

The Method of Learning Personal Preference with Reinforcement Learning

Yong Hee Park, Won Seok Choi, Seong Gon Choi

Information & communication Engineering, Chungbuk National University, Chungcheongbuk-do, Korea

yh0360@chungbuk.ac.kr, wschoi@chungbuk.ac.kr, choisg@chungbuk.ac.kr

Abstract— We propose a method of learning personal preference using Reinforcement Learning (RL). Existing research on personal preference is a method using machine learning, and the learning results are simply related to input data. In order to better learn personal preference, evaluation of learning result is also important. Therefore, we use RL to calculate reward for learners' choice. For this, we design a model that calculates and evaluates reward. The results show learned policy and reward using proposed method in environment.

Keywords— Reinforcement Learning, Learning Preference, Recommendation System, Personal Preference, Q-learning

I. INTRODUCTION

There have been many studies on learning personal preferences for personal users. Learning personal preference is widely used in marketing and advertising [1]. Actually, the possibility has been suggested that consumer preferences can affect marketing [2], [3]. Recently, methods of providing recommended routes using personal preferences to tourism are being studied [4]-[6].

Learning methods for user preferences are mainly divided into two paradigms. First, there is a method of recommending items that the user liked in the past as a content-based recommendation system. In addition, the system of collaborative recommendation identifies users similar to those preferred by the user. Recently, demographic, knowledge-based, and hybrid systems are being studied. Most systems for learning personal preferences use the methods of probabilistic models, Bayesian or natural networks, decision trees, and association rules in machine learning [1], [7], [8].

Reinforcement Learning (RL) is a method of learning by interacting with agents and environments. This has been demonstrated as an effective robot learning method and has been successfully applied in many areas. RL's agent is a sequential decision task. It learns a policy that determines an action in a specific state. This can be model with 5 tuples of MDP. Q-learning is used as an RL method without a model using reward [9].

Most user preference learning selects machine learning methods of supervised learning and unsupervised learning. Our purpose is to propose a learning method that compares learned data with data by user and evaluates it. Therefore, there is a difference from the existing learning methods. For

This, we propose a new method of reinforcement learning-based preference learning using user decision data. The proposed method can learn decisions for each state and create policy for user patterns. To this end, we select a q-learning method. We also create probabilistic virtual data. And check the learned results by created a python-based agent.

Section 2 describes studies on existing personal preference learning and Q-learning. Section 3 describes the learning algorithm and its contents as a method for personal preference learning using RL. Section 4 describes creating virtual data, simulation setting, and simulation results. Section 5 describes the results.

II. RELATED WORKS

A. Learning Personal Preference

Several methods have been proposed to learn personal preferences. In particular, methods of learning using machine learning are being studied a lot. Learning user preferences using Support Vector Machine (SVM) is a well-known method [14]. User preference learning using deep learning is also being studied. In one study, an algorithm was proposed to learn and recommend user preferences using Restricted Boltzmann machine [15]. A method of learning a user's preference by extracting keywords from an image is also being studied. One study proposed an algorithm that analysis of relationship of preference within an image [16]. Personal preference learning of existing studies derives results that classify results or calculate weights. These results are dependent on the input data. Our goal is to compare and evaluate input data and learned data like a real person, which is different from existing studies. Therefore, we propose a method of continuously evaluating and compensating learned results.

B. Reinforcement Learning

Markov decision process (MDP) is process of making decisions sequentially and receiving rewards in a specific state [10]. It includes a 5-tuple consisting of $\{S, A, T, R, \gamma\}$ [11]. S means state. State literally means a state in the environment. When Go is taken as an example, the state can be expressed in a two-dimensional coordinate space and a state in which a stone is placed. A policy is a rule that determines an action in

a characteristic state. A is Action and it means the work to be performed in a particular state. T is the probability of state transition and it means to the probability of performing a specific action in a specific state. R is Reward and it means to a compensation value when a specific action is performed in a specific state. γ is a discount factor. This means the discount rate of the next state when calculating the sum of the rewards. Environment is the object of learning. The agent selects an action using policy and performs learning. Episode is a sequence that means the beginning and end of learning. Since the initial policy has not been learned, the agent randomly selects an action [12]. As number of episodes increases, policy updates and it selects action that receives best reward. There is a problem of exploring only the initially learned policy. To solve this problem, there is a method of randomly selecting an action by a specific probability, such as ϵ -greedy.

