step, $r_{\tau}^{\pi}(s)$ is the expected value of reward that an agent acquired at step t where policy is π and initial state is s and $E(\ )$ denotes the expected value. To simplify, we use *gain-optimality* criterion in LC-Learning (Konda et al., 2002a). In that, average reward can be calculated by both the expected length of a reward acquisition cycle and the expected sum of the rewards in the cycle.

Then we introduce *every-visit-optimality* as the new learning criterion based on average reward. *Every-visit-optimal* policy is the *optimal* policy that visits every reward in the reward function. For example, if the reward function has two rewards, the *every-visit-optimal* policy is the largest average reward one which visits both two rewards. Fig.3 shows the example of an *every-visit-optimal* policy with two rewards.



(a) a MDP with two rewards r_1 and r_2 (b) the every-visit-optimal policy

Fig. 3. An *Every-visit-optimal* policy with two rewards

2.3.3 Policy search

Policy search block searches *every-visit-optimal* policies on an identified model according to optimality of policies. Each policy is converted to a round plan by extracting a cycle. The detail of this block is described in next section.

3. Preparing various round plans

This section describes the definition of various round plans and the method for searching various policies.

3.1 Illustrated example

To begin with, we show an illustrated example. Fig.4 shows an overview of preparing various round plans within two rewards. When a MDP has two rewards as shown in Fig.4 (a), then $2^2 - 1$, three kinds of every-visit-optimal policies are prepared (Fig.4 (b)). Each policy is converted to a round plan by extracting a reward acquisition cycle (Fig.4 (c)), since each policy is consists of a reward acquisition cycle and some transit passes.

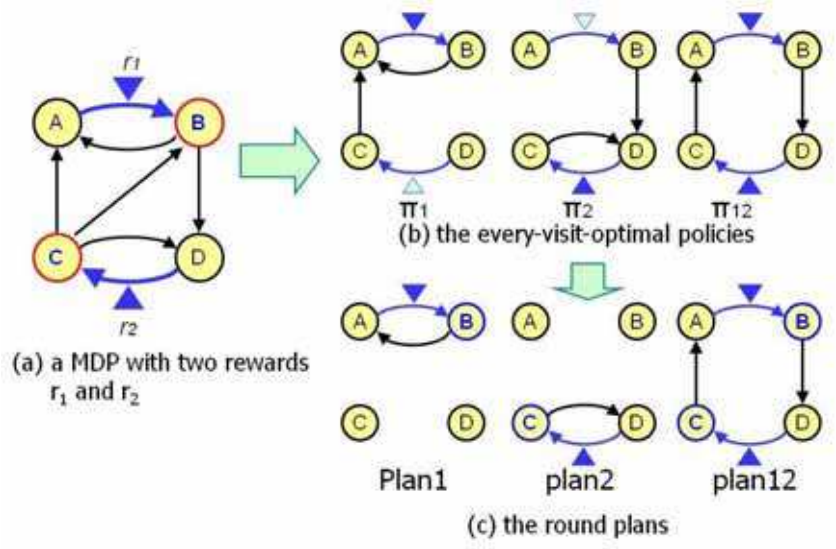


Fig. 4. Overview of preparing various round plans

3.2 Definition of various round plans by *every-visit-optimality*

Various round plans are defined by following steps.

1. Enumerate the all subsets of the reward function.
2. Search an *every-visit-optimal* policy for each subset of the reward function.
3. Collect all *every-visit-optimal* policies and convert them into round plans.

Fig. 5 illustrates the process for searching *various round plans*. When a reward function is identified as {*Rw1, Rw2*}, enumerated subsets of the function are {*Rw1*}, {*Rw2*}, {*Rw1, Rw2*} in step 1. Then an *every-visit-optimal* policy is decided for each subset of the reward function in step 2. At last, these *every-visit-optimal* policies are collected as *various round plans*. The number of plans in the *various round plans* is $2^r - 1$, where r is the number of rewards in the model.

3.3 Searching various policies

This section describes our *various policies* search method by interactive LC-Learning (Satoh & Yamaguchi, 2006). LC-Learning (Konda et al., 2002a; Konda et al., 2002b) is one of the average reward model-based reinforcement learning methods (Mahadevan, 1996). The features of LC-Learning are following;

1. Breadth search of an optimal policy started by each reward rule.
2. Calculating average reward by a reward acquisition cycle of each policy.

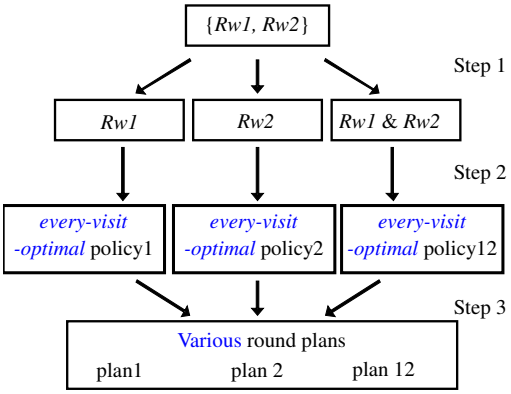
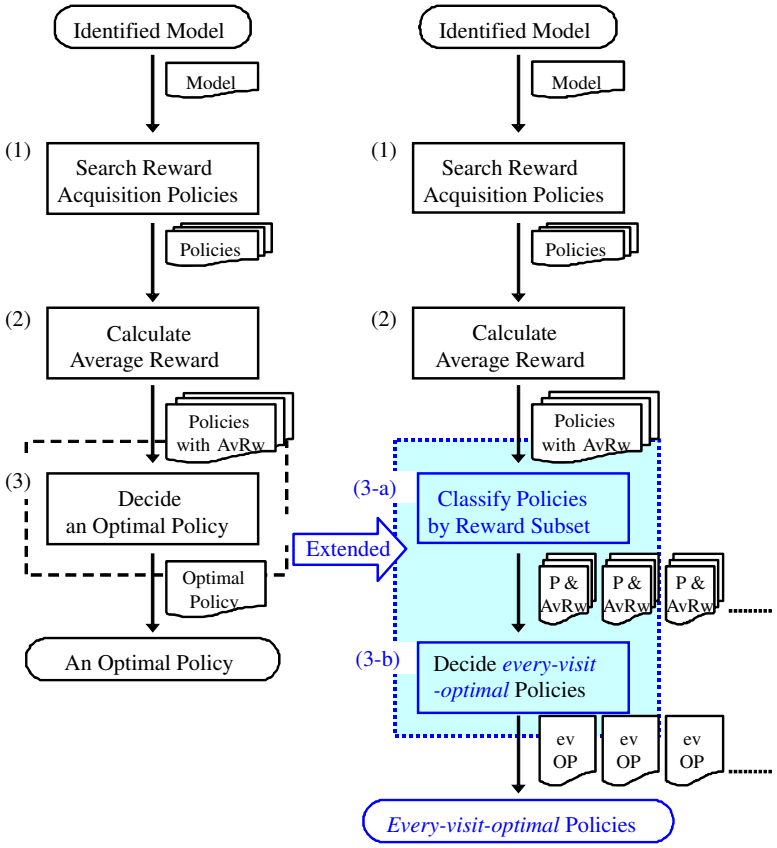


