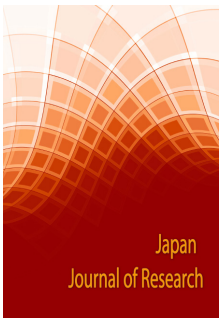
Japan Journal of Research

Original Article



Prediction for Occurrence of Character Identification Difficulty during Web Browsing by Apprenticeship Learning and Gaze Data

Matsushita Yutaka[1] and Aoki Shusei[2]

[1]Professor, Department of Media Informatics, Kanazawa Institute of Technology, Ishikawa, Japan
[2]Graduate Student, Department of Media Informatics, Kanazawa Institute of Technology, Ishikawa, Japan

Correspondence
Yutaka Matsushita
Department of Media Informatics, Kanazawa Institute of Technology 3-1 Yatsukaho, Hakusan, Ishikawa 924-0838, Japan

Abstract

Aiming to build a web-browsing assistance system that enlarges characters when users have difficulty identifying them and fix their gaze at them, this study proposes a method for predicting the occurrence of character identification difficulty during web browsing based on eye-tracking data. A criterion is needed to determine when characters should be enlarged. However, establishing such a criterion is highly challenging because the fixation duration associated with identification difficulty is not necessarily longer than that of resulting from other factors. Therefore, the criterion is calculated by introducing the user's saccadic velocity and amplitude as external parameters and classifying their combinations. In this process, the study employs apprenticeship learning (a form of inverse reinforcement learning), thereby eliminating the need to evaluate rewards. Moreover, a new algorithm, called "Recurrent Two-Step Linear Programming Apprenticeship Learning (RTLPAL)," is developed based on linear programming to reduce processing time and improve accuracy. This algorithm is characterized by the recursive repetition of two processes: deriving the optimal policy for the apprentice and generating an improved expert (reward). When applied to eye-tracking data from 10 participants during web browsing, RTLPAL improves accuracy metrics and reduces processing time to approximately 1/20th of that by SARSA.

- Received Date: 07 Aug 2026
- Accepted Date: 18 Aug 2026
- Publication Date: 21 Aug 2026

Copyright
2026 Authors. This is an open- access article distributed under the terms of the Creative Commons Attribution 4.0 International license.

Introduction

For users who have difficulty identifying characters, a web browsing assistance system that enlarges the character by fixing their gaze on it would be useful (Figure 1). Aiming to build such a system, this study proposes a method for predicting the occurrence of character identification difficulty based on gaze data. This browsing system must minimize two types of errors: enlarging characters when the user is not experiencing identification difficulty (**Error 1**), and failing to enlarge characters when the user is experiencing identification difficulty (**Error 2**). Although a criterion for determining whether a character should be enlarged is essential, establishing such a criterion is challenging because the fixation time owing to identification difficulties is not necessarily longer than that resulting from other factors. Figure 2 illustrates the fixation time attributable to identification difficulty and other factors. As shown, setting the criterion at Level 1 results in two Error 1 occurrences, whereas setting the criterion at Level 2 results in four Error 2 occurrences.

There is no supervisor signal (implying that neural networks are unsuitable tools). Therefore, reinforcement learning [1] is effective for this task because, through trial and error, an agent (learner) can learn the eye movements that occur when character identification becomes difficult. To enable rapid prediction and address the difficulty in setting the criterion, the following measures [2] were taken:

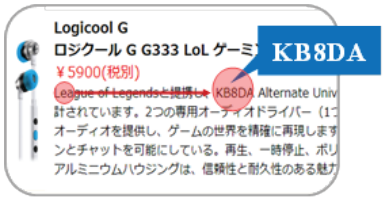


Figure 1. *Web browsing assistance system*

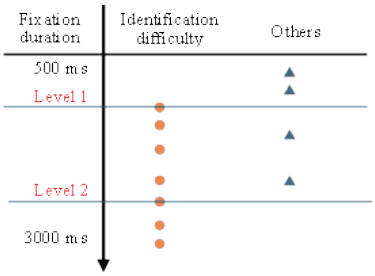


Figure 2. *Fixation time attributable to factors*

Citation: Matsushita Y, Aoki S , Prediction for Occurrence of Character Identification Difficulty during Web Browsing by Apprenticeship Learning and Gaze Data. Japan J Res. 2026;7(6):193.

