Identifying Issues for Learners in Completing Online Courses on Machine Learning and Deep Learning: Five Issues Found in a Fully Automated Learning Environment for the Purpose of Scalable AI Education

Keisuke Seya*
Graduate School of System Design and Management, Keio University

Takayuki Okatani
Graduate School of Information Sciences, Tohoku University

Yutaka Matsuo
Department of Technology Management for Innovation, Graduate School of Engineering the University of Tokyo

Nobuyuki Kobayashi
The System Design and Management Research Institute of Graduate School of System Design and Management, Keio University

Seiko Shirasaka
Graduate School of System Design and Management, Keio University

ABSTRACT

Information technology is becoming increasingly sophisticated and rapidly developing. Although the demand for highly skilled technical human resources is rising, the supply is insufficient. Since this is a global problem, not only industries but also governments have become active in trying to find a solution. Online education is one method which enables training of a large number of people. However, our knowledge on how to train a large number of technical professionals in highly advanced emerging technologies such as artificial intelligence is insufficient. In this paper, we propose a fully automated online teaching method for learners who want to understand Machine Learning and Deep Learning and identify the issues that learners face in completing online courses on their own in a fully automated learning environment. This study concludes by presenting the five issues identified through the data collected from the Deep Learning and Machine Learning online courses designed by the proposed teaching method.

Keywords: Machine Learning, Deep Learning, Online Course, Artificial Intelligence.

Received 29 March 2019 | Revised 15 June 2019 | Accepted 3 July 2019.

1. INTRODUCTION

The advent of new technologies and services such as Big Data, IoT, and Artificial Intelligence (AI) means that IT utilization will be increasingly sophisticated and diversified in the future. Along with this, the demand for personnel with skills in these

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

technologies is likely to continue to increase. Several reports warn the shortage of advanced IT human resources (human resources engaged in AI, IoT, Big Data) will expand ("Gartner Survey Finds Talent Shortage Considered A Top Risk Among Executives," n.d.).

Machine Learning and Deep Learning are a particularly interesting category in the field of artificial intelligence in its third boom. We developed an online Machine Learning course between March 2016 and March 2017 to start human resource development in this field and it has been provided since April 2017. A Deep Learning course was also developed independently as an extension course of the Machine Learning course and has been provided since April 2018.

The goal of these two courses is not only to teach the theory of Machine Learning and Deep Learning but also to increase the number of practical technical professionals who can implement its technology. Therefore, practical engineering exercises cannot be omitted from the courses. However, it is not easy to properly assess and guide the growth of learners because each learner normally has a different level of knowledge and experience, which results in different learning outcomes when learning AI theory and when performing exercises on its realistic implementation (Felder and Brent, 2005). For that reason, advanced IT courses with such characteristics often take place in the form of a combination of face-to-face training and online learning (i.e., blended learning).

There are only a few Machine Learning and Deep Learning courses which have fully automated grading features. In fact, these two courses were the only qualified courses with fully automated grading features in Japan in October 2018, which were required to be taken before taking the E-certification exam ("Japan Deep Learning Association," 2018) standardized by the Japan Deep Learning Association. Therefore, the purpose of this research is to identify the issues that learners face in completing fully automated online courses on Machine Learning and Deep Learning on their own with a good understanding of both AI theory and its implementation.

Five issues were identified through the data collected from these two courses. First, basic mathematics knowledge and programming skill need to be checked to predict whether learners can finish online courses on their own. Second, low scorers in a basic pretest need some personal support. Third, it is difficult for learners to shorten their time spent on the exercises because the time spent between the exercises is almost the same among the learners regardless of their pretest score. Fourth, it is necessary to pay attention to the size of the variance of the number of trials on each exercise in order to judge which exercises need to be improved. Fifth, in order to make a smooth transition from AI knowledge to practical exercises, basic versions of the practical programming exercises in advance helps learners.

The remainder of this paper is organized as follows. Chapter 2 presents the research proposal. Chapter 3 describes the evaluation of the teaching method. Chapter 4 describes the improvement methods. Chapter 5 concludes.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

2. PROPOSED TEACHING METHOD FOR THE ONLINE MACHINE LEARNING AND DEEP LEARNING COURSES

2.1 Profile of Candidate and Course Syllabus

The online Machine Learning course and Deep Learning course that we developed were designed for learners with the same or similar intellectual level as a first-year university student. The goal of these two courses is to increase the number of engineers who not only understand AI theory but also can practically implement it. The course syllabus was created with reference to several leading AI massive open online courses (MOOCs) (“Machine Learning | ML (Machine Learning) at Georgia Tech,” n.d., “Machine Learning Engineer | Udacity,” n.d., “Machine Learning,” n.d.). Main subjects covered in the syllabus were selected from the perspective of fostering AI engineers capable of dealing with unknown problems in the future. Also, different types of solutions on the same problem are explained for the learners to be capable of selecting an appropriate solution under constraints they are given in the future.