Fig. 5. Process for searching various round plans



(a) Standard LC- Learning (b) Interactive LC-Learning

Fig. 6. Algorithm for preparing *various policies*

Fig. 6 (a) shows standard LC-Learning algorithm. Previous LC-Learning decides an optimal-policy by following three steps.

1. Search policies that have a reward acquisition cycle.
2. Calculate average reward of searched policies.
3. Decide an optimal policy that has the maximum average reward.

Fig. 6 (b) shows algorithm for interactive LC-Learning. Major differences from standard LC-Learning are following;

1. Collecting various policies by *every-visit-optimality*
2. A stochastic version based on *occurring probability*
3. Adaptable for incremental reward addition

Next, we describe the three steps for interactive LC-Learning as shown in Fig.6 (b).

(1) Search reward acquisition policies

In this step, reward acquisition policies are searched by converting a MDP into the tree structures where reward acquisition rules are root rule. We show an illustrated example. Fig. 7 shows a MDP model with two rewards r_1 and r_2. It is converted into two tree structures. Fig. 8 shows two trees. First, a tree from reward r_1 as shown in Fig. 8 (a) is generated, then a tree from reward r_2 as shown in Fig. 8 (b) is generated. In a tree structure, a policy is a path from a root node to the state that is same state to the root node. In a path, an expanded state that is same state to the previous node is pruned since it means a local cycle. In Fig.8, node D and B are pruned states.

Fig. 9 shows all reward acquisition policies in Fig. 7. In a stochastic environment, several rules branch stochastically. In such case, a path from parent node of a stochastic rule to the state that is already extracted is part of a policy that contains the stochastic rule. The policy 12 in Fig.9 is an example of this.

(2) Calculate average reward

In this step, average reward of each policy is calculated by using *occurring probability* of each state of the policy. *Occurring probability* of a state is expected value of the number of transiting the state during the agent transit from the initial state to the initial state. Eq. (2) shows definition of the *occurring probability* of state s_j where initial state is s_i. *Occurring probability* of each state is calculated approximately by value iteration using eq. (2).

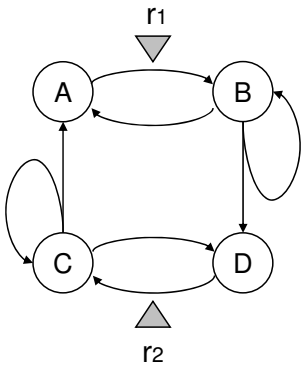
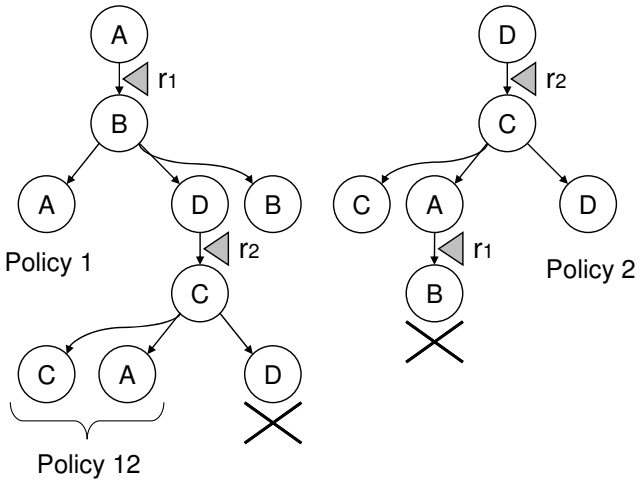


Fig. 7. An example of MDP model



(a) A tree from reward r_1 (b) A tree from reward r_2

Fig. 8. Searching reward acquisition policies

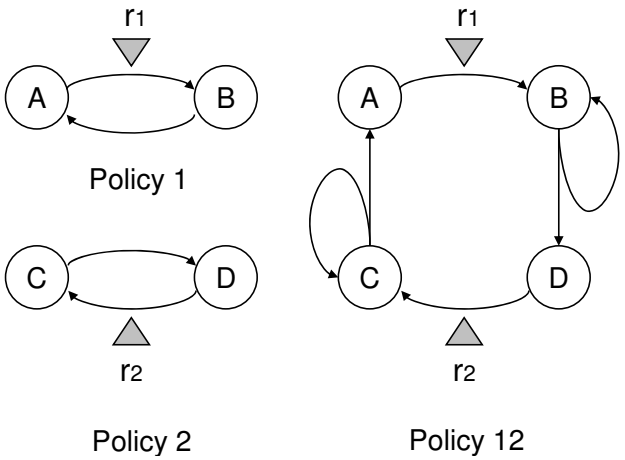


Fig. 9. Three kind of reward acquiring policies

$$P_o(s_j,s_i)=\begin{cases}1 & (j=i)\\ \sum_{s_k} P_o(s_k,s_i)P(s_j \mid s_k,a_k) & (j\neq i)\end{cases} \tag{2}$$

Where a_k is the action that is executed at state s_k.

$$g^{\pi}(s_i)=\frac{\sum_{s_j} P_o(s_j,s_i)R(s_j,a_j)}{\sum_{s_j} P_o(s_j,s_i)} \tag{3}$$

The average reward of policies is calculated by eq. (3) using *occurring probability* calculated by eq. (2).

