PAPER *Special Section on Data Engineering and Information Management*

Does Student-Submission Allocation Affect Peer Assessment Accuracy?

Hideaki OHASHI[†], ***Nonmember***, **Toshiyuki SHIMIZU**[†a)], ***and*** **Masatoshi YOSHIKAWA**[†], ***Members***

SUMMARY Peer assessment in education has pedagogical benefits and is a promising method for grading a large number of submissions. At the same time, student reliability has been regarded as a problem; consequently, various methods of estimating highly reliable grades from scores given by multiple students have been proposed. Under most of the existing methods, a nonadaptive allocation pattern, which performs allocation in advance, is assumed. In this study, we analyze the effect of student-submission allocation on score estimation in peer assessment under a nonadaptive allocation setting. We examine three types of nonadaptive allocation methods, random allocation, circular allocation and group allocation, which are considered the commonly used approaches among the existing nonadaptive peer assessment methods. Through simulation experiments, we show that circular allocation and group allocation tend to yield lower accuracy than random allocation. Then, we utilize this result to improve the existing adaptive allocation method, which performs allocation and assessment in parallel and tends to make similar allocation result to circular allocation. We propose the method to replace part of the allocation with random allocation, and show that the method is effective through experiments.

key words: *peer assessment, statistical models, MOOCs*

1. Introduction

Recently, online education platforms such as massive open online courses (MOOCs), in which a larger number of students participate in a single online class than in a conventional offline class, have become popular. For such large classes, peer assessment is a promising method for reviewing open-ended assignments, such as design problems and essays, which are difficult to review in an automated manner [1], [2]. In peer assessment, the number of submissions a student reviews does not depend on the total number of students because a large number of students review the work of other students instead of these reviews being performed by a small number of teachers or TAs. In other words, peer assessment offers scalability in such scenarios.

However, peer assessment has a problem of low reliability because it relies on students' reviews. Therefore, in peer assessment, a reviewing criterion called a rubric is often used, and a single student's submission is reviewed by multiple other students [3], [4]. In addition, methods of using statistical models to estimate a single reliable score by combining the scores given by multiple students have been proposed [1], [2], [5], [6]. However, the related studies have not focused on student-submission allocation, which could affect the score estimation.

In this study, we analyze the effect of nonadaptive student-submission allocation, which performs allocation in advance, on score estimation in peer assessment. Note that we assume each student's reviewing ability is unknown and there is no information other than the scores given by each student.

For the analysis, we focus on nonadaptive student-submission allocation patterns that satisfy the following conditions:

1. A student cannot grade his or her own submission.
2. A student cannot grade the same submission twice.
3. Each student reviews the same total number of submissions, and each submission is reviewed by the same total number of students.

We call these conditions the basic principles of student-submission allocation. Note that we refer to the number of submissions a student reviews as the "reviewing number" and to the number of students who review a submission as the "reviewed number". The first and second of the basic principles are obvious. The third is generally applied to avoid unfairness in the reviewing number and the reviewed number among the students.

We analyze three allocation methods: random allocation, circular allocation, and group allocation. These allocation methods are depicted in Fig. 1. Each node represents a student; the left side of each bipartite graph represents the reviewers, and the right side represents the reviewees. Note that nodes to which the same ID is assigned on the left side and the right side represent the same student. Each edge drawn from a reviewer to a reviewee represents a student-submission allocation. For example, in Fig. 1 (a), the student with ID 1 reviews the submissions of the students with ID 4 and ID 5. In Fig. 1, each student's reviewing number and reviewed number are 2, satisfying the third condition described above.

Figure 1 (a) illustrates random allocation. Random allocation is a method of simply allocating students to submissions randomly while satisfying the above conditions.

Figure 1 (b) illustrates circular allocation. Circular allocation is a method in which a certain order relation is assigned to the students (in this case, an order relation based on ID), and each student is allocated to review the submissions of the next k students, where k is the reviewing number. In this figure, the student with ID 1 reviews the submis-

Manuscript received June 18, 2021.
Manuscript revised November 16, 2021.
Manuscript publicized January 5, 2022.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
a) E-mail: tshimizu@i.kyoto-u.ac.jp

DOI: 10.1587/transinf.2021DAP0002

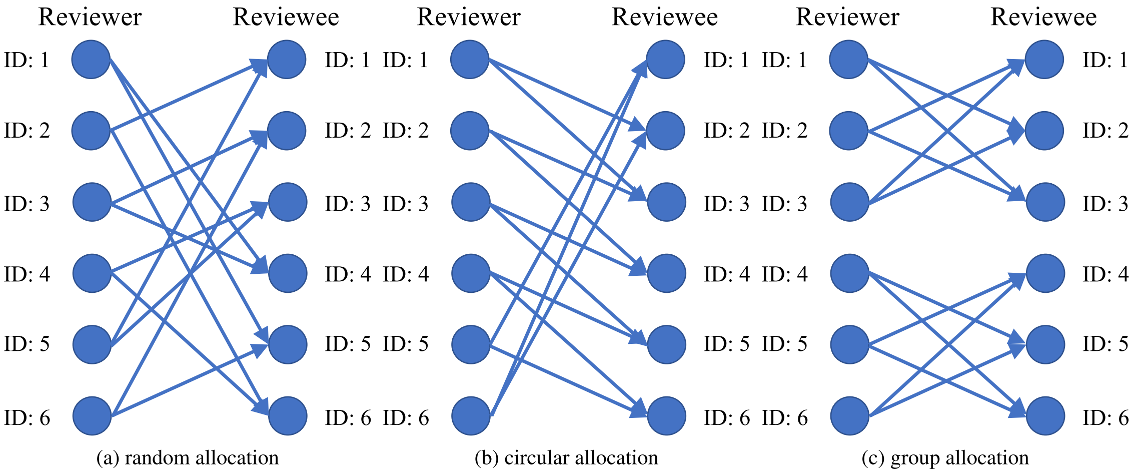


Fig. 1 Allocation patterns.

sions of the students with ID 2 and ID 3, the student with ID 2 reviews the submissions of the students with ID 3 and ID 4, and so on. This allocation method can easily satisfy the basic principles and is often used in actual peer assessment.