- The fixation duration at each fixation point was increased sequentially, and the agent determined at each increment whether character identification difficulty occurred.
- The user's saccadic velocity and amplitude were introduced as external parameters, and a criterion for character enlargement was provided based on a combination of the categories of these two parameters. This treatment is consistent with the psychological [3, 4] and ergonomic [5] studies.

In reinforcement learning, it is necessary to evaluate rewards assigned to the agent when an enlargement decision is successful, so the evaluation process becomes more complex as the number of categories for the two parameters increases. Furthermore, the premise of setting the criterion (threshold) imposes the **constraint** on the choice of actions: "At every fixation point where the fixation time exceeds the threshold, the system must enlarge the character." This does not involve a precise prediction for each fixation point; rather, it requires a uniform threshold across all fixation points to determine whether to enlarge the character, even though the predictions (correct action choices) of certain fixation points are sacrificed. This markedly increases the number of training iterations. Notably, apprenticeship learning (which is a form of inverse reinforcement learning) [6, 7, 8] does not require reward evaluation, and the linear programming-based algorithm developed by Sayed et al. [9] significantly reduces the processing time. The algorithm is built on the assumption that an apprentice makes the best choice when following the worst expert (reward). Therefore, it has been reported that prediction accuracy decreases when the expert's selection deviates significantly from the optimal reward. In this study, we develop a useful prediction method that improves upon Sayed's algorithm. The specific research topics are as follows:

- Incorporate the above constraint into the objective function of a linear programming problem.
- Develop an algorithm that generates an ideal expert from many experts and infers an apprentice policy superior to that of the ideal expert.

Preliminary

Basic Concepts

In reinforcement learning, an agent chooses an action in each state, reaches the next state, and assesses whether the action contributes to a future expected cumulative reward. Let S and A be the sets of all states and actions, respectively. Let S_t and A_t denote the sets of states and actions at each time step t, which are generally random variables. The reward is expressed as function $r(s, a)$. The agent selects an appropriate action at each step t based on policy $\pi(a \mid s) = \Pr(A_t = a \mid S_t = s)$, which is expressed as a conditional probability. State transitions are expressed as conditional probability $\theta(s' \mid a, s) = \Pr(S_t = s' \mid A_t = a, S_t = s)$. Finally, the future reward is expressed as an expected value (expected return) with respect to π and θ as follows:

$$V^{\pi} = \mathbb{E}(\lim_{T\to\infty} \sum_{t=0}^{T} \gamma^t r(s_t, a_t) \mid \pi, \theta), \qquad (1)$$

where T is the length of time steps and γ is a discounting rate. The goal of reinforcement learning is to find a policy π that maximizes V^{π} through repeated learning.

Environment Variables

A state is defined as the elapsed time at which the fixation duration evolves step-by-step. If the time increment is Δd, the state (fixation time) s_t at step t (= 0, ..., T) is expressed by

$$s_t = F_{\min} + \Delta d \cdot t, \qquad (2)$$

where $F_{\min}$ = 500 ms (the minimum fixation time). Since the state is uniquely determined at step t by (2), $S_t = \{s_t\}$, and hence $\Pr(S_t = s_t) = 1$. This property is specific to the learning task that we consider. An action is a character enlargement ($a = 1$) or non-enlargement ($a = 0$), i.e., $A = \{0, 1\}$. It is assumed that s_t evolves up to 3000 ms regardless of the choice of action. In both cases, $a = 0$ and $a = 1$, and the state transition probability is written in the form

$$\theta(s_{t+1} \mid a, s_t) = 1.$$

To address the problem of setting thresholds for character enlargement mentioned in Introduction, we divide the saccade velocity and amplitude into N categories and determine a character enlargement criterion, $\mathrm{Fix}_{(i,j)}$, for each combination of categories (i, j) $(1 \le i, j \le N)$. The constraint in Introduction is described as follows:

$$s_t \ge \mathrm{Fix}_{(i,j)} \Rightarrow a = 1;\ s_t < \mathrm{Fix}_{(i,j)} \Rightarrow a = 0. \qquad (3)$$

However, as the number of categories increases, the number of rewards to evaluate also increases, thereby increasing the complexity of the evaluation process. Apprenticeship learning could solve this problem.