2.2 Theoretical Basis for the Course Design

In order to evaluate the quality of the online course design and the learner's proficiency, a theoretical framework used to measure the learners’ performance was required. The learning theories that can be used as a theoretical framework have been studied for a long time. We adopted the argument of Ertmer and Newby (1993) as the theoretical basis for our course design, which is described below.

Ertmer and Newby (1993) argued their learning theory from the three perspectives of behaviorism, cognitivism, and constructivism, and discussed in detail the philosophical differences of these perspectives and the differences of teaching methods which arose from such philosophical differences. They suggested that learning theory serves as a basis to verify the correctness of pedagogical learning strategies and also as a basis for selecting specific strategies. These learning theories differ in terms of "how to define learning," and such difference yields different learning goals, teaching methods, and evaluation of learning outcomes.

Behaviorism focuses only on observable learner behavior changes. The changes that occur in the unobservable mind are considered to be ultimately manifested in behavior, and therefore only visible behavior is considered for evaluation. In other words, the definition of learning in this view is "the change of behavior (the ability to reflect acquired knowledge in behavior)."

Cognitivism focuses on cognition as a premise of behavior and posits that behavior changes when the way of cognition changes. The definition of learning from this point of view is "the change in cognitive structure (the ability to extract a pattern and use acquired knowledge)." Although there is a difference, behaviorism and cognitivism are the same in that the world is regarded as an object that can be learned about as an objective existence separate from the learner.

Constructivism does not treat the learner and the world as separate entities, and the world is understood as something that is given meaning and configured by the learner

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

itself. The definition of learning from this perspective is "finding the meaning by yourself (being able to create and use knowledge by yourself)."

Since learning is a complicated and time-consuming process and strongly influenced by the individual's experience and knowledge, Ertmer and Newby (1993) insist that appropriate learning strategies will change according to the learner's proficiency and learning goals. According to this view, in selecting a learning strategy, it is necessary to sufficiently consider both the learner's knowledge level and the cognitive processing level, which are required to handle the target learning task. Figure 1 (Ertmer and Newby, 1993, p.69) shows which learning theory-based learning strategy is appropriate for the learner, based on these two levels.

Figure 1 shows that learning strategies based on different learning theories overlap. However, it turns out that it is difficult to switch from behaviorism strategies to cognitivism strategies if the level of the learner’s task knowledge with behaviorism strategies does not increase. Similarly, it has been shown that it is difficult to switch from cognitivism strategies to constructivism strategies if the level of the learner’s task knowledge with cognitivism strategies does not increase.