Figure 1 (c) illustrates group allocation. In group allocation, the students are divided into several sets, and student-submission allocations are generated within each set. In this figure, six students are divided into two sets of three students each, and allocation is performed within each of these sets. This method is also often applied to satisfy the basic principles. More specifically, if each student's reviewing number (and reviewed number) is k, the students are first divided into sets, each of which consists of $k + 1$ students. Then, the basic principles can be achieved by allocating each student to all submissions other than his or her own within his or her assigned group. In addition, group allocation is often used in an effort to divide a class in consideration of the students' profiles [7]. Note that, in practical cases, we cannot standardize the scores of students who belong to different groups [8]. In this study, we analyze the accuracy of group allocation method while avoiding score standardization among groups, and focus on showing that the accuracy of score estimation deteriorates as the number of people per group decreases due to group division.

In the experiments, we applied the above allocation methods to artificial data and real data and applied a typical score estimation method proposed by Piech et al. [1]. Then, we compared the allocation methods using the root mean square error (RMSE) as the evaluation index. Our experimental results show that circular allocation and group allocation tend to yield lower accuracy than random allocation.

In addition, using the above analysis result, we propose methods to improve accuracy for the existing adaptive allocation method, called RRB [9], which performs allocation and assessment in parallel and tends to make similar allocation result to circular allocation. In RRB allocation, an index called RR imbalance is used as an index of the inequality of the number of reviews. We propose a method that considers the trade-off between peer assessment accuracy and RR imbalance. The RRB allocation algorithm is an adaptive algorithm referring to students' reviewing order. The proposed method replaces the RRB allocation algorithm with a random allocation at specified intervals. In this study, we conduct experiments to confirm the usefulness of the proposed method using artificial reviewing order data and real reviewing order data. We perform the evaluation of the proposed method while changing the frequency of random allocation. As a result, it is confirmed that the peer assessment accuracy can be improved without impairing the RR imbalance by making a small substitution to the random allocation.

The remainder of this paper is organized as follows. Section 2 describes related work. In Sect. 3, we show the experimental results of the effect of student-submission allocation on score estimation. In Sect. 4, we propose the method for improving the existing adaptive allocation pattern and show its effectiveness. Finally, we conclude this work and suggest future work in Sect. 5.

2. Related Work

This section describes existing research on score estimation in peer assessment and on student-submission allocation.

2.1 Methods of Score Estimation in Peer Assessment

Early on, only a few studies were conducted in attempts to improve the accuracy of score estimation in peer assessment [10], but following a study by Piech et al. [1], an increasing number of such studies have emerged. Piech et al. proposed several statistical score estimation methods for peer assessment data. They first proposed a basic statistical method (PG1) and then extended PG1 to a method that exploits the estimation results for different assignments (PG2). In addition, they extended PG1 to a method based on the hypothesis that there is a correlation between a student's reviewing ability and the score of that student's own submission (PG3). Subsequently, Mi et al. proposed PG4 and PG5, which extend the relationship between a student's reviewing ability and the student's own score to a probabilistic relationship [11]. In the above studies, the score estimation methods are based on absolute evaluation, but methods based on relative evaluation have also been proposed [5], [6], [12], [13]. Research on relative evaluation has been motivated by the hypothesis that relative evaluation is easier for humans than absolute evaluation. Other similar studies include research involving matrix factorization [14] and work inspired by PageRank [15]. These studies were influenced by quality control research in the context of crowdsourcing [16]–[18]. Most of the above methods focus on the state of the reviewers for combining the scores by reviewers. In educational research fields, item response models, which focus on the reviewees, are often utilized to analyze grading results. Some item response models that incorporate the reviewer's parameters and apply to peer assessment results have been proposed [19]–[21].

There are also studies that have used data other than the score data to improve the accuracy of score estimation. For example, Chan et al. used data on students' social connections [22], and Sunahase et al. proposed a method using corrected parts in submissions [2]. In addition, a score estimation method for small private online courses (SPOCs), in which online lessons and offline lessons are conducted in parallel, is also under discussion [23].

2.2 Allocation Methods for Peer Assessment

In this section, we first focus on research that has considered the relationship between student-submission allocation and score estimation in peer assessment. Then, we mention the existing method that we improve in this study.

Marsico et al. examined whether the topology of the student-submission allocation graph affects score estimation in peer assessment and addressed research questions similar to those of our study [24]. However, Marsico et al. used a simple Bayesian-network-based model based on scores assessed by teachers [25] and analyzed the effect of propagation based on teachers' scores in the student-submission allocation graph. By contrast, we examine the accuracy of estimation for a general score estimation method [1], which does not assume scoring by teachers.

Chan et al. proposed a method of adaptively allocating students to submissions while sequentially estimating scores based on the strong assumption that each student's reviewing ability is known in advance [26]. Our research assumes that each student's reviewing ability is unknown and analyzes the relationship between student-submission allocation and score estimation in a batch estimation setting.

In this research, we improve the existing allocation method called RRB [9], which makes similar allocation result to circular allocation. The RRB allocation algorithm solves the imbalance in the number of reviews due to dropouts. We show the details in Sect. 4.1.

3. Analysis of the Effect of Student-Submission Allocation on Score Estimation

We analyze the effects of three types of student-submission allocation on score estimation. This section first describes the three student-submission allocation algorithms considered: random allocation, circular allocation, and group allocation. Then, we introduce the existing score estimation method applied in our experiments. Finally, we describe the experiments. We first show the experimental results for the artificial dataset, and then we present the results for the real dataset. Note that since the size of the real dataset is small, the experimental results for the real dataset are less reliable than the results for the artificial dataset. The reason why we utilize the small dataset is described in Sect. 3.4.1.

3.1 Allocation Algorithms

This study focuses on student-submission allocation patterns that satisfy the basic principles described in Sect. 1. The number of students is n, and the reviewing number and reviewed number are both k ($< n$). We consider a set of students $V = \{v_1, \dots, v_n\}$. Additionally, we consider a graph with the student set V as the node set, where the set of directed edges of this graph is denoted by E. Note that a directed edge (v_i, v_j) indicates that student v_i is allocated to review student v_j's submission. The following algorithms take as input a student set V and a reviewing number (reviewed number) k and output an edge set E that represents the student-submission allocations.