(3'-1) Classify policies by reward subset

In this step, all policies searched by step 1 are classified by acquisition reward set.

(3'-2) Decide *every-visit-optimal* policies

In this step, an *every-visit-optimal* policy is decided for each group classified in step (3'-1). Each *every-visit-optimal* policy is a policy that had maximum average reward in the each group.

4. Plan recommendation

This section describes the plan recommendation system and the *coarse to fine recommendation* strategy (Yamaguchi & Nishimura, 2008). In this section, a *goal* is a reward to be acquired, and a *plan* means a cycle that acquires at least one reward in a policy.

4.1 Grouping various plans by the visited goals

After preparing various round plans in section 3.3, they are merged into group by the number of acquired reward. Fig. 10 shows grouping various plans by the number of visited goals. When three goals are input by a user, they are converted into three kinds of reward as *Rw1, Rw2*, and *Rw3*. Then, Group1 in Fig. 10 holds various plans acquiring only one reward among *Rw1, Rw2*, or *Rw3*. Group2 holds various plans acquiring two kinds of reward among *Rw1, Rw2*, or *Rw3*, and Group3 holds various plans acquiring *Rw1, Rw2*, and *Rw3*.

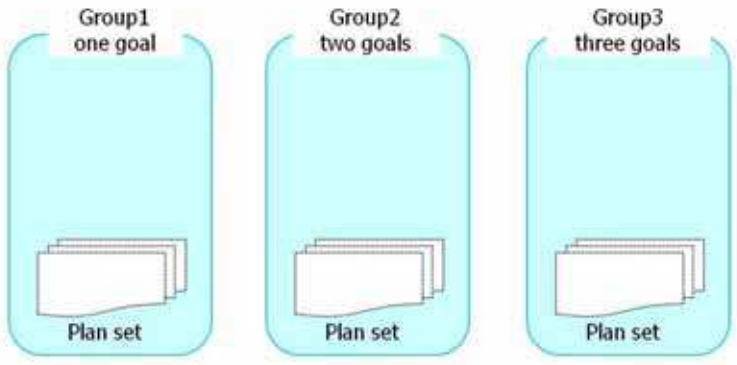


Fig. 10. Grouping various plans

4.2 Coarse to fine recommendation strategy

After grouping various plans by the number of visited goals, they are presented to the user sequentially for selecting the most preferable plan. We call the way to decide this order as recommendation strategy. In this paper, we propose *coarse to fine recommendation* strategy that consists of two steps, coarse recommendation step and fine recommendation step.

(1) Coarse recommendation step

For the user, the aim of this step is to select a preferable group. To support the user's decision, the system recommends a representative plan in each selected group to the user.

Fig. 11 shows a coarse recommendation sequence when a user changes his/ her preferable group as Group1, Group2, and Group3 sequentially. When the user selects a group, the system presents the representative plan in the group as the recommended plan.

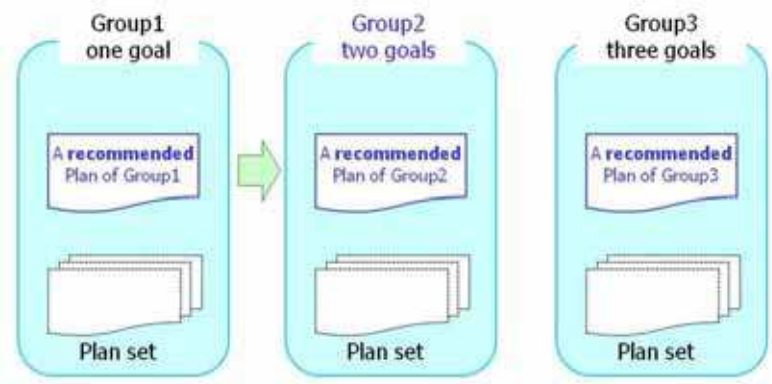


Fig. 11. Coarse recommendation

(2) Fine recommendation step

For the user, the aim of this step is to decide the most preferable plan in the selected group in previous step. To support the user's decision, the system recommends plans among his/ her selected group to the user. Fig. 12 shows a fine recommendation sequence after the user selects his/ her preferable group as Group2. In each group, plans are ordered according to the length of a plan.

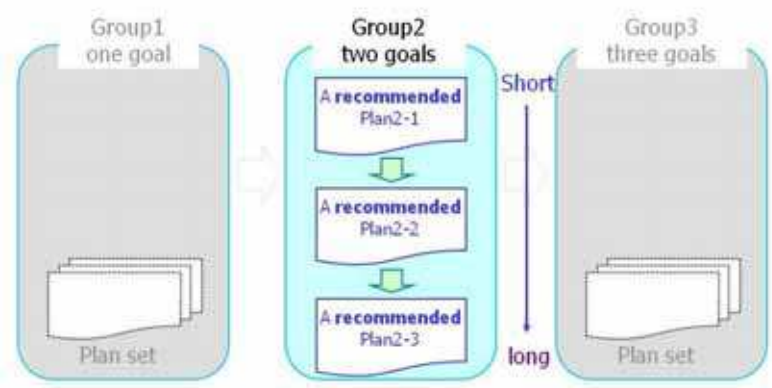


Fig. 12. Fine recommendation in the selected group

5. Experiment

We perform the experiment with twenty subjects from 19 to 21 years old to evaluate the effectiveness of our method.

5.1 The round-trip plan task

Fig. 13 shows the round-trip plan Recommendation task in Hokkaido. For a subject, this task is executed by following steps.