A learner decides character enlargement based on policy $\pi_{(i,j)}(a \mid s)$. Since $\pi_{(i,j)}(a \mid s)$ is a probability, the enlargement is determined by

$$\pi_{(i,j)}(1 \mid s_t) > \pi_{(i,j)}(0 \mid s_t) \Rightarrow a = 1.$$

Here, the policy is decided in a way that maximizes the expected return. The constraint of (3) can be formulated using this policy. The constraint implies that the change from $a = 0$ to $a = 1$ occurs only once during the evolution of s_t. Therefore, by recalling that $\pi_{(i,j)}(a \mid s)$ is a probability, we only need to minimize the formula

$$\sum_t \left| \pi_{(i,j)}(a \mid s_{t+1}) - \pi_{(i,j)}(a \mid s_t) \right| - 1. \qquad (4)$$

Furthermore, based on the previous results [2], the reward and penalty are evaluated as follows:

- To reduce the frequency of Error 1, a small reward is provided if the enlargement is correct and only a small penalty is imposed if it is incorrect.
- To reduce the frequency of Error 2, a serious penalty is imposed if enlargement is not performed during a long fixation duration, and if the choice is incorrect.

Prediction model

Expert and Apprentice

For simplicity, the index (i, j) is removed from the notation of the policy and expected return. When assessing the expected return, if we use (1), then since we set $\gamma = 0.9$, the term $\gamma^t r(s_t, a_t)$ decreases as step t progresses, thus we adopt the time average of expected returns as follows:

$$V(\pi) = \frac{1}{T} \sum_{t=0}^{T-1} V^{\pi}(s_t), \text{ where}$$

$$V^{\pi}(s_t) = \sum_{i=t}^{T-1} \sum_{a} \gamma^{i-t} r(s_i, a) \pi(a \mid s_i). \qquad (5)$$

Let $V(\pi^E)$ and $V(\pi^A)$ denote the average of expected returns

for the expert and apprentice determined by (5). The goal of apprenticeship learning [6] is to find an apprentice policy such that

$$V(\pi^A) \geq V(\pi^E). \qquad (6)$$

Expert's Policy and Average Expected Return

Let N_d be the total number of fixation points emerging from difficulty in character identification. For each fixation point i and each state s_t, $a_i(s_t) = 1$ if $a = 1$ and $a_i(s_t) = 0$ if $a = 0$. An expert's policy is calculated as

$$\pi^E(1 \mid s_t) = \frac{\sum_{i=1}^{N_d} a_i(s_t)}{N_d}.$$

To address the problem of accuracy varying depending on the expert settings, this study provides m types of experts and a corresponding basis reward function $r^k(s_t, a)$ $(k = 1,..,m)$. Table 1 lists the values of the reward and penalty for four categories into which the fixation duration (500 to 3000 ms) is divided, reflecting the knowledge mentioned at the end of the previous section. *TP* and *FP* denote success and failure for the prediction of enlargement, and *TN* and *FN* denote success and failure for the prediction of non-enlargement. Moreover, std is the standard value assigned to each expert. For example, when $m = 5$, the values of std are 1800, 2000, 2200, 2400, and 2600. This suggests that experts with smaller numbers tend to enlarge characters when the fixation time is short. The final reward is evaluated as a convex combination of these functions:

$$\text{üüüüüü}) = \sum_{k=1}^{m} {}^{k,E}\,{}^{k}(\ {}_{t},\ ), \sum_{k=1}^{m} {}^{k,E} = 1, \quad {}^{k,E} \geq 0.$$

Weight $w^{k,E}$ is the share ratio of each expert. Figure 3 shows the *TP* reward and *FP* penalty (absolute value) when $w^{k,E} = 0.2$ for each $1 \leq k \leq 5$. This demonstrates that rewards and penalties can be finely evaluated, suggesting that the expert with these rewards and penalties is an ideal expert. When $w^{k,E} = 1$, k-th expert is ideal. From this discussion, it is seen that an ideal expert can be characterized by the weight vector $\boldsymbol{w}^E = (w^{k,E})_{k=1,\ldots,m}$ which determines the expert's reward.