3.1.1 Random Allocation

The random allocation algorithm (algorithm 1) generates allocations that satisfy the basic principles randomly. Note that since the candidates to which a given student v_i can be allocated change during the sequential allocation process, they are managed by the variable $C(v_i)$. Specifically, given a Graph $G(V, E)$, $C(v_i)$ is the set of students, excluding v_i him- or herself, to whom submissions v_i have not yet been

Algorithm 1 Random Allocation Algorithm

```
INPUT: $V = \{v_1, \dots, v_n\}$        ▷ a set of $n$ students
INPUT: $k$                              ▷ reviewing (reviewed) number
OUTPUT: $E$                             ▷ student-submission allocations
 1: $E \leftarrow \{\}$
 2: for $i \leftarrow 1$ to $n$ do
 3:     Initialize $C(v_i)$
 4:     for $t \leftarrow 1$ to $k$ do
 5:         if $C(v_i)$ is empty then
 6:             go to 1
 7:         end if
 8:         $v_j$ is selected from $C(v_i)$ at random
 9:         $E \leftarrow E \cup \{(v_i, v_j)\}$
10:         Update $C(v_i)$
11:     end for
12: end for
```

Algorithm 2 Circular Allocation Algorithm

```
INPUT: $V = \{v_1, \dots, v_n\}$            ▷ a set of $n$ students
INPUT: $k$                                  ▷ reviewing (reviewed) number
OUTPUT: $E$                                 ▷ student-submission allocations
$E \leftarrow \{\}$
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $k$ do
        $E \leftarrow E \cup \{(v_i, v_{((i+j) \mod n)})\}$
    end for
end for
```

Algorithm 3 Group Allocation Algorithm

```
INPUT: $V = \{v_1, \dots, v_n\}$            ▷ a set of $n$ students
INPUT: $k$                                  ▷ reviewing (reviewed) number
INPUT: $d$                                  ▷ number of groups
OUTPUT: $E$                                 ▷ student-submission allocations
$E \leftarrow \{\}$
$l \leftarrow n/d$                          ▷ group size
for $i \leftarrow 1$ to $d$ do
    $V' \leftarrow \{v_{(l\cdot(i-1)+1)}, \dots, v_{l\cdot i}\}$
    $E \leftarrow E \cup RandomAllocationAlgorithm(V', k)$
end for
```

allocated and whose submissions have been assigned to be reviewed by fewer than k students each, as follows:

$$C(v_i) = \{v_j | v_j \neq v_i, (v_i, v_j) \notin E, N(v_j) < k\}$$

Here, $N(v_j)$ represents the number of students who are reviewing the submission of v_i.

Due to greedy allocation, the candidate set $C(v_i)$ may be empty. For example, when v_n is being assigned to submissions and the only candidate submission that is not already being reviewed by v_n is the submission of v_n him- or herself, $C(v_n)$ becomes empty. In this case, this algorithm terminates, as shown in the fifth line, and the allocation process repeats from the beginning.

3.1.2 Circular Allocation

The circular allocation algorithm (algorithm 2) allocates a given student to the submissions of students with adjacent IDs.

This algorithm differs from the random allocation algorithm in that the output is fixed to one type.

3.1.3 Group Allocation

The group allocation algorithm (algorithm 3) divides students into several groups and then performs random allocation in each group. In the following algorithm, the number of groups is represented by d. Note that the group allocation algorithm considers only cases in which n/d is an integer for simplicity.

If d and k are set such that $n/d = k + 1$, an algorithm that satisfies the basic principles can be easily realized in a manner similar to the circular allocation algorithm.

3.2 Estimation Method

In this study, we focus on the statistical estimation model known as PG1 [1]. As mentioned in Sect. 2, various score estimation methods have been proposed, but there are skeptical claims that there is actually little difference in accuracy among these methods [27]. Similar discussions have been taking place recently with respect to crowdsourcing, which is a field that is adjacent to peer assessment [28]. Therefore, in this study, we utilize PG1, which is a basic and representative method, instead of a more complicated and advanced method. The details of PG1 are as follows:

$$(Reliability)\ \tau_v \sim \mathcal{G}(\alpha_0,\ \beta_0)$$
$$(Bias)\ b_v \sim \mathcal{N}\left(0,\ \frac{1}{\eta_0}\right)$$
$$(True\ score)\ s_u \sim \mathcal{N}\left(\mu_0,\ \frac{1}{\gamma_0}\right)$$
$$(Observed\ score)\ z_u^v \sim \mathcal{N}\left(s_u + b_v,\ \frac{1}{\tau_v}\right)$$

$\mathcal{G}$ is a gamma distribution with fixed hyperparameters α_0 and β_0, while η_0 and γ_0 are the hyperparameters for the priors over the biases and true scores, respectively. $\mathcal{N}$ means a normal distribution. τ_v represents the reliability of student v, and b_v represents the bias of student v. s_u represents the true score of the submission created by student u, and z_u^v represents the score of the submission of student u as reviewed by student v. We estimate τ_v, b_v, and s_u in accordance with this model given each student's reviewing scores z_u^v.

In the estimation process performed in this study, Gibbs sampling was used, with α_0, β_0, η_0, and γ_0 all set to 1. The number of iterations was 3,000, and the first 1000 iterations were used for burn-in. PyMC3 was used for the implementation.

3.3 Experiment on the Artificial Dataset

We first explain the artificial dataset used in our experiments. Then, we give an experimental overview and compare the results obtained with the above three algorithms.

3.3.1 Artificial Dataset

For the artificial dataset, we consider a five-level evaluation, which is a typical situation of assessment, and generate artificial data from the last formula in Sect. 3.2. τ_v was generated as a uniform random number from 1 to 2, b_v was generated as a uniform random number from −1 to 1, and s_u was generated as a uniform random number from 1 to 5. Then, z_u^v was generated in accordance with the fourth equation of the PG1 model. Note that the range of s_u was set to 1 to 5 to consider a five-level evaluation. In addition, we set τ_v and b_v such that z_u^v would vary within approximately one level below or above s_u. For each simulation, we generated 500 data