Q-learning is one of the RL techniques for learning without a model [13]. It can be used to find the optimal policy in a given MDP. Q-value is a function that helps predict and calculate the expected reward for a certain action in a specific state. Q-learning learns this q-value to derive the optimal policy. Q-value consists of a pair of state and action, and the update formula is shown in Formula 1 below.

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a (Q(s_{t+1}, a) - Q(s_t, a_t))) \quad (1)$$

III. METHOD OF LEARNING PERSONAL PREFERENCE

We propose an RL-based learning method for personal preferences. The algorithm used q-learning.

The overall learning structure is as follows.

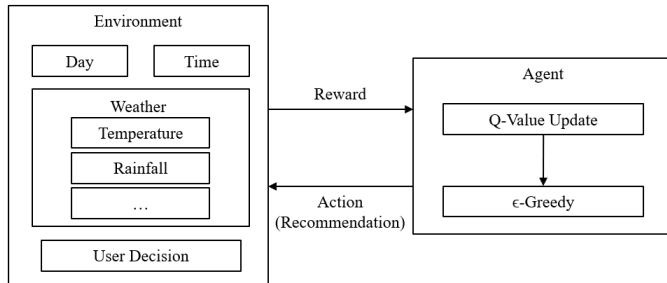


Figure 1. Architecture of Learning personal Preference

The state includes day, time, decision, weather information in the current time zone. Weather information includes data such as temperature and humidity. The end of the episode is the end of the day. Action was defined as an item that user could select. The ϵ -greedy method was selected as a random policy to select Action. Reward is set to increase when selected action from policy matches the actual decision.

The algorithm for personal preference learning is as follows.

Algorithm 1 Learning Personal Preference with RL

```

1: Initialize day, hour = 0
2: Initialize environment from data
3: Initialize s, a from environment
4: Initialize Q(s, a) from agent
5: Initialize episode_count, day_count
6: for all n ∈ episode_count
7:   while done != true
8:     Update a from ε-greedy
9:     Update s', r, done
        from Environment Step Algorithm
10:    Update Q(s, a) from Q-value Update
11:  end while
12: end for

```

First, the algorithm initializes day, hour to zero. Environment is a workspace for operating algorithms and it is initialized using collected data. s and a mean state and action, and they are initialized using environment based on imported data. $Q(s, a)$ is the Q-value value and initializes the matrix in the agent. $episode_count$ is the number of times an episode is performed. day_count is the number of times to learn for each day. Each episode confirms termination using the *done* parameter. Action follows the ϵ -greedy policy. s' is the next state and it includes next day, hour, decision, and weather information. r means reward. Parameters $s', r, done$ can be updated using the Environment Step Algorithm. Q-value value can be updated through the Agent Update Algorithm. The update for the Q-value uses Formula 1. By performing the algorithm as many time as given, the learned q-value can be obtained.

The Environment Step is an algorithm that updates the environment every time a step is performed, and the process is shown in algorithm 2.

Algorithm 2 Environment Step

```

1: Initialize r = 0
2: Initialize done = false
3: If a == '0' then
4:   If a == st then
5:     Update r ← r1
6:   end if
7: else then
8:   If a == st then
9:     Update r ← r2
10:  end if
11: end if
12: Update hour ← hour + 1
13: If hour > 23 then
14:   Update done ← true
15:   Update hour ← 0
16:   Update day_number ← day_number + 1
17: end if
18: If day_number > day_count then
19:   Update day ← day + 1
20: Update s' from environment, day, hour

```

21: **return** $s', reward, done$

First, reward is initialized to 0, and done to false. In addition, action selected by policy and selected decision at current time zone are compared, and if this is correct, the agent increases reward. In this case, the agent checks whether a user has made a decision and gives a difference in reward. 1 step means 1 hour. At 24 steps a day, episode ends and the *done* is updated to true. 1 day is learned by the set number of *day_count*. Environment returns the next state using day, step in the collected data.