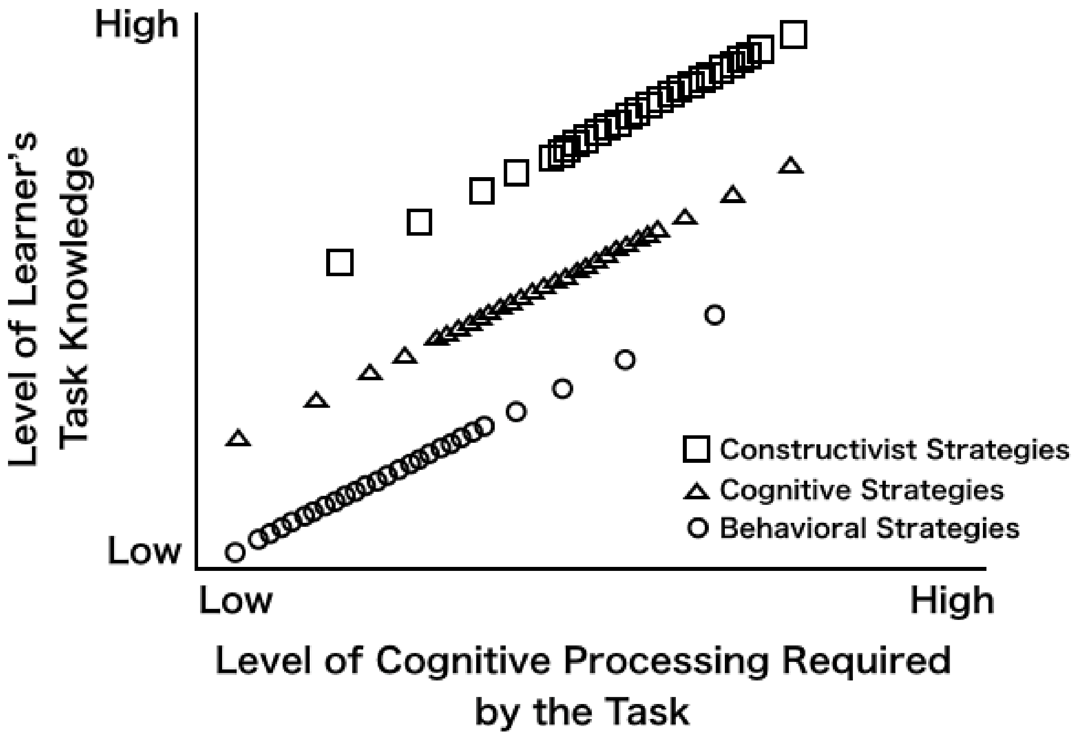

Figure 1. Comparison of the associated instructional strategies of the behavioral, cognitive, and constructivist viewpoints based on the learner's level of task knowledge and the level of cognitive processing required by the task (Ertmer and Newby, 1993, p.69).

2.3 Implementation of the Learning Theory and Course Architecture

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

In aligning the goals of the two courses with the learning strategies, it is better to use behaviorism strategies when learners are trying to accumulate knowledge about AI theory because learners' knowledge is too low to use cognitive strategies. At the stage of learning about how to implement the theory as working programming code, it is necessary to demonstrate that the acquired knowledge can be used effectively in practice, so a cognitivism learning strategy is appropriate.

Design methods for online courses have been studied in the field of instructional design and various instructional design models have been proposed. We propose a hybrid course architecture that combines behaviorism and cognitivism strategies (Figure 2) because the goals of the two courses do not require constructivist learning strategies. Here, the system built based on the behaviorism approach (Figure 2, left) is a system with the goal of acquiring knowledge, while the system built based on the cognitivism approach (Figure 2, right) is a system that allows learners to learn how to implement the knowledge through exercises. For the reasons above, we will call the behaviorism based system 'the knowledge learning management system' and the cognitivism based system 'the exercise management system' respectively.