Table 1 RMSE results and their sample standard deviation on the artificial dataset + random allocation simulation

n	PG1(3)	PG1(5)	PG1(10)	avg(3)	avg(5)	avg(10)	PG1/avg(3)	PG1/avg(5)	PG1/avg(10)
5	0.475 ± 0.169			0.498 ± 0.169			0.967 ± 0.198		
10	0.484 ± 0.120	0.373 ± 0.102		0.514 ± 0.118	0.399 ± 0.103		0.948 ± 0.144	0.945 ± 0.162	
20	0.482 ± 0.084	0.368 ± 0.065	0.261 ± 0.056	0.521 ± 0.086	0.404 ± 0.070	0.283 ± 0.056	0.930 ± 0.103	0.919 ± 0.130	0.928 ± 0.126
50	0.481 ± 0.052	0.365 ± 0.039	0.250 ± 0.030	0.522 ± 0.055	0.406 ± 0.043	0.288 ± 0.032	0.924 ± 0.067	0.901 ± 0.083	0.869 ± 0.082
100	0.483 ± 0.036	0.362 ± 0.029	0.246 ± 0.020	0.525 ± 0.039	0.406 ± 0.030	0.288 ± 0.022	0.921 ± 0.043	0.892 ± 0.058	0.856 ± 0.063
1000	0.484 ± 0.011	0.362 ± 0.009	0.241 ± 0.006	0.527 ± 0.012	0.408 ± 0.010	0.288 ± 0.007	0.918 ± 0.014	0.887 ± 0.020	0.836 ± 0.020

Table 2 RMSE results and their sample standard deviation on the artificial dataset + circular allocation simulation

n	PG1(3)	PG1(5)	PG1(10)	avg(3)	avg(5)	avg(10)	PG1/avg(3)	PG1/avg(5)	PG1/avg(10)
5	0.487 ± 0.180			0.513 ± 0.177			0.959 ± 0.199		
10	0.483 ± 0.113	0.375 ± 0.103		0.515 ± 0.122	0.397 ± 0.100		0.949 ± 0.137	0.953 ± 0.159	
20	0.490 ± 0.084	0.380 ± 0.074	0.263 ± 0.054	0.521 ± 0.085	0.401 ± 0.075	0.283 ± 0.059	0.944 ± 0.092	0.958 ± 0.145	0.938 ± 0.135
50	0.500 ± 0.054	0.383 ± 0.047	0.262 ± 0.034	0.528 ± 0.054	0.406 ± 0.049	0.288 ± 0.040	0.948 ± 0.056	0.947 ± 0.085	0.920 ± 0.116
100	0.498 ± 0.038	0.386 ± 0.033	0.263 ± 0.025	0.525 ± 0.037	0.408 ± 0.032	0.288 ± 0.027	0.949 ± 0.038	0.947 ± 0.060	0.917 ± 0.079
1000	0.500 ± 0.012	0.386 ± 0.011	0.263 ± 0.008	0.528 ± 0.013	0.409 ± 0.011	0.288 ± 0.009	0.947 ± 0.012	0.945 ± 0.019	0.911 ± 0.027

subsets in accordance with the specified number of students n, reviewing number k, and allocation algorithm.

3.3.2 Experimental Overview

In this section, we first compare random allocation and circular allocation and then compare random allocation and group allocation. The simulation results for the artificial dataset are shown in Tables 1 and 2. In these experiments, we performed 500 simulations and obtained the average RMSEs of the estimated values while varying the number of students n, the reviewing number k, and the allocation pattern as follows: random allocation (Table 1) or circular allocation (Table 2).

The reason why we did not explicitly perform group allocation is as follows. Group allocation is an allocation method in which the student set is divided into small groups and then random allocation is performed in each group. Therefore, when there are two sets of allocations with the same reviewing number, but one has a larger total number of students than the other, the one can be interpreted as random allocation, while the other can be interpreted as group allocation. Accordingly, in this study, to evaluate the performance of group allocation, we used the results of random allocation with a small number of students instead of results obtained explicitly through group allocation.

Now, let us explain how to read the experimental result tables using Table 1 as an example. The leftmost column represents the number of students n, and each row represents the results obtained using data generated under the assumption of n students. The second to fourth columns (PG1(k), $k = 3, 5, 10$) show the average and standard deviation on RMSEs of the estimated values when PG1 is applied to the generated 500 subsets of data. The values in the second column (PG1(3)) are the results for a reviewing number of 3, the values in the third column (PG1(5)) correspond to $k = 5$, and the values in the fourth column (PG1(10)) correspond to $k = 10$.

The fifth to seventh columns (avg(k)) show the average and standard deviation on RMSEs of the simple average ($\hat{s}_u = \sum_v z_u^v/k$). These three columns similarly present the results for $k = 3, 5, 10$.

The 8th to 10th columns (PG1/avg(k)) show the average and standard deviation of the values obtained by dividing the RMSE obtained with PG1 by the RMSE obtained through simple averaging for the 500 data subsets. The reason why we derive not only the average RMSE value of PG1 (PG1(k)) but also the average value of the ratio between the RMSEs of PG1 and simple averaging is as follows. When artificial data generated from the same distribution are used, the expected RMSE value of the simple average is independent of n. However, as seen from the avg(k) columns in Table 1, the RMSE value is smaller when n is smaller. We consider that sampling error occurred due to the use of the RMSE, and therefore, we need to normalize out this effect.

First, we will compare random allocation and circular allocation on artificial data based on Tables 1 and 2. Next, we will compare random allocation and group allocation using Table 1.

3.3.3 Random Allocation vs. Circular Allocation

As seen by comparing the values in the corresponding cells in the PG1/avg columns of Tables 1 and 2, the values in Table 1 are smaller in most cases. In particular, as n increases, the difference becomes larger. This finding indicates that random allocation is superior to circular allocation.

The cause of this difference in performance might be explained in the following manner. When we ignore the direction of edges in the allocation graph, the distances between nodes in the circular allocation graphs generally tend to be larger than the distances between nodes in the random allocation graphs. Since the PG1 model can be interpreted as recursively using adjacent values in the allocation graph at the time of estimation, it is difficult to use information from distant nodes for estimation. Thus, the greater the distance between nodes is, the more adversely the estimation may be affected.

Table 3 RMSE results and their sample standard deviations on the real dataset + random allocation simulation

n	PG1(3)	PG1(5)	PG1(7)	avg(3)	avg(5)	avg(7)	PG1/avg(3)	PG1/avg(5)	PG1/avg(7)
4	1.555 ± 0.486			1.467 ± 0.474			1.080 ± 0.168		
6	1.547 ± 0.401	1.349 ± 0.401		1.466 ± 0.385	1.231 ± 0.322		1.066 ± 0.123	1.105 ± 0.172	
8	1.582 ± 0.366	1.392 ± 0.318	1.299 ± 0.308	1.493 ± 0.347	1.276 ± 0.273	1.175 ± 0.253	1.066 ± 0.105	1.096 ± 0.142	1.108 ± 0.129
10	1.565 ± 0.339	1.394 ± 0.286	1.258 ± 0.249	1.496 ± 0.327	1.283 ± 0.252	1.153 ± 0.222	1.050 ± 0.097	1.092 ± 0.122	1.096 ± 0.128

3.3.4 Random Allocation vs. Group Allocation

We compare random allocation and group allocation based on Table 1. As seen from Table 1, the larger n is, the smaller the values of PG1/avg, indicating that random allocation without division of the students is superior to group allocation. Considering the recursive estimation method of the PG1 model, the available information increases as the value of n increases; hence, PG1 may function more effectively as a result.