The average expected return of each expert is as follows:

$$\hat{V}^{k,E} = \frac{1}{T}\sum_{t=0}^{T-1} \hat{V}^{k,E}(s_t), \quad \text{where}$$

$$\hat{V}^{k,E}(s_t) = \sum_{i=t}^{T-1}\sum_{a} \gamma^{i-t} r^k(s_i, a)\pi^E(a \mid s_i).$$

The average expected return of an ideal expert is obtained by solving the following linear programming problem for $w^{k,E}$:

$$\hat{V}^E = \max_{\boldsymbol{w}^E} \sum_{k=1}^{m} w^{k,E}\hat{V}^{k,E}. \qquad (7)$$

Learning by Apprentice

An apprentice first finds a policy that satisfies (6) and then re-evaluates the weights that maximize the average expected return, because the return changes owing to the policy change.

Table 1. *Basis functions for reward and penalty*

State	Reward		Penalty	
	TP	*TN*	*FP*	*FN*
$500 \leq s_t \leq$ std - 300	0.25	2.00	-7.00	-0.25
std - 300 $\leq s_t \leq$ std	0.50	1.50	-6.00	-0.50
std $\leq s_t \leq$ std + 300	3.00	0.25	-0.50	-5.00
std + 300 $\leq s_t \leq$ 3000	5.00	0.25	-0.10	-10.00

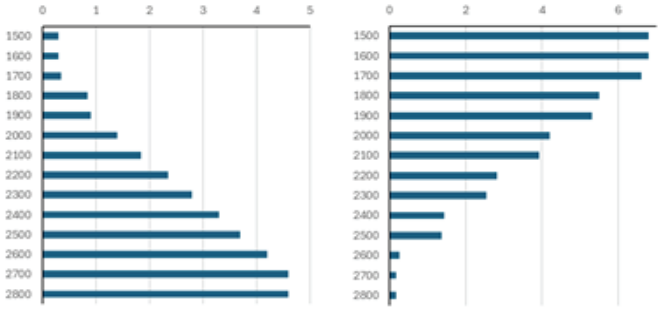


Figure 3. *Evaluation values for the TP reward and FP penalty*

Considering the discussion in the previous subsection, the re-evaluation process can be interpreted as the process of selecting a new expert. Similarly to the case of experts, the average expected return for the apprentice is expressed as

$$\hat{V}^{k,A} = \frac{1}{T}\sum_{t=0}^{T-1} \hat{V}^{k,A}(s_t),$$

$$\hat{V}^{k,A}(s_t) = \sum_{i=t}^{T-1}\sum_{a} \gamma^{i-t} r^k(s_i, a)\pi^A(a \mid s_i)$$

$$\hat{V}^A = \sum_{k=1}^{m} w^{k,E}\hat{V}^{k,A}. \qquad (8)$$

Note that the apprentice's weight is set to be the same as that of the expert, which corresponds to the best reward.

In what follows, the two-step learning approach is explained. In the first step, a policy that satisfies (6) is obtained by solving the linear programming problem:

$$\max_{\pi^A} \hat{V}^A - \hat{V}^{E^*} + \alpha\left(1 - \sum_t \left|\pi^A(1 \mid s_{t+1}) - \pi^A(1 \mid s_t)\right|\right)$$

$$\text{s.t.} \sum_a \pi^A(a \mid s_{t+1}) = 1 \qquad (9)$$

where $\alpha > 0$ is a constant and $\hat{V}^{E^*}$ is initially given as $\hat{V}^E$. Note that the constraint condition of (4) is incorporated into the objective function as a regularization term. It is easily seen that $\hat{V}^A \geq \hat{V}^{E^*}$ holds. Based on the policy π^A with (9), $\hat{V}^A$ is computed by $\hat{V}^A = \Sigma_{k=1}^m w^{k,E^*}\hat{V}^{k,A}$ where $w^{k\,E^*}$ is initially given as $w^{k,E}$. In the second step, assuming the existence of a new expert (*pseudo-expert*) with the policy π^A (i.e., the substitution $\pi^{E^*} \leftarrow \pi^A$), we optimize the weights to obtain a larger average expected return. Specifically, we solve the following linear programming problem:

$$\hat{V}^{E^*} = \max_{\boldsymbol{w}^{E^*}} \sum_{k=1}^{m} w^{k,E^*}\hat{V}^{k,A}$$

$$\text{s.t.} \sum_k w^{k,E^*} = 1, w^{k,E^*} \geq 0,\ 1 \leq k \leq m. \qquad (10)$$