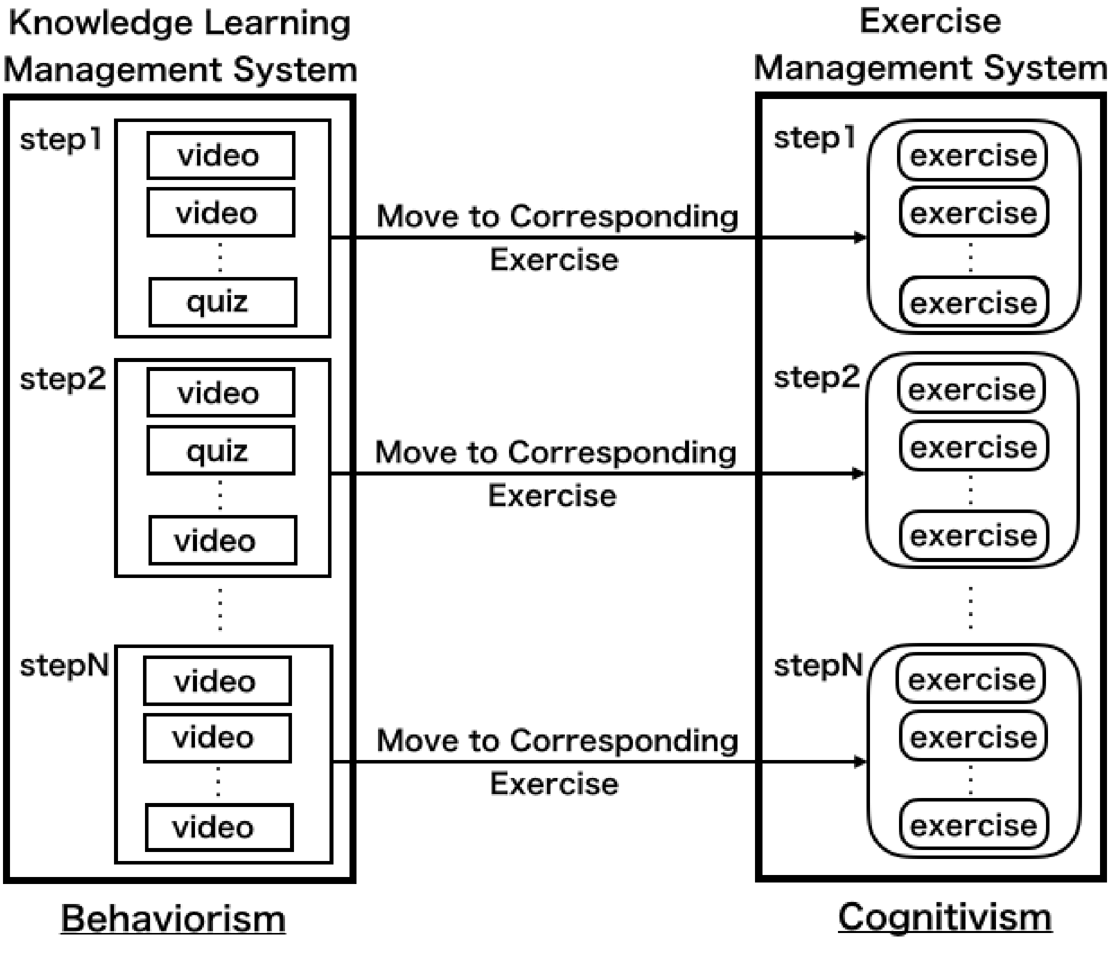

Figure 2. Hybrid course architecture.

Behaviorism learning strategies can be implemented using a conventional learning management system (LMS) that can handle lecture-style teaching methods using videos and texts (Figure 3, upper). The learning outcomes can be objectively evaluated by

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

testing the knowledge learned through the learning tasks, i.e. quizzes, placed in the learning units using the choice problem format (Figure 3, bottom).

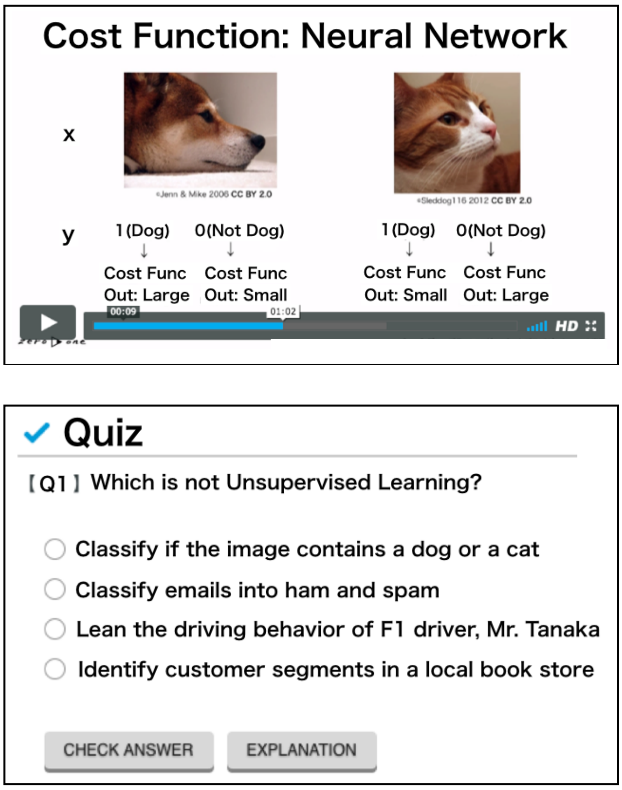

Figure 3. Example: Short video (top) and quiz (bottom)

In order to implement a cognitivism-based learning strategy, a traditional learning management system (LMS) cannot be used because it must provide the learners with an environment where programming code can be executed. Therefore, we prepared such a learning environment with Jupyter Notebook, which is an open source computational notebook ("Project Jupyter," n.d.). A computational notebook is an application like Microsoft Word, which can manage a text document as a file, but it can also execute programming code written in the document.