1. Each subject selects four cities to visit. Various round-trip plans are recommended.
2. The subject decides the most preferred round-trip plan among them. The task for a subject is to decide the most preferred round-trip plan after selecting four cities to visit among eighteen cities. The task for the system is to estimate the preferable round-trip plans to each user and to recommend them sequentially.

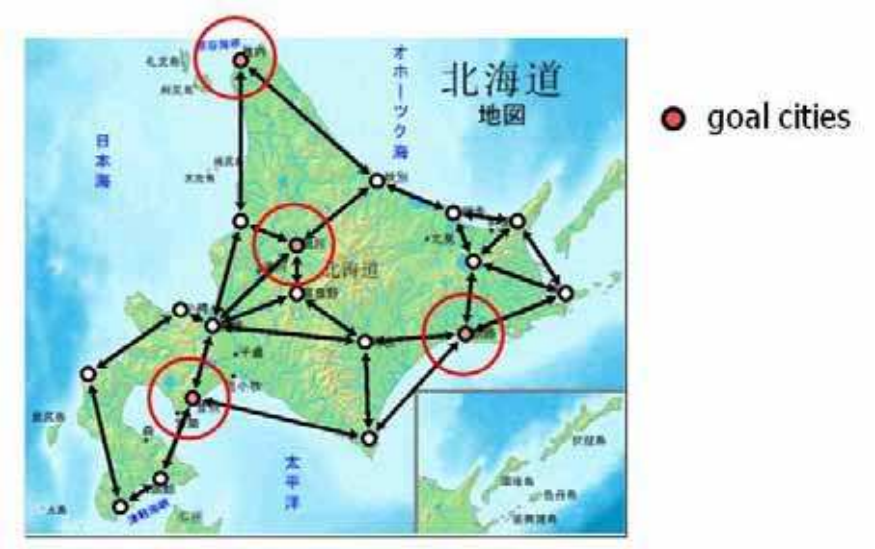


Fig. 13. The round-trip plan Recommendation task

5.2 Experimental results

Fig.14 shows the result of the most preferred plans of each twenty subjects. Horizontal axis is the number of visited cities (goals), and vertical axis is the number of subjects. The summary of the experimental result is as follows. First, the majority of subjects prefer each *every-visit* plan (visit all four cities) than the *optimal* plan. Second, majority prefers *shorter* plans, and minority prefers *longer* plans. Then we focus on these two points.

First point is the effectiveness of *every-visit* criterion. After selecting four cities, 15 (three-quarter) subjects preferred *every-visit* plans those visit selected four cities. In contrast, only 5 subjects preferred *optimal* plans with shorter length, yet these plans do not visit all four cities. This suggests that the *every-visit* criterion is preferable to the *optimality* criterion for human learners.

Second point is that the users' preferences are divided into two groups, *shorter* plans, or *longer* plans. We look more closely the preference for *every-visit* plans among 15 subjects. Among them, 10 (two-thirds) subjects preferred shorter (*every-visit-optimal*) plans, and 5 (third) subjects preferred longer (*every-visit-non-optimal*) plans. Among all 20 subjects, they indicate a similar tendency. Table 2 shows the summary of the experimental result. In table 2, a majority of subjects prefer shorter plans those are either *optimal* or *every-visit-optimal*, a

minority of subjects prefer longer plans those are *every-visit-non-optimal*. The reason why the end-users' preferences are divided into two groups will be discussed in the next section.

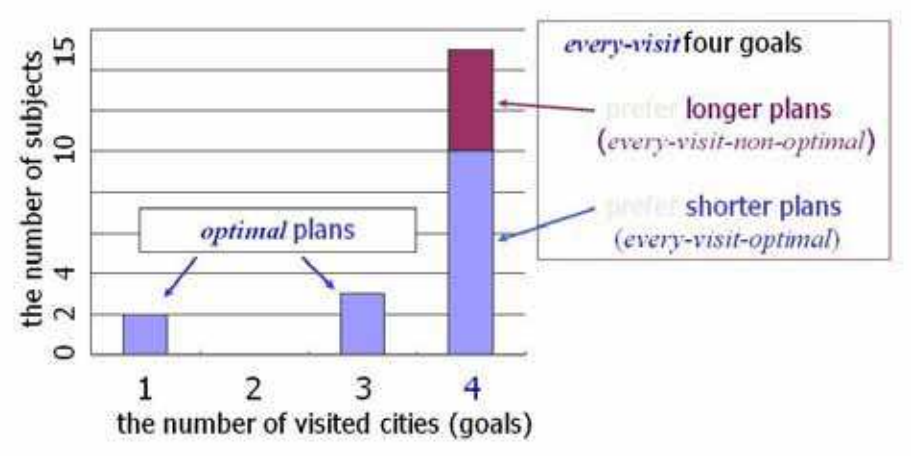


Fig. 14. The result of the most preferred plans

optimal plan	*every-visit* plan	
	every-visit-optimal	*every-visit-non-optimal*
short	shorter	long
5	10	5

Table 2. Summary of the most preferred plans

5.3 Discussions

(1) Why the end-users' preferences are divided?

We discuss the reason why the end-users' preferences are divided into two groups. Fig. 15 shows one of the *every-visit-optimal* plans those major subjects preferred. According to the results of the questionnaire survey, a majority of subjects selected an *every-visit-optimal* plan have less knowledge on Hokkaido (or no experience to visit Hokkaido).

In contrast, a minority of subjects selected *every-visit-non-optimal* plans those have additional cities to visit by the plan recommendation. Fig. 16 shows one of the *every-visit-non-optimal* plans the minority of subjects preferred. According to the results of the questionnaire survey, a majority of subjects selected an *every-visit-non-optimal* plan have much knowledge or interest on Hokkaido.

It suggests that the preference of a user depends on the degree of the user's background knowledge of the task. In other word, the change of the end-users' preference by the recommendation occurs whether they have the background knowledge of the task or not. Note that in our current plan recommendation system, no background knowledge on the recommended round-trip plan except Fig. 13 is presented to each subject. If any information about recommended plan is provided, we expect that the result on preference change of these two kinds of subjects will differ.