IV. SIMULATION

This section is divided into three categories: creating virtual data, simulation setting, and simulation result.

A. Creating Virtual Data

Data on the user's selection includes information on day, hour, decision, and weather. Day is 365 days from the 1st, and hour is from 0:00 to 23:00 on the day. The Decision is the user's choice for each time. The Decision consists of nine types of selection, including 'A' to 'H' and 'X'. 'X' means non-decision. In the simulation of this algorithm, weather includes temperature. Temperature information uses the Meteorological Administration's ground time data inquiry service API in a public data portal in Korea [18]. We need decision data to learn the user's preferences. Therefore, we created data probabilistically. The created data is content that the user selects a specific option and it means menu selection for breakfast, lunch, dinner, and night food. The following table 1 is the time range for each meal time.

TABLE 1. RANGE OF MEAL TIME

Meal time	Range
Breakfast	5 ~ 7
Lunch	11 ~ 13
Dinner	18 ~ 22
Night food	23 ~ 24

There are four meal times: breakfast, lunch, dinner, and night food. personal preferences not only select menus, but also have meal times. Therefore, we set the range for each time zone of meal times. For example, some people have breakfast just before work hours, and that time is most likely at 6 o'clock. In addition, meal times can change depending on the temperature.

The time probability of menu determination can be seen in Table 2 below. 'ta' is the temperature in centigrade units.

TABLE 2. PROBABILITY OF MEAL TIME

t	0	1	2	3	4	5	6	7
ta<20	0.7					0.2	0.5	0.3
ta>20	0.7					0.6	0.1	0.3

t	8	9	10	11	12	13	14	15
ta<20				0.4	0.5	0.1		
ta>20				0.4	0.5	0.1		
t	16	17	18	19	20	21	22	23
ta<20			0.2	0.3	0.3	0.1	0.1	0.3
ta>20			0	0.2	0.3	0.5	0	0.3

There is also a preference for individual meals. Therefore, we express how often individual person eats at a specific time zone by giving probability for each meal time zone.

The probability of menu selection can be seen in Table 3 below.

TABLE 3. PROBABILITY OF MENU DECISION

Meal time	Menu Decision
Breakfast	ta<20 A= 0.1, B=0.6, C=0.3
	ta>20 A= 0.1, B= 0.6, C= 0.3
Lunch	ta<20 C=0.1, D=0.5, E=0.4
	ta>20 C=0.6, D=0, E=0.4
Dinner	ta<20 A=0.2, E=0.4, F=0.4
	ta>20 A=0.2, E=0.4, F=0.4
Night food	ta<20 F=0.5, G=0.5
	ta>20 F=0.5, G=0.5

Each individual has a preferred menu for each meal time. For example, someone can often go to a restaurant near their workplace during lunchtime. Weather can also influence individual menu choices.

B. Simulation Setting

The setting for the observable state is as shown in Table 4.

TABLE 4. OBSERVATION STATE

Observation	Range	Description
Day	1 ~ 365	a year range, per day
Step	0 ~ 23	a day range, per hour
Decision	0 ~ 7	Index of [X, A, B, C, D, E, F, G] (X = Non-decision)
Temperature	-40 ~ 40	Per 1 centigrade

The date to learn was set to a total of one year. Because individual meal choices can also be affected by the season. Since the step has a unit of 1 hour, the range is from 0 to 23. There are a total of seven menus, from "A" to "G." There are eight menu choices, including 'X', which means not to make a choice. The temperature ranges from -40 to 40, and the unit is 1 celsius.