3.4 Experiment on the Real Dataset

We explain the real dataset and then show the experimental results. The real dataset used in our experiment is small, so the results for the real dataset are less reliable than those for the artificial dataset. Note that we did not compare random allocation with circular allocation using the real dataset. This is because when the number of students n is small (e.g., $n = 5$ or 10), considering the simulation results for the artificial dataset, there is no significant difference between random allocation and circular allocation (see the results in Sect. 3.3.3).

3.4.1 Real Dataset

For the real data, we used an open word pair similarity dataset [29] whose true score is the gold standard score reported by Miller et al. [30]. Although this dataset was collected in the context of crowdsourcing research, the situation in which low-skilled workers review multiple tasks is similar to a peer assessment setting; therefore, it seems reasonable to use it in this experiment. In this dataset, workers can be interpreted as reviewers, and tasks can be interpreted as submissions. Note that the set of reviewers and the set of reviewees who created the submissions are not the same in this case, but this does not pose a problem for PG1 because this estimation method does not assume that the reviewer set and the reviewee set are the same.

In addition, this dataset consists of a total of 300 scores assigned to all 30 tasks by 10 workers. The data size is small, but the allocation graph is very dense. When performing a simulation using the real dataset, restoration extraction was performed 500 times from the data such that the specified n and k were satisfied. Therefore, it was desirable for the allocation graph of the original dataset to be tightly connected, but it was difficult to create a large amount of data with such an allocation graph. Hence, we considered the word pair similarity dataset to be suitable for our purposes, though the size of the dataset is small.

3.4.2 Random Allocation vs. Group Allocation

We perform a comparison between random allocation and group allocation on the real dataset. The results of the simulation are shown in Table 3. The approach for reading the table is the same as in the simulation on the artificial dataset. We consider the cases of $n = 4, 6, 8, 10$ and $k = 3, 5, 7$. The performance of PG1 is inferior to the performance of the simple average because the data are too few, but the ratio between PG1 and the simple average decreases as n increases. Therefore, it is suggested that random allocation is superior to group allocation based on this dataset.

All of the experimental results suggest that random allocation is superior to both circular allocation and group allocation.

4. Improving the Accuracy for the Existing Adaptive Allocation Pattern

In this section, we propose a method to improve the accuracy for the existing adaptive allocation pattern, which is called RRB [9], based on the observation that random allocation is superior to circular allocation.

4.1 RRB Algorithm

The RRB allocation algorithm is an adaptive allocation method to solve the imbalance in the number of reviews due to dropouts [9]. In the RRB allocation algorithm, students request to review a submission; then, a submission is allocated to the requesting student. Note that the submission of the student who contributes the most at each point in time, that is, the submission of the student whose difference between his or her reviewing number and reviewed number is the largest, is allocated with priority.

Let a student performing the i-th request under the adaptive allocation approach be $x_i \in V$. A submission by a student $y_i(\neq x_i) \in V$ is allocated to x_i before a student x_{i+1} can request a submission. This allocation is represented by a directed edge from x_i to y_i. In the graph G_i, let the set of students whose submissions are allocated to student $v \in V$ be $N_i(v)$ and $\bar{N}_i(v) = V \setminus \{N_i(v) \cup \{v\}\}$; then, $y_{i+1} \in \bar{N}_i(x_{i+1})$. The reviewing number (outdegree) of student v in graph G_i is defined as $\delta_i^+(v)(= |N_i(v)|)$, and the reviewed number (indegree) is defined as $\delta_i^-(v)$. The RRB algorithm determines y_{i+1} according to the following formula. Note that y_{i+1} is selected randomly when multiple candidates exist.

$$y_{i+1} \in \underset{v \in \bar{N}_i(x_{i+1})}{\arg\max}(\delta_i^+(v) - \delta_i^-(v))$$

The RRB algorithm adopts a greedy approach to reduce the difference between the reviewing number and the reviewed number, which is called RR imbalance. When the t-th allocation is finished, RR imbalance $I_t(V)$ can be calculated by the following equation:

$$I_t(V) = \sum_{v \in V} |\delta_t^+(v) - \delta_t^-(v)|$$

It is proven that RR imbalance can be limited to a certain amount when using the RRB algorithm. However, this allocation method does not take into account the effect on the peer assessment accuracy. We point out that the RRB algorithm may have an adverse effect on peer assessment accuracy.

When students review multiple submissions, students often review them collectively. Therefore, when applying the RRB algorithm to practical settings, students are expected to make requests continuously.

We consider an extreme situation where all students request submissions in succession. At this time, student-submission allocation is as shown in Fig. 2. Note that each node represents a student, and each edge represents the allocation from the reviewer to the reviewee. Additionally, in this figure, it is assumed that the students request three submissions in succession in the clockwise order from the student with ID 1.

First, before the student with ID 1 requests submissions, the difference between the reviewing number and reviewed number of all the students is 0, so three submissions of other students are randomly allocated to the student with ID 1. In this figure, we omit the nodes that are selected randomly in the RRB algorithm. In addition, we assume the submissions of students with IDs 1-6 are not subject to random allocation. Subsequently, when the student with ID 2 requests the submission, the difference between the reviewing number and the reviewed number of the student with ID 1 is the largest (= 3), so the submission of the student with ID 1 is allocated first to the student with ID 2, and then two other submissions are randomly allocated. The student with ID 3 is allocated to the submissions of ID 1 and ID 2 first, and then one submission is randomly allocated. The student with ID 4 is allocated to the submissions of ID 1, ID 2 and ID 3 with priority. Then, after the student with ID 4, the requesting student is allocated to the three submissions of students who request immediately before him- or herself.