Fig. 15. One of the *every-visit-optimal* plans

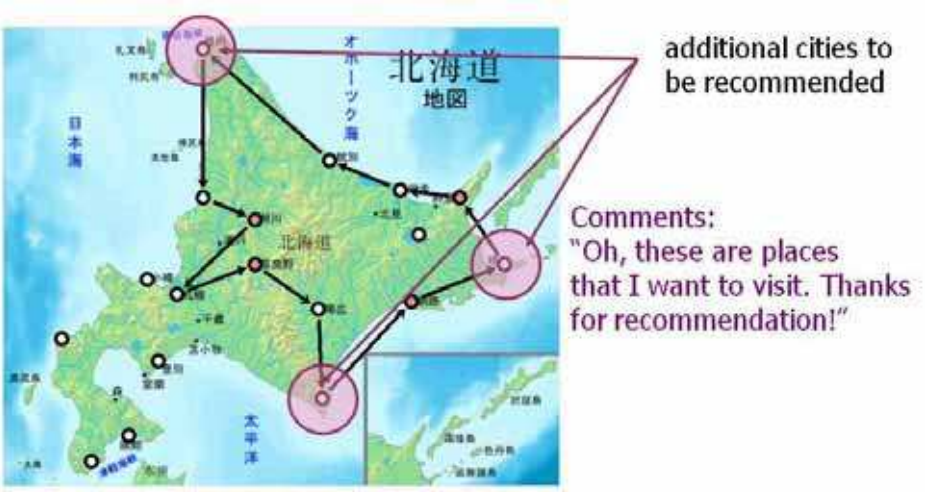


Fig. 16. One of the *every-visit-non-optimal* plans

(2) The search ability of interactive LC-learning

The computing time of the round-trip plan task in Fig. 13 including graphical output by interactive LC-learning is no more than one second or less per user input, since it is a deterministic MDP model. So we summarize the search ability of LC-Learning in a stochastic case (Satoh & Yamaguchi, 2006).

We compare two kinds of search abilities of LC-Learning to that of Modified-PIA (Puterman, 2006). First, the search cost of LC-Learning increases linearly when the number of rewards increases linearly. However, the search cost of Modified-PIA increases nonlinearly when the number of rewards increases linearly. Besides, Modified-PIA collects no *every-visit optimal* policy when the number of rewards is more than three. These suggest that our method is better than previous reinforcement learning methods for interactive

reinforcement learning in which many rewards are added incrementally. We go into the comparative experiments in detail in section 6.

(3) *Every-visit-optimality* in a non-deterministic environment

In a stochastic environment, *every-visit-optimality* is defined as *p-every-visit-optimality* where each reward is visited stochastically by not less than probability p ($0 < p =< 1$). It can be calculated by *occurring probability* of each rewarded rule described in section 3.3 (2). Note that *1-every-visit-optimality* is that each reward is visited deterministically even in a stochastic environment.

6. Evaluating the search ability of interactive LC-learning

To evaluate the effectiveness of interactive LC-learning in a stochastic domain, comparative experiments with preprocessed Modified-PIA are performed when the number of rewards increases. We compare the two kinds of search abilities as follows.

1. The search cost for *every-visit optimal* policies
2. The number of collected *every-visit-optimal* policies

6.1 Preprocess for Modified-PIA

Modified-PIA(Puterman, 2006) is one of the model-based reinforcement learning methods based on PIA modified for the average reward. However Modified-PIA is the method to search an optimal policy. So it is not valid to compare the search cost of the Modified-PIA and LC-Learning that searches *various policies*. To enable to search *various policies* by Modified-PIA, following preprocess is added. Fig. 17 shows the preprocessed Modified-PIA.

1. Enumerate the models those contain the subset of reward set of the original model.
2. Search an optimal policy for each subset of the reward function using Modified-PIA.
3. Collect optimal policies.

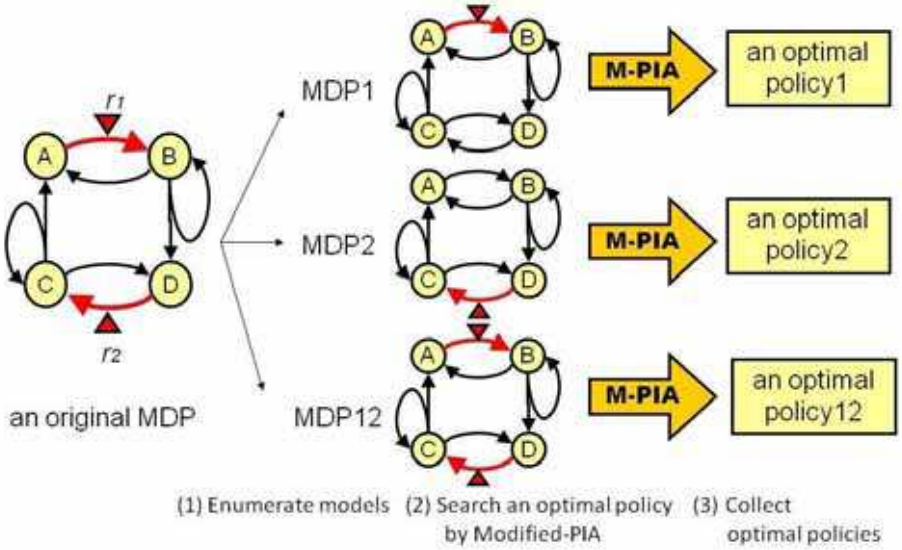


Fig. 17. The preprocessed Modified-PIA

6.2 Experimental setup

We use a hundred of MDP models those consist of randomly set state transition probability and reward function for experimental stochastic environment, in which the number of rewards is varied among 1 to 10, the number of states is 10 and the number of actions is 4. As the measure of the search cost, we used the iteration count in calculating the *occurring probability* of state for LC-Learning and we used the iteration count in calculating the value function for Modified-PIA.