Action selection is made using index, which is shown in Table 5.

TABLE 5. ACTION VECTOR

Action	X	A	B	C	D	E	F	G
Index	0	1	2	3	4	5	6	7

In the simulation, action selection is performed using a vector consisting of indexes from 0 to 7.

The parameter of the environment is set as shown in the following table 6.

TABLE 6. PARAMETER OF ENVIRONMENT

Parameter	Value
<i>episode_count</i>	1,825,000 (365 * 5,000)
<i>day_count</i>	5000

day_count is set as 5,000 times. In prior learning, the results were stable when about 5000 learning was conducted. Since the learning date is 365 days in total, *episode_count* is 1,825,000.

The parameter of the agent is set as shown in Table 7 below.

TABLE 7. PARAMETER OF AGENT

Parameter	Value
Epsilon (ϵ)	0.1
Learning rate (α)	0.01
Gamma (γ)	0.1
Reward 1 (r_1)	5
Reward 2 (r_2)	100

The proposed learning model does not learn well if it follows only the action selected in the policy. Therefore, A has a slightly higher value of 0.1. In this simulation, the number of episodes is about 1.8 million, which is slightly higher. Therefore, in the case of policy update, A is set to 0.01 to apply a lot of previously learned results. Since menu selection in the current time zone has little effect on menu selection in the next time zone, A is set to a small value of 0.1. Reward 1 is the reward when 'X' is matched. Reward 2 is a reward when matching menu choices. We make a difference in this value because matching menu choices is more important.

C. Simulation Result

Reward for each episode is shown in Figure 2 below.

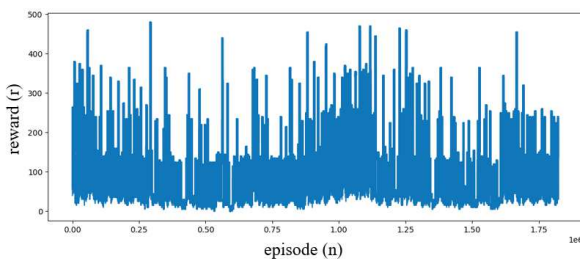


Figure 2. Reward for each Episode

As shown in Figure 2, each menu is created probabilistically, so it does not always fit the policy. The goal of the agent is to learn the policy of maximizing the reward for each episode. The maximum value of created policy is shown in Figure 3 below.

```
j = 0
[7, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 5, 5, 0, 0]
j = 1
[7, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 3, 5, 0, 0, 0, 0, 0, 5, 5, 0, 0]
```

Figure 3. Maximum value of Learned Policy

When $j=0$, the temperature is less than 20 degrees, and when $j=1$, the temperature is more than 20 degrees. Since the policy aims to maximize rewards, it follows the probability of virtually created data. In Figure 3, it can be seen that the policy has been learned approximately to the probability of menu selection. For example, if $j=0$, menu D is recommended at lunch, and if $j=1$, C and E are recommended. This is similar to the probability of created virtual data. However, it can be seen that breakfast recommends 6 o'clock when $j=0$ and 7 o'clock when $j=1$. For this part, there is an error because the probability of virtual data is the highest at 5 o'clock when $j=1$. In addition, in the case of dinner, the algorithm recommends in three time zones when $j=0$ because the probabilities of the time zones are similar in virtual data.

V. CONCLUSIONS AND FURTHER STUDY

We propose a method of learning personal preference using RL. To this end, a learning model was designed and the results were confirmed in simulation. From the simulation results, it can be seen that the selection by policy is similar to probability of virtual data. However, some results include errors. Because input data is based on probabilistic data, and the learning model is heuristic method. It can be seen that errors are also based on the probability of virtual data. In this paper, only the basic process for learning personal preference was proposed. Therefore, further research is needed such as optimal learning methods and accuracy improvement. A method of learning areas that are difficult to express with probability by utilizing the characteristics of RL (Heuristic learning methods) is also a subject of study.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1A2C1006167).