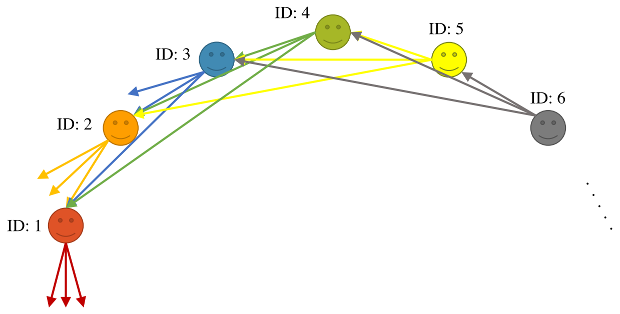


Fig. 2 Circular allocation-like example where RRB is applied.

In Sect. 3.3, it is pointed out that the allocation method called circular allocation has an adverse effect on peer assessment accuracy. Circular allocation is a method that assigns a certain order relation to students and allocates the next k students' submissions to students, where k is the reviewing number. The allocation in Fig. 2 is equal to the circular allocation except for students with IDs 1-3. Therefore, it is considered that the allocation in Fig. 2 also has an adverse effect on peer assessment accuracy.

4.2 Proposed Method

The proposed method is shown in algorithm 4. Here, V is a student set, S is a student's reviewing order, and h is an interval of random allocation. At every h request, the allocation in the RRB algorithm is replaced with random allocation.

4.3 Experiment

In this experiment, we utilize the reviewing order based on the real data and the artificial reviewing order and create the student-submission allocation using the proposed method while changing the interval h of random allocation. We first describe the details of the real reviewing order and the artificial reviewing order and then describe the experimental results.

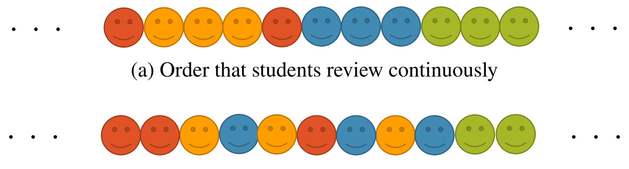
4.3.1 Reviewing Order

In this experiment, we use the data published by Canvas Network[†]. Specifically, we utilize those data whose class ID is 770000832960949 and whose assignment ID is 770000832930436 (denoted as real data 1) and those data whose class ID is 770000832945340 and assignment ID is 770000832960431 (denoted as real data 2). We construct the reviewing order using real data based on the time when the comments were created. Figure 3 shows examples of

Algorithm 4 RRB algorithm with partially random allocation

```
INPUT: $V = \{v_1, \ldots, v_n\}$                    ▷ a set of $n$ students
INPUT: $S = \langle v_{i_1}, \ldots, v_{i_m} \rangle$              ▷ reviewing order
INPUT: $h$                                ▷ an interval of random allocation
OUTPUT: $E$                                ▷ student-submission allocation
$E \leftarrow \{\}$
for $l \leftarrow 1$ to $m$ do
    if $l$ mod $h == 0$ then
        $v_j$ is selected from $V \setminus \{v_{i_l}\}$ at random
    else
        $v_j$ is selected according to RRB algorithm
    end if
    $E \leftarrow E \cup \{(v_{i_l}, v_j)\}$
end for
```

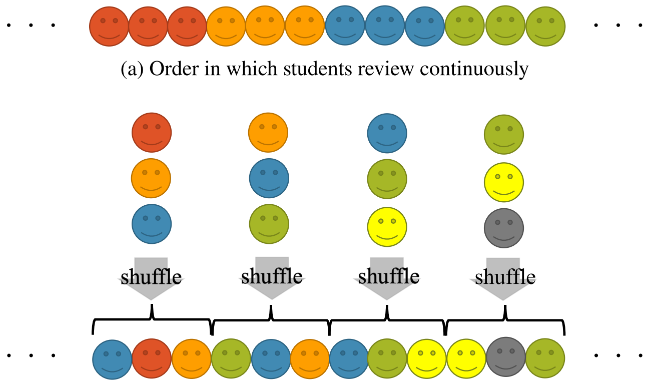
[†] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/XB2TLU



(a) Order that students review continuously

(b) Order that students review alternately

Fig. 3 A part of a real reviewing order.



(a) Order in which students review continuously

(b) Order in which students review alternately

Fig. 4 A part of the artificial reviewing order.

extracting part of the real reviewing order.

Each node in Fig. 3 represents a student who reviews a submission, and nodes of the same color represent the same student. Real reviewing orders shown in Fig. 3 are both examples in which students tend to review submissions collectively. Most of the requests are continuous in Fig. 3 (a), but the requests are alternated in Fig. 3 (b). Such an order in Fig. 3 (b) can occur when the number of students who make requests at a specific time is large. This situation occurs frequently in environments with a large number of participants, such as MOOCs. In this study, the following two artificial orders are created to consider two extreme cases.

The first artificial reviewing order is a sequence in which each student requests k times consecutively, assuming that the reviewing number is k (Fig. 4 (a)). Another artificial reviewing order consists of the steps described in algorithm 5. Figure 4 (b) shows the example of the reviewing order and how the algorithm works. With this algorithm, except for the first and last $k(k-1)/2$ students, the students in the set $\{v_i, \ldots, v_{i+k-1}\}(i = 1, \ldots, n-k+1)$ are shuffled, and the submissions are reviewed in order. This algorithm generates a sequence as shown in Fig. 4(b), which can be regarded as an extreme case of the order in which students review alternately.

4.3.2 Experimental Results

We present experimental results for the artificial reviewing order, and then we show the results for the real reviewing order. For each reviewing order, we demonstrate the RR imbalance and estimation accuracy when using the proposed adaptive allocation method while changing an interval h of random allocation. We show the estimation accuracy based on the same setting in the previous section.