6.3 The search cost for *every-visit-optimal* policies

To begin with, the search cost for *every-visit-optimal* policies is evaluated. Fig. 18 shows the comparative search cost when the number of rewards increases. The result indicates that the tendency of search cost of LC-Learning is linear and one of Modified-PIA is non-linear when the number of rewards increases.

Then we discuss the theoretical search cost. In Modified-PIA, MDP models those contain the subset of reward set of an original MDP are made and an optimal policy for each MDP is searched. So original Modified-PIA is performed 2^r-1 times where r is the number of rewards. After one reward is added, incremental search cost is following.

$$(2^{r+1}-1) - (2^r-1) = 2^r \tag{4}$$

Eq. (4) means that the search cost of Modified-PIA increases nonlinearly when the number of rewards increases. In contrast, in LC-Learning, the number of tree structure increase linearly when the number of rewards is increase. So it is considered that the search cost of LC-Learning increase linearly when the number of rewards increase.

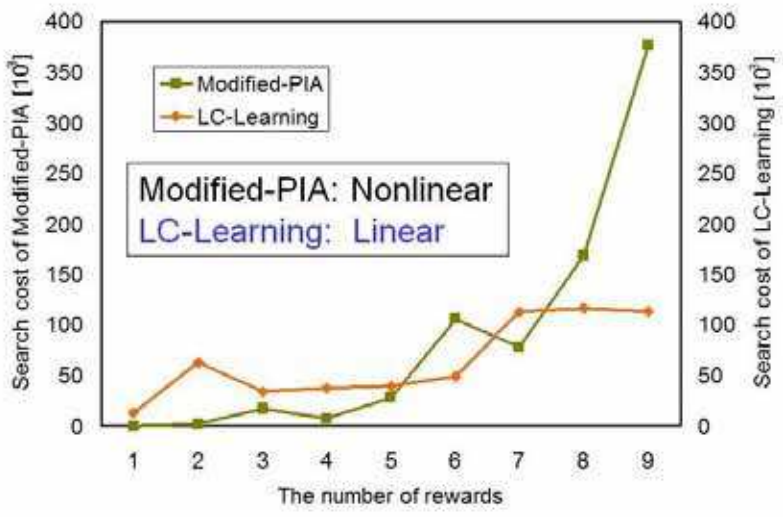


Fig. 18. Search cost when the number of rewards increases

6.4 The number of collected *every-visit-optimal* policies

To evaluate the effectiveness of interactive LC-learning, another search ability is compared with preprocessed Modified-PIA. Note that the experimental setup is same as the setup described in section 6.2. Fig. 19 shows the number of collected *every-visit-optimal* policies. Compared with LC-learning collecting all *every-visit-optimal* policies, the number of collected *every-visit-optimal* policies by preprocessed Modified-PIA is smaller than LC-learning.
Then, carefully analyzing the case of six rewards, Fig. 20 shows the rate of collected *every-visit-optimal* policies, that is percentage of LC-learning of preprocessed Modified-PIA. It

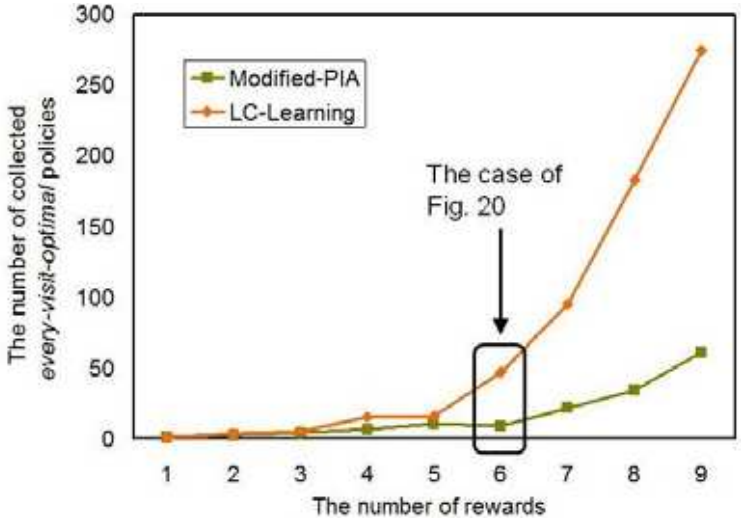


Fig. 19. The number of collected *every-visit-optimal* policies

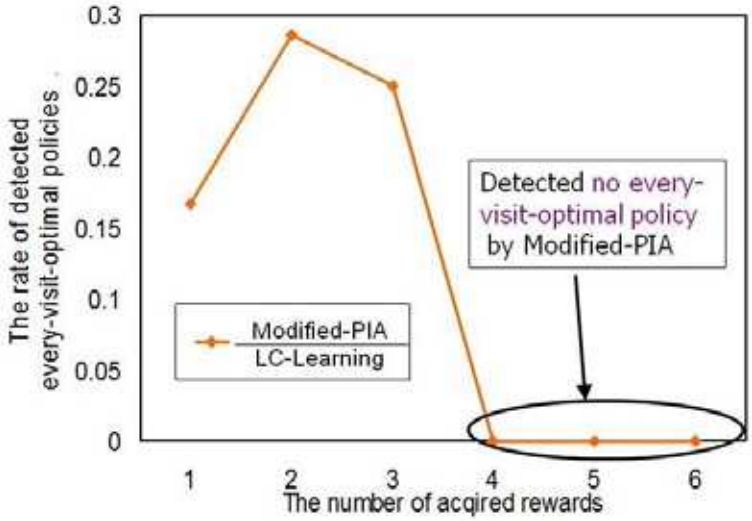


Fig. 20. The rate of collected *every-visit-optimal* policies

shows that preprocessed Modified-PIA collects no *every-visit-optimal* policy when the number of rewards is more than three.
Then we discuss the reason why the number of collected *every-visit-optimal* policies by preprocessed Modified-PIA is smaller than LC-learning. Since preprocessed Modified-PIA is based on the standard optimality, it searches an *optimal* policy in each MDP with the subset of reward set of the original model as shown in Fig.17. It means that preprocessed Modified-PIA finds an *every-visit-optimal* policy only if it is same as the *optimal* policy in each MDP model. As the number of rewards increases, the rate of *every-visit-optimal* policy that is same as the *optimal* policy decreases. In other words, the distinction between two criteria becomes larger according to the number of rewards increases.
Since most previous reinforcement learning methods including Modified-PIA are based on the standard optimality criterion, they only learn an *optimal* policy. Therefore, under *every-visit-optimality* criterion, our method is better than previous reinforcement learning methods for interactive reinforcement learning in which many rewards are added incrementally.