The two linear programming problems (9) and (10) are solved until $\hat{V}^A$ converges. This algorithm is called Recurrent Two-Step Linear Program Apprenticeship Learning (**RTLPAL**). Figure 4 presents an overview of the RTLPAL algorithm. In summary, the apprentice first learns its own policy based on the expert's reward, and then the pseudo-expert is updated such that an optimal weight vector is found. These two processes are repeated until the apprentice's average expected return converges.

Experiment

The stimulus consisted of a virtual website featuring computer peripherals. The website contained images, names, and detailed descriptions of 15 products, with approximately 200 Japanese characters per description. The layout presented

Algorithm (RTLPAL algorithm)

Input: Compute $\hat{V}^{k,E}$ based on rewards and penalties in Table 1.
Derive maximal average expected return $\hat{V}^{E}$.

Initialization: $\hat{V}_1^{E^*} = \hat{V}^{E}, w_1^{k,E^*} = w^{k,E}$ for $k = 1, \ldots, m$.

for $l = 1,2, \ldots,$ **do**

Find a solution π_l^A to the linear program:

$\max_{\pi_l^A} \hat{V}_l^A - \hat{V}_l^{E^*} + \alpha(1 - \sum_t |\pi_l^A(1|s_{t+1}) - \pi_l^A(1|s_t)|)$

s.t. $\sum_a \pi_l^A(a|s_t) = 1$

where an apprentice's average expected return is computed by

$\hat{V}_l^A = \sum_{k=1}^m w_l^{k,E^*} \hat{V}_l^{k,A}$ $(\hat{V}_l^{k,A}$ is derived based on $\pi_l^A)$

If $|\hat{V}_l^A - \hat{V}_{l-1}^A| \le \varepsilon$ $(l \ge 2)$ **break**

Find a solution w_{l+1}^{k,E^*} to the linear program:

$\max_{w_{l+1}^{E^*}} \sum_{k=1}^m w_{l+1}^{k,E^*} \hat{V}_l^{k,A}$ s.t. $\sum_k w_{l+1}^{k,E^*} = 1, w_{l+1}^{k,E^*} \ge 0$ for $k = 1, \ldots, m$

Compute a pseudo-expert's average expected return:

$\hat{V}_{l+1}^{E^*} = \sum_{k=1}^m w_{l+1}^{k,E^*} \hat{V}_l^{k,A}$

end for

Return: $\pi^A(a|s_t) = \pi_l^A(a|s_t), \hat{V}^A = \hat{V}_l^A$

Figure 4. *RTLPAL algorithm*

the products vertically on a single page, with no page transitions. The font used was "MSP Gothic" with a size of 16 px, which is consistent with that employed on the Yahoo Japan website.

The participants were 10 undergraduate and graduate students (male) in Kanazawa Institute of Technology. The participants were requested to browse the stimulus site freely, presented on a 14-inch laptop PC (Panasonic CF-LF), and click the mouse when difficulty in character identification arose. During web browsing, participants' gaze movements were recorded using an eye tracker (Tobii Pro Spark; 60 Hz). To increase the frequency of the occurrence of character identification difficulty, the stimulus was blurred by using a blur function such that we could reduce the visual acuity of participants with visual acuity 1.0 or higher to 0.7. The average number of fixations (duration greater than or equal to 500 ms) was 436, in which the average number of fixations when the participants had difficulty identifying a character and fixed their eyes on it was 40.

The saccadic amplitude was defined as the distance between each pair of consecutive fixation points. The saccadic velocity at a certain point was calculated by dividing the saccadic amplitude at the temporal point by the difference between the two sampling times. Thus, all fixation durations, saccadic velocities, and amplitudes were obtained during this browsing. When saccadic velocities and amplitudes were introduced as external parameters in the next section, these were divided into three or nine categories such that the number of fixations in each category is approximately equal.