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2020R1A6A1A12047945)

* Corresponding Author: S.G. Choi(choisg@cbnu.ac.kr)

REFERENCES

- [1] G. Semeraro, F. Narducci, Preference learning in recommender systems, ECML PKDD 2009, 2009.
- [2] B. J. Phillips, E. F. McQuarrie and W. G. Griffin, "How visual brand identity shapes consumer response", *Psychology & Marketing*, vol. 31, no. 3, pp. 225-236, 2014.
- [3] D. Lee, K. Hosanagar and H. S. Nair, "Advertising content and consumer engagement on social media: evidence from Facebook", *Management Science*, 2018.
- [4] K. A. Farni, F. Nafis, B. Agoutane, A. Yahyaoui, J. Riffi, A. Sabri, "Hybrid recommender system for tourism based on big data and AI: A conceptual framework", *Big Data Mining and Analytics*, vol. 4, Issue. 1, pp. 47-55, 2021
- [5] W. Yin, Y. Su, J. Zhao, "Personalized Tourism Route Recommendation System Based on Dynamic Clustering of User Groups", 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), 2021
- [6] K. Li, C. Qu, "Design and Implementation of Tourism Route Recommendation System Based on LBS", 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2021.
- [7] L. Beilin, S. Yi, "Survey of personalized recommendation based on society networks analysis", 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, 2013.
- [8] M. Li, L. Wang, "A Survey on Personalized News Recommendation Technology", *IEEE Access*, vol. 7, pp. 145861 - 145879, 2019.
- [9] G. Li, R. Gomez, K. Nakamura, B. He, "Human-Centered Reinforcement Learning: A Survey", *IEEE Transactions on Human-Machine Systems*, vol.49, issue. 4, pp. 337 - 349, 2019.
- [10] R. Bellman, "A markovian decision process", *Indiana Univ. Math. J.*, vol. 6, pp. 679-684, 1957.
- [11] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- [12] W. Qiang, Z. Zhongli, "Reinforcement learning model, algorithms and its application", 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), 2011.
- [13] C. Watkins and P. Dayan, "Q-learning", *Machine learning*, vol. 8, no. 3-4, pp. 279-292, 1992.
- [14] H. Wang, H. Zhang, "Movie genre preference prediction using machine learning for customer-based information", 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018.
- [15] J. Ge, C. Yang, Y. Fang, "Recommendation algorithm based on users' interest preferences and Restricted Boltzmann machine", 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), 2019.
- [16] J. Chen, C. Wang, J. Wang, X. Ying, X. Wang, "Learning the Personalized Intransitive Preferences of Images", *IEEE Transactions on Image Processing*, vol. 26, issue. 9, pp. 4139-4153, 2017.
- [17] Meteorological Administration ground time data inquiry service, Public data portal, <https://www.data.go.kr/data/15057210/openapi.do>



Won Seok Choi received B.S. and Ph.D. degree in the College of Electrical and Computer Engineering, Chungbuk National University, Korea in 2008 and 2015 respectively. He is currently researcher in Research Institute of Computer and Information Communication(RICIC), Chungbuk National University. His research interests include Vehicle network, AI, Energy saving network, SDN, NFV and NGN.



Seong Gon Choi received B.S. degree in Electronics Engineering from Kyeongbuk National University in 1990, and M.S. and Ph.D. degree from Information Communications University, Korea in 1999 and 2004, respectively. He is currently an associate professor in College of Electrical & Computer Engineering, Chungbuk National University. His research interests include Autonomous Vehicle, smart grid, IoT, mobile communication, high-speed network architecture and protocol.



Yong Hee Park received B.S. degree in the College of Information & Communication Engineering, Chungbuk National University, Korea in 2019. He is currently a M.S. candidate in School of Information & Communication Engineering, Chungbuk National University. His research interests include Autonomous Vehicle, AI, Smart Grid.