7. Related works on recommender systems

This section describes relations between our proposed solutions and current research issues on recommendation systems. The main feature of our recommendation system is interactive and adaptable recommendation for human users by interactive reinforcement learning. First, we describe two major problems on traditional recommenders. Second, interactive recommendation system called Conversational Recommender is summarized. At last, adaptive recommenders with learning ability are described.

7.1 Major problems on traditional recommenders

Main objective of recommender systems is to provide people with recommendations of items, they will appreciate based on their past preferences. Major approach is collaborative filtering, whether user-based or item-based (Sarwar et al., 2001) such as by Amazon.com. The common feature is that similarity is computed for users or items, based on their past preferences.
However, there are two major issues. First issue is the similar recommendations problem (Ziegler et al., 2005) in that many recommendations seem to be "similar" with respect to content. It is because of lack of novelty, serendipity (Murakami et al., 2007) and diversity of recommendations. Second issue is the preference change problem (Yamaguchi et al., 2009) that is inability to capture the user's preference change during the recommendation. It often occurs when the user is a beginner or a light user. For the first issue, there are two kinds of previous solutions. One is topic diversification (Ziegler et al., 2005) that is designed to balance and diversify personalized recommendation lists for user's full range of interests in specific topics. Another is visualizing the feature space (Hijikata et al., 2006) for editing a user's profile to search the different items on it by the user. However, these solutions do not directly considering a user's preference change. To solve this, this paper assumes a user's preference change as two-axes space, coarse and fine axes.

7.2 Interactive recommendation systems

Traditional recommenders are simple and non-interactive since they only decide which product to recommend to the user. So it is hard to support for recommending more complex

products such as travel products (Mahmood et al., 2009). Therefore, conversational recommender systems (Bridge et al., 2006) have been proposed to support more natural and interactive processes. Typical interactive recommendation is the following two strategies (Mahmood et al., 2008):

1. Ask the user in detail about her preferences.
2. Propose a set of products to the user and exploit the user feedback to refine future recommendations.

A major limitation of this approach is that there could be a large number of conversational but rigid strategies for a given recommendation task (Mahmood et al., 2008).

7.3 Adaptive recommenders with learning ability

There are several adaptive recommenders using reinforcement learning. Most of them observe a user's behavior such as products the user viewed or selected, then learn the user's decision processes or preferences. To improve the rigid strategies for conversational recommenders, learning personalized interaction strategies for conversational recommender systems has been proposed (Mahmood & Ricci, 2008; Mahmood & Ricci, 2009; Mahmood et al., 2009).

Major difference from them, the feature of our approach is adaptable recommendation for human users by passive recommendation strategy called *coarse to fine recommendation*. Adaptable recommendation means that during our recommendation, a user can select these two steps (coarse step or fine step) as his/ her likes before deciding the most preferable plan.

8. Conclusions

In this paper, we proposed a new method of interactive LC-learning for recommending preferable solutions of a user.

1. *Every-visit-optimality* as the optimality criterion of preference for most of end-users was assumed.
2. To cover the end-user's preference changes after the reward function is given by the end-user, interactive LC-learning prepared *various policies* by generating variations of the reward function under *every-visit-optimality*.
3. For guiding the end-user's current preference among *various policies*, *coarse to fine recommendation* strategy was proposed.

As the experimental results, first, the majority of subjects preferred each *every-visit* plan (visiting all goals) than the *optimal* plan. Second, majority preferred *shorter* plans, and minority prefers *longer* plans. We discussed the reason why the end-users' preferences are divided into two groups. Then, the search ability of interactive LC-learning in a stochastic domain was evaluated.

The future work is to assist a user for deciding the most preference plan to make his/ herself known the potential preference of the user. To realize this idea, we are evaluating passive recommendation by visualizing the *coarse to fine recommendation* space and the history of the recommendation of it (Yamaguchi et al., 2009).

9. References

Bridge, D., Go"ker, M. H., McGinty, L. & Smyth, B. (2005). Case-based recommender systems, *The Knowledge Engineering Review*, Volume 20, Issue 3 (September 2005), pp.315 - 320, Cambridge University Press, ISSN:0269-8889

Hijikata, Y., Iwahama, K., Takegawa, K., Nishida, S. (2006). Content-based Music Filtering System with Editable User Profile, *Proceedings of the 21st Annual ACM Symposium on Applied Computing (ACM SAC 2006)*, pp.1050-1057, Dijon, France, April, 2006

Kaplan, F.; Oudeyer, P-Y.; Kubinyi, E. & Miklosi, A. (2002). Robotic clicker training, *Robotics and Autonomous Systems*, Dillmann, R. et al. (Eds.), pp.197-206, ELSEVIER, ISBN0921-8890, Amsterdam

Konda, T.; Tensyo, S. & Yamaguchi, T. (2002). LC-Learning: Phased Method for Average Reward Reinforcement Learning - Analysis of Optimal Criteria -, *PRICAI2002: Trends in Artificial Intelligence*, Lecture notes in Artificial Intelligence 2417, Ishizuka, M. & Sattar, A. (Eds.), pp.198-207, Springer, ISBN 3-540-44038-0, Berlin