Figure 5 shows the total number of fixations by all participants in each category pair (i, j) (i corresponds to velocity and j corresponds to amplitude) when the external parameters are divided into nine categories. The figure indicates that fixations were concentrated around the diagonal of the category matrix, suggesting a high correlation between velocity and amplitude (correlation coefficient = 0.86).

Verification of RTLPAL algorithm

RTLPAL was applied to eye-tracking data from ten participants to predict the occurrence of character

i \ j	1	2	3	4	5	6	7	8	9
1	308	96	41	21	7	4	5	2	
2	104	82	60	61	56	55	41	21	5
3	71	245	50	50	14	12	15	20	8
4		63	306	22	74	11	4	3	2
5			27	312	5	105	28	6	2
6				19	312	5	129	9	11
7					18	293	47	121	6
8							216	191	78
9								112	373

Figure 5. *Total number of fixations in the category matrix*

identification difficulties. Twenty-one different experts were provided; specifically, the standard reward values std were set from 1000 to 3000 ms in 100 ms increments. The third and fourth categories in Table 1 were unified if std + 300 ≥ 3000. Moreover, the parameter of the objective function in (9) is set to $\alpha = 0.1$. The results of RTLPAL were compared with those of the conventional reinforcement learning SARSA [10]. Simultaneously, we examined how accuracy metrics (precision and recall) changed when the number of categories for saccadic velocity and amplitude were modified. **Precision (P)** and **recall (R)** are defined by the following formulas:

$$P = \frac{TP}{TP + FP} \text{ and } R = \frac{TP}{TP + FN},$$

where *TP, FP, TN*, and *FN* indicate the frequency of occurrence of the events mentioned in the third section. Since the decreases in *FP* and *FN* induce those of Errors 1 and 2, it is seen that the increases in precision and recall imply the decreases in Errors 1 and 2.

Figures 6 and 7 show the boxplots for precision and recall, respectively, comparing the results of RTLPAL and SARSA, with symbol × denoting the average. In both figures, the vertical axis indicates the value of each metric, and the horizontal axis indicates the number of categories. Table 2 lists the precision, recall, and the number of elements in the 9×9 matrix (denoted "U") for which a solution satisfying the constraint (3) was

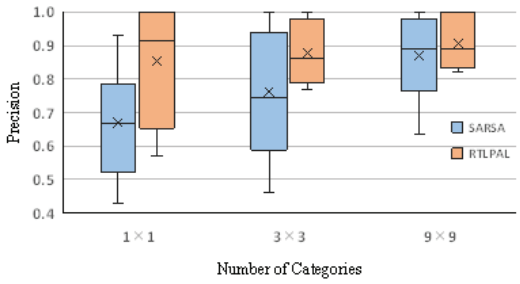


Figure 6. *Precision of RTLPAL and SARSA*

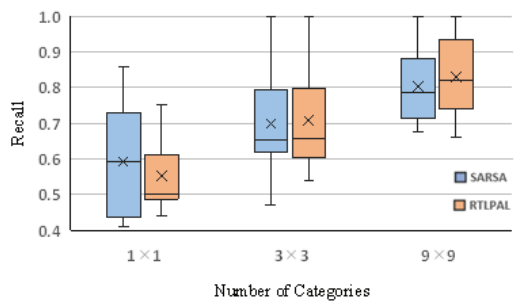


Figure 7. *Recall of RTLPAL and SARSA*

Table 2. *Accuracy metrics for each participant* (9×9)

No.	RTLPAL			SARSA		
	P	R	U	P	R	U
1	0.82	0.71	0	0.95	0.68	4
2	1.00	1.00	0	1.00	1.00	0
3	0.96	0.85	0	0.97	0.90	2
4	0.90	0.83	0	0.63	0.76	4
5	0.84	0.77	0	0.74	0.81	0
6	0.88	0.66	0	0.77	0.68	2
7	1.00	0.81	0	0.86	0.86	4
8	0.82	0.75	0	0.90	0.72	5
9	0.84	0.91	0	0.88	0.75	0
10	1.00	1.00	0	1.00	0.88	0
Average	0.91	0.83	0	0.87	0.80	2.1