Algorithm 5 Create artificial order in which students review alternately

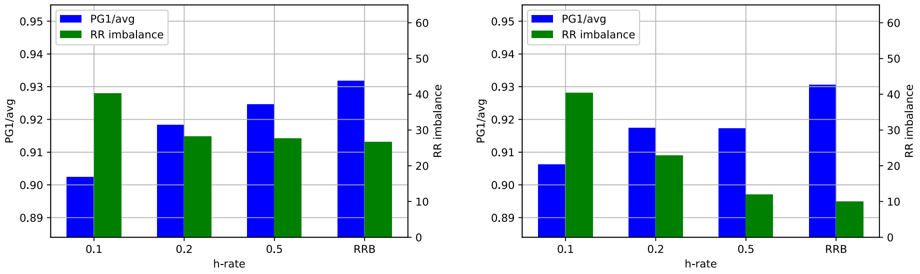
```
INPUT: $V = \{v_1, \ldots, v_n\}$                 ▷ a set of $n$ students
INPUT: $k$                                         ▷ reviewing number
OUTPUT: $S$                                        ▷ reviewing order
$S \leftarrow \langle\rangle$
for $i \leftarrow 1$ to $k-1$ do
    $S.append(shuffle(\langle v_1, \ldots, v_i \rangle))$
end for
for $i \leftarrow 1$ to $n-k+1$ do
    $S.append(shuffle(\langle v_i, \ldots, v_{i+k-1} \rangle))$
end for
for $i \leftarrow n-k+2$ to $n$ do
    $S.append(shuffle(\langle v_i, \ldots, v_n \rangle))$
end for
```

The results are shown in Fig. 5, 6 and 7. Each horizontal axis indicates a parameter h-rate that adjusts the size of an interval h of random allocation. Note that $h = \lceil n \cdot h\text{-rate} \rceil$, when n is the number of students. For example, if $n = 36$ and h-rate $= 0.2$, then $h = 7$. When h is larger than the length of the reviewing order, our proposed algorithm is equal to the RRB algorithm. Therefore, we represent the results of applying the RRB algorithm in the rightmost area in each figure. The vertical axis indicates the PG1/avg and RR imbalance, with the blue bar representing PG1/avg and the green bar representing the RR imbalance. Small values are preferable for both PG1/avg and RR imbalance.

First, we describe the results when using artificial data. Figures 5 and 6 show the results when the number of students $n = 100$ and $n = 1000$, and all the reviewing numbers are given by $k = 5$. In addition, each subfigure (a) shows the results with the order in which students review continuously, and each subfigure (b) shows the results with the order in which students review alternately.

In most cases, the RR imbalance decreases and PG1/avg increases as the h-rate increases, that is, the results of our proposed algorithm approach those of the RRB algorithm. In addition, the peer assessment accuracy (PG1/avg) is improved without impairing the RR imbalance by making a small substitution to the random allocation in some cases. For example, in Fig. 5, the RR imbalance when the h-rate is 0.5 is almost the same as that when using RRB, but the PG1/avg when the h-rate is 0.5 is lower than that when using RRB.

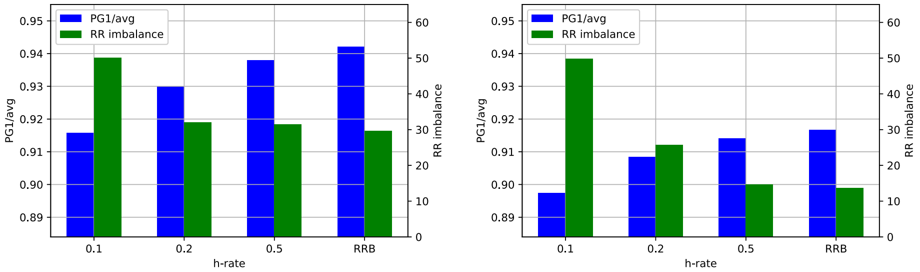
In each subfigure (a), except when the h-rate is 0.1, the values of the RR imbalance are similar. However, in each subfigure (b), the RR imbalance when the h-rate is 0.2 is clearly larger than that when the h-rate is 0.5. Therefore, when we focus on the RR imbalance, the proposed method is considered to be more effective for the order in which students review continuously than for the order in which students review alternately. On the other hand, when we focus on the values of PG1/avg, it is difficult to find an obvious difference between the two experimental results in each figure, but both results show the same tendency in that the PG1/avg increases as the h-rate increases.



(a) order in which students review continuously

(b) order in which students review alternately

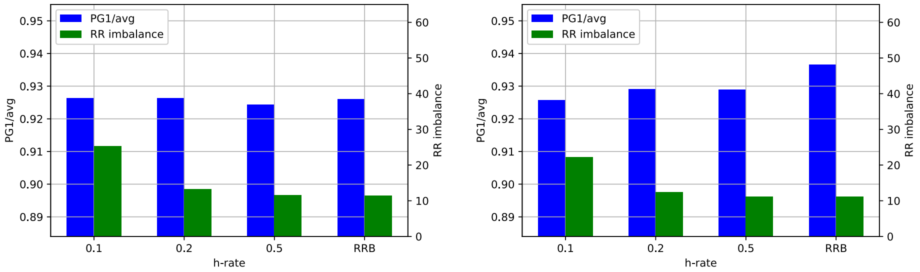
Fig. 5 Experimental results with the artificial reviewing order ($n = 100, k = 5$).



(a) order in which students review continuously

(b) order in which students review alternately

Fig. 6 Experimental results with the artificial reviewing order ($n = 1000, k = 5$).



(a) real data 1

(b) real data 2

Fig. 7 Experimental results with the real reviewing order.

In addition, we describe the experimental results when using real data (see Fig. 7). Real data 2 shows a similar tendency to the result when using artificial data, but real data 1 shows almost no change in estimation accuracy even if the *h*-rate changes. This suggests that the student-submission allocation on real data 1 created by the RRB algorithm contains enough randomness of allocation. This is because real data 1 has a large number of students, and the reviewing requests at a specific time are more crowded than the case in which algorithm 5 is used.

Through this experiment, we demonstrate that the peer assessment accuracy can be improved without impairing the RR imbalance by making a small substitution to the random allocation. How to set an interval *h* of random allocation appropriately is our future work.

5. Conclusion

In this study, we analyze the relationships between student-submission allocation and the accuracy of score estimation. We reveal the fact that circular allocation and group allocation, both of which are often used in peer assessment, have detrimental effects on the estimation results when using a typical statistical score estimation method. Then, we propose methods to improve the accuracy of score estimation for adaptive allocation methods. The proposed methods replace part of the allocation with random allocation. This study asserts the usefulness of the proposed method through simulation. Since this study offers only experimental results, we plan to further consider this issue from a theoretical perspective in the future.

References

[1] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller, “Tuned models of peer assessment in moocs,” Educational Data Mining, 2013.

[2] T. Sunahase, Y. Baba, and H. Kashima, “Probabilistic modeling of peer correction and peer assessment,” Educational Data Mining, 2019.

[3] K.J. Topping, “Peer assessment,” Theory into practice, vol.48, no.1,

pp.20–27, 2009.