Konda, T.; Tensyo, S. & Yamaguchi, T. (2002). LC-Learning: Phased Method for Average Reward Reinforcement Learning - Preliminary Results -, *PRICAI2002: Trends in Artificial Intelligence*, Lecture notes in Artificial Intelligence 2417, Ishizuka, M. & Sattar, A. (Eds.), pp.208-217, Springer, ISBN 3-540-44038-0, Berlin

Konidaris, G. & Barto, A. (2006). Automonous Shaping: Knowledge Transfer in Reinforcement Learning, *Proceedings of the 23rd International Conference on Machine Learning*, pp.489-496, ISBN 1-59593-383-2, Pittsburgh, June, 2006

Mahadevan, S. (1996). Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results, *Machine Learning*, Vol.22, No.1-3, pp.159-195, Springer (Kluwer Academic Publishers), New York

Mahmood, T. & Ricci, F. (2008). Adapting the interaction state model in conversational recommender systems, *Proceedings of the 10th international conference on Electronic commerce*, ISBN 978-1-60558-075-3, Innsbruck, August, 2008, ACM, New York

Mahmood, T. & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies, *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pp.73-82, ISBN 978-1-60558-486-7, Torino, June to July, 2009, ACM, New York

Mahmood, T.; Ricci, F.; Venturini, A. & Hopken, W. (2008). Adaptive Recommender Systems for Travel Planning, *Information and Communication Technologies in Tourism 2008*: Proceedings of ENTER 2008 International Conference in Innsbruck, Hopken, W. & Gretzel, U. (Eds.), pp.1-11, Springer, ISBN 978-3-211-77279-9, New York

Mahmood, T.; Ricci, F. & Venturini, A. (2009). Improving Recommendation Effectiveness by Adapting the Dialogue Strategy in Online Travel Planning, *International Journal of Information Technology and Tourism*, Volume 11, No.4, pp.285-302, ISSN 1098-3058, Cognizant Communication Corporation, New York

Marthi, Bhaskara. (2007). Automatic shaping and decomposition of reward functions, *Proceedings of the 24th international conference on Machine learning*, pp.601-608, ISBN 978-1-59593-793-3, Corvallis, USA, June, 2007, ACM, New York

Murakami, T.; Mori, K. and Orihara, R. (2007). Metrics for Evaluating the Serendipity of Recommendation Lists, *New Frontiers in Artifical Intelligence*, Lecture notes in Artificial Intelligence 4914, Satoh, K. et al. (Eds.), pp.40-46, Springer, ISBN 978-3-540-78196-7, Berlin

Ng, Andrew Y.; Harada, Daishi; & Russell, Stuart J. (1999). Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping, *Proceedings of the 16th International Conference on Machine Learning*, pp.278-287, Bled, Slovenia, June, 1999

Preda, M.; Mirea, A.M.; Teodorescu-Mihai, C. & Preda, D. L. (2009). Adaptive Web Recommendation Systems, *Annals of University of Craiova, Math. Comp. Sci. Ser.*, Vol. 36 (2), pp.25-34

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, JOHN WILEY & SONS, INC, pp.385-388, ISBN 0471619779, New York

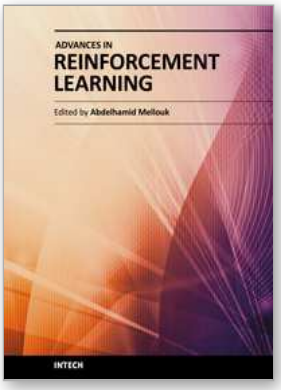
Sarwar, B.; Karypis, G.; Konstan, J. & Reidl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms, *Proceedings of the 10th International Conference on World Wide Web*, pp.285-295, Hong Kong, May, 2001, ACM, New York

Satoh, K. & Yamaguchi, T. (2006). Preparing various policies for interactive reinforcement learning, *Proceedings of the SICE-ICASE International Joint Conference 2006 (SICE-ICCAS 2006)*, pp.2440-2444, Busan, Korea, October, 2006

Yamaguchi & T., Nishimura, T. (2008): How to recommend preferable solutions of a user in interactive reinforcement learning?, *Proceedings of the International Conference on Instrumentation, Control and Information Technology (SICE2008)*, pp.2050-2055, , Chofu, Japan, August, 2008

Yamaguchi, T.; Nishimura, T. & Takadama, K. (2009). Awareness based filtering - Toward the Cooperative Learning in Human Agent Interaction -, *Proceedings of the ICROS-SICE International Joint Conference (ICCAS-SICE 2009)*, pp.1164-1167, Fukuoka, Japan, August, 2009

Ziegler, C.N.; McNee, S.M.; Konstan, J.A. & Lausen, G. (2005). Improving Recommendation Lists Through Topic Diversification, *Proceedings of the 14th international conference on World Wide Web(WWW2005)* , pp.22-32, Chiba, Japan, May, 2005, ACM, New York



Advances in Reinforcement Learning

Edited by Prof. Abdelhamid Mellouk

ISBN 978-953-307-369-9
Hard cover, 470 pages
Publisher InTech
Published online 14, January, 2011
Published in print edition January, 2011

Reinforcement Learning (RL) is a very dynamic area in terms of theory and application. This book brings together many different aspects of the current research on several fields associated to RL which has been growing rapidly, producing a wide variety of learning algorithms for different applications. Based on 24 Chapters, it covers a very broad variety of topics in RL and their application in autonomous systems. A set of chapters in this book provide a general overview of RL while other chapters focus mostly on the applications of RL paradigms: Game Theory, Multi-Agent Theory, Robotic, Networking Technologies, Vehicular Navigation, Medicine and Industrial Logistic.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tomohiro Yamaguchi, Takuma Nishimura and Kazuhiro Sato (2011). How to Recommend Preferable Solutions of a User in Interactive Reinforcement Learning?, Advances in Reinforcement Learning, Prof. Abdelhamid Mellouk (Ed.), ISBN: 978-953-307-369-9, InTech, Available from: http://www.intechopen.com/books/advances-in-reinforcement-learning/how-to-recommend-preferable-solutions-of-a-user-in-interactive-reinforcement-learning-

INTECH
open science | open minds

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