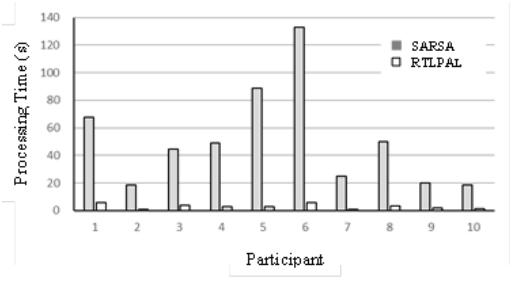


Figure 8. *Processing time of RTLPAL and SARSA* (9×9)

not obtained through learning, for each participant. Figures 6 and 7 show that both precision and recall improved as the number of categories increased, with the effect being more pronounced for recall than for precision. By Figure 6, the worst level of precision for RTLPAL increased (average 0.91) for the 9×9 partition. This suggests that RTLPAL can predict the occurrence of character identification difficulty with the frequency of Error 1 maintained at an acceptable level. In the case of the 9×9 partition, recall was at a good level (average 0.83), so RTLPAL is considered a suitable algorithm for decreasing the frequencies of Errors 1 and 2. Table 2 shows that, in some cases, a threshold could not be determined using SARSA, implying that SARSA is not suitable for determining the criteria for character enlargement. Meanwhile, RTLPAL always derives a threshold and demonstrates the effectiveness of incorporating the constraint into the objective function as in (9).

Figure 8 illustrates the processing times for RTLPAL and SARSA for each participant in the case of the 9×9 partition. The figure shows that RTLPAL significantly reduced the processing time compared with SARSA by up to approximately 1/20th (for Participant 6). In view of the development of a web browsing assistance system (i.e., systems where users browse websites daily and a massive amount of eye-tracking data is obtained), RTLPAL can be clearly a practical prediction method from a processing time perspective. Note that there was only one iteration. This is because the rewards and penalties assessed based on our knowledge were excellent (Table 1). Consequently, for all elements in the 9×9 matrix, $w^{k,E} = 1$ for some k(=1,…,9), and no updates to the pseudo-experts were necessary. However, this assessment based on

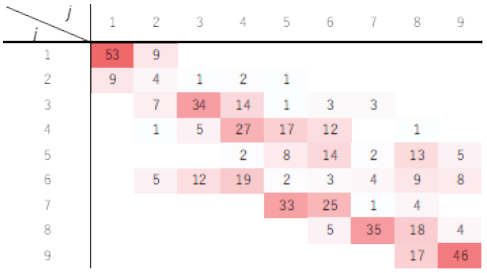


Figure 9. *The number of fixations* (*Participant* 1)

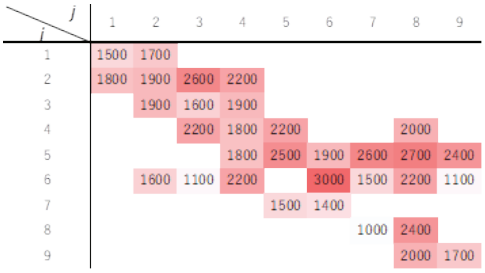


Figure 10. *Predicted threshhold for enlargement* (*Participant* 1)

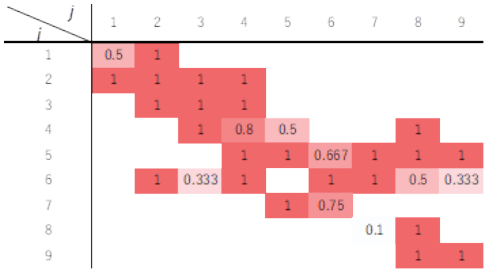


Figure 11. *Precision* (*Participant* 1)

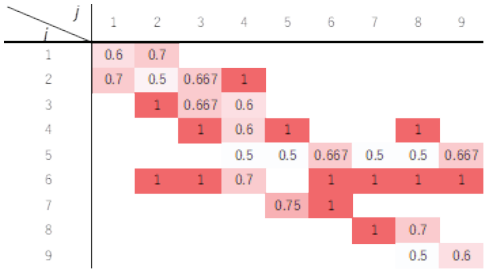


Figure 12. *Recall* (*Participant* 1)

prior knowledge may only have been valid in the experiment, which used a virtual website with no page transitions. Actual web browsing environments with page transitions may comprise unique web browsing characteristics that are beyond our knowledge. Therefore, regardless of prior knowledge, it is necessary to develop a systematic evaluation method, such as defining the basis function $r^k(s_t, a)$ of rewards and penalties as a step function. Furthermore, since the evaluation can increase the number of iterations, we need to investigate the changes in the processing time of RTLPAL.