[4] H.K. Suen, "Peer assessment for massive open online courses (moocs)," The International Review of Research in Open and Distributed Learning, vol.15, no.3, 2014.

[5] K. Raman and T. Joachims, "Methods for ordinal peer grading," Proc. 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.1037–1046, 2014.

[6] K. Raman and T. Joachims, "Bayesian ordinal peer grading," Proc. Second ACM Conference on Learning@ Scale, pp.149–156, 2015.

[7] H. Lynda, B.-D. Farida, B. Tassadit, and L. Samia, "Peer assessment in moocs based on learners' profiles clustering," 8th International Conference on Information Technology (ICIT), pp.532–536, 2017.

[8] M. Uto, "Accuracy of performance-test linking based on a many-facet rasch model," Behavior Research Methods, vol.53, no.4, pp.1440–1454, 2021.

[9] H. Ohashi, Y. Asano, T. Shimizu, and M. Yoshikawa, "Adaptive balanced allocation for peer assessments," IEICE Trans. Information and Systems, vol.E103-D, no.5, pp.20–27, 2020.

[10] J. Hamer, K.T. Ma, and H.H. Kwong, "A method of automatic grade calibration in peer assessment," Proc. 7th Australasian conference on Computing education-Volume 42, pp.67–72, 2005.

[11] F. Mi and D.Y. Yeung, "Probabilistic graphical models for boosting cardinal and ordinal peer grading in moocs," Twenty-Ninth AAAI Conference on Artificial Intelligence, pp.454–460, 2015.

[12] N.B. Shah, J.K. Bradley, A. Parekh, M. Wainwright, and K. Ramchandran, "A case for ordinal peer-evaluation in moocs," NIPS Workshop on Data Driven Education, pp.1–8, 2013.

[13] T. Wang, Q. Li, and J. Gao, "Improving peer assessment accuracy by incorporating relative peer grades," Educational Data Mining, 2019.

[14] J. Díez Peláez, Ó. Luaces Rodríguez, A. Alonso Betanzos, A. Troncoso, and A. Bahamonde Rionda, "Peer assessment in moocs using preference learning via matrix factorization," NIPS Workshop on Data Driven Education, 2013.

[15] T. Walsh, "The peerrank method for peer assessment," Frontiers in Artificial Intelligence and Applications, vol.263, pp.909–914, 2014.

[16] A.P. Dawid and A.M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," Journal of the Royal Statistical Society: Series C (Applied Statistics), vol.28, no.1, pp.20–28, 1979.

[17] P. Welinder, S. Branson, P. Perona, and S.J. Belongie, "The multidimensional wisdom of crowds," Advances in neural information processing systems, pp.2424–2432, 2010.

[18] Q. Liu, J. Peng, and A.T. Ihler, "Variational inference for crowdsourcing," Advances in neural information processing systems, pp.692–700, 2012.

[19] R.J. Patz and B.W. Junker, "Applications and extensions of mcmc in irt: Multiple item types, missing data, and rated responses," Journal of educational and behavioral statistics, vol.24, no.4, pp.342–366, 1999.

[20] M. Uto and M. Ueno, "Item response theory for peer assessment," IEEE transactions on learning technologies, vol.9, no.2, pp.157–170, 2015.

[21] M. Uto, D.-T. Nguyen, and M. Ueno, "Group optimization to maximize peer assessment accuracy using item response theory and integer programming," IEEE Transactions on Learning Technologies, vol.13, no.1, pp.91–106, 2019.

[22] H.P. Chan and I. King, "Leveraging social connections to improve peer assessment in moocs," Proc. 26th International Conference on World Wide Web Companion, pp.341–349, 2017.

[23] Y. Han, W. Wu, S. Ji, L. Zhang, and H. Zhang, "A human-machine hybrid peer grading framework for spocs," Educational Data Mining, 2019.

[24] M. De Marsico, L. Moschella, A. Sterbini, and M. Temperini, "Effects of network topology on the openanswer's bayesian model of peer assessment," European Conference on Technology Enhanced Learning, pp.385–390, 2017.

[25] A. Sterbini and M. Temperini, "Openanswer, a framework to support teacher's management of open answers through peer assessment," IEEE Frontiers in Education Conference (FIE), pp.164–170, 2013.

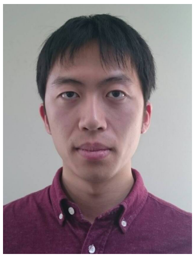
[26] H.P. Chan, T. Zhao, and I. King, "Trust-aware peer assessment using multi-armed bandit algorithms," Proc. 25th International Conference on World Wide Web, pp.899–903, 2016.

[27] M.S.M. Sajjadi, M. Alamgir, and U. von Luxburg, "Peer grading in a course on algorithms and data structures: Machine learning algorithms do not improve over simple baselines," Proc. third ACM conference on Learning@ Scale, pp.369–378, 2016.

[28] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?," Proc. VLDB Endowment, vol.10, no.5, pp.541–552, 2017.

[29] R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," Proc. conference on empirical methods in natural language processing, pp.254–263, 2008.

[30] G.A. Miller and W.G. Charles, "Contextual correlates of semantic similarity," Language and cognitive processes, vol.6, no.1, pp.1–28, 1991.



Hideaki Ohashi received his B.E. degree in 2015, his M.E. degree in 2017 and his Ph.D. degree in Informatics from Kyoto University in 2020. His research interests include educational data mining. He is a member of DBSJ.

Toshiyuki Shimizu received his B.E. degree in 2003, his M.S. degree in Information Science from Nagoya University in 2005, and his Ph.D. degree in Informatics from Kyoto University in 2008. He is currently an assistant professor in the Graduate School of Informatics, Kyoto University. His research interests include the handling of semistructured data, scientific data management, and scholarly databases. He is a member of the ACM, IPSJ, and DBSJ.

Masatoshi Yoshikawa received his B.E., M.E., and Dr. Eng. degrees from the Department of Information Science, Kyoto University, in 1980, 1982, and 1985, respectively. From 1985 to 1993, he was with Kyoto Sangyo University. In 1993, he joined the Nara Institute of Science and Technology as an associate professor in the Graduate School of Information Science. From June 2002 to March 2006, he served as a professor at Nagoya University. Since April 2006, he has been a professor at Kyoto University. His general research interests lie in the area of databases. His current research interests include privacy protection technologies, personal data markets, and multiuser routing algorithms and services. He is a member of the ACM and IPSJ.