We consider the reason for the low precision of Participant 1 in RTLPAL, as in shown Table 2. Figures 9 to 12 show the distributions of fixation count, threshold, precision, and recall for Participant 1. First, by Figure 11, the elements with extremely low precision were (6, 3)-th and (6, 9)-th elements.

Meanwhile, Figures 9 and 10 show that in both elements, the number of fixations was low, and the threshold was low (1100 ms). In fact, Participant 1 reported no difficulty in character identification even when gaze durations exceeded those associated with identification difficulty. This resulted in the threshold being set at a low level, causing Error 1 to occur frequently. This corresponds to the situation where the threshold was set at Level 1, as shown in Figure 1. The validity of this observation is verified by the high recall (1.0) for these elements, as shown in Figure 12. Therefore, this problem is expected to be mitigated as the sample size increases.

Conclusion

Aiming to build a web-browsing assistance system that enlarges characters when users have difficulty identifying them and fixate on them, this study proposed a method for predicting the occurrence of character identification difficulty during web browsing based on eye-tracking data. This prediction is unique in that it requires determining a threshold for character enlargement; specifically, characters must always be enlarged at all fixation points where fixation time exceeds the threshold even if no identification difficulty is present. Two approaches were employed to reduce the frequencies of Errors 1 and 2. Specifically, the threshold was determined in a combination of saccadic velocity and amplitude categories, and apprenticeship learning was used to reduce the burden of reward evaluation in the determination. Based on this approach, we developed an algorithm based on linear programming problems (RTLPAL), enabling flexible expert configuration and reducing the processing time. This algorithm features a recursive repetition of two processes: deriving the optimal apprentice's policy and generating an improved expert (reward). When applied to eye-tracking data from 10 participants during web browsing, RTLPAL improved the accuracy metrics (average precision 0.91, average recall 0.83) and reduced the processing time to approximately 1/20th of that required by SARSA. Furthermore, we confirmed that RTLPAL consistently yields a threshold, whereas SARSA does not always. This is because the constraints can be easily incorporated into the objective function of RTLPAL. These results suggest that RTLPAL is a suitable algorithm for determining character enlargement criteria. To efficiently utilize RTLPAL in a real-world web browsing environment, future work includes assigning initial experts regardless of prior knowledge and investigating how this affects the processing time. .

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 26K15028.

References

1. Sutton RS, Barto AG. Reinforcement Learning. 4th ed. MIT Press; 2018.
2. Matsushita Y, Aoki S. Prediction for occurrence of character identification difficulty during web browsing based on gaze data. Japan J Res. 2025;6(10):1-6.
3. Rayner K. Eye movements and information processing: 20 years of research. Psychol Bull. 1998;124(3):372-422.
4. Rayner K, Morris RK, Pollatsek A. Eye movement guidance in reading: the role of parafoveal letter and space information. J Exp Psychol Hum Percept Perform. 1990;16(2):268-281.
5. Goldberg JH, Stimson MJ, Lewenstein M, Scott N, Wichansky AM. Eye tracking in web search tasks: design implications. In: Proceedings of the 2002 Symposium on Eye Tracking Research & Applications (ETRA); 2002; New Orleans, LA. 51-58.
6. Abbeel P, Ng AY. Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the 21st International Conference on Machine Learning. 2004.
7. Abbeel P, Ng AY. Exploration and apprenticeship learning in reinforcement learning. In: Proceedings of the 22nd International Conference on Machine Learning. 2005.
8. Syed U, Schapire RE. A game-theoretic approach to apprenticeship learning. In: Advances in Neural Information Processing Systems. 2008.
9. Syed U, Bowling M, Schapire RE. Apprenticeship learning using linear programming. In: Proceedings of the 25th International Conference on Machine Learning. 2008.
10. Rummery GA, Niranjan M. Online Q-learning Using Connectionist Systems. University of Cambridge; 1994:23-86.