Although notebooks of this type have gained popularity in commercial math systems such as Mathematica ("Wolfram Mathematica," n.d.) and Maple ("Maplesoft - Software for Mathematics, Online Learning, Engineering," n.d.), Jupyter Notebook is open source, actively developed and supported. Many types of computer language are supported too. By using Jupyter Notebook, learners can write their own program and visually confirm the execution results in real time (Figure 4). In addition, we implemented an automatic scoring feature for the exercises in order to give the learners immediate feedback for what they did with them, and the executions of the exercises were recorded in a log file in detail. Moreover, we provided the results of the exercises visually so that the learners could monitor progress by themselves.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

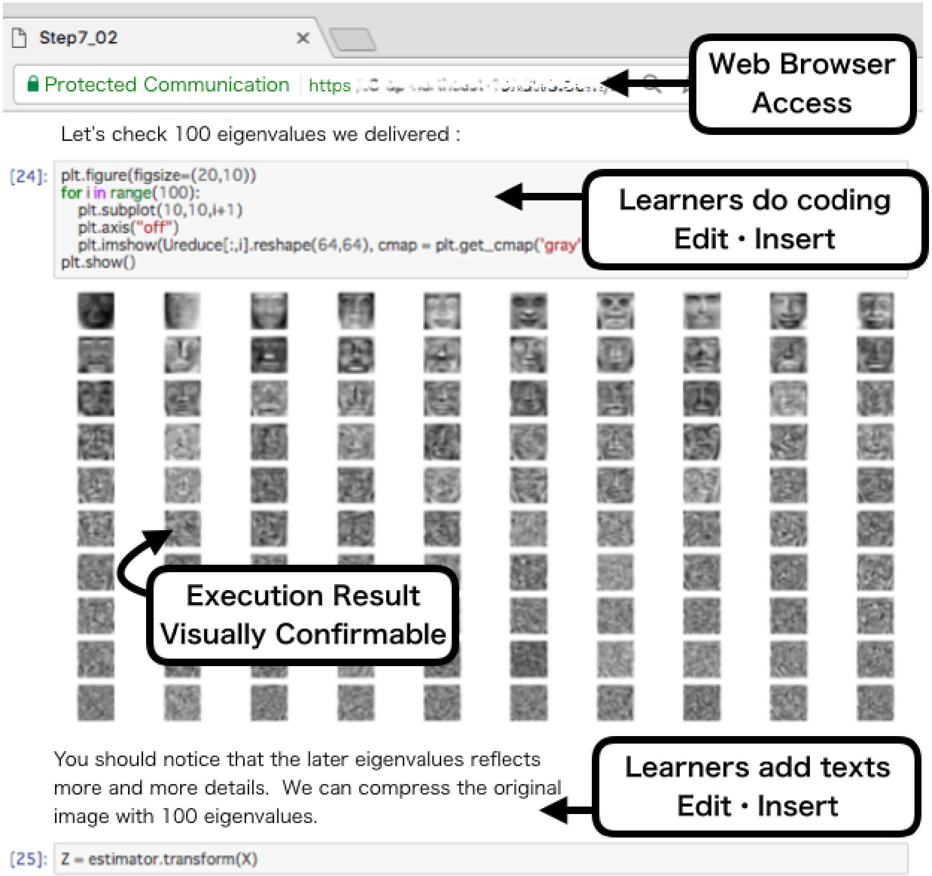

Figure 4. Exercise environment implemented with Jupyter Notebook.

3. EVALUATION OF THE TEACHING METHOD

Figure 1 suggests that efficient learning strategies differ depending on the level of learners' task knowledge (Ertmer and Newby, 1993). Therefore, it is important to understand the level of learners' task knowledge to provide an appropriate learning strategy for each learner. Figure 1 also suggests that learners found below the line of behaviorism learning strategies cannot work on a given task by themselves, so they won't appear in this area. Therefore, the distribution of the learners should appear to the upper right of this line. In fact, we observed the same pattern with the distribution of the learners in both the Machine Learning course and the Deep Learning course.

3.1 Evaluation of Implemented Learning Theory

If the implementation of learning theory is properly done, we should be able to obtain a Figure 1-like diagram with two types of collected data from the implemented system which represent the level of learners' task knowledge and level of cognitive processing required by the task. For example, we can create a diagram with the results of the pretest for the knowledge on the vertical axis and the number of trials each learner made over all the exercises at the end of the course on the horizontal axis; it will be like Figure 5. On the horizontal axis in Figure 5 the cognitive processing level is lower on the right

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

side of the figure because it is considered that the cognitive processing level is lower as the number of trials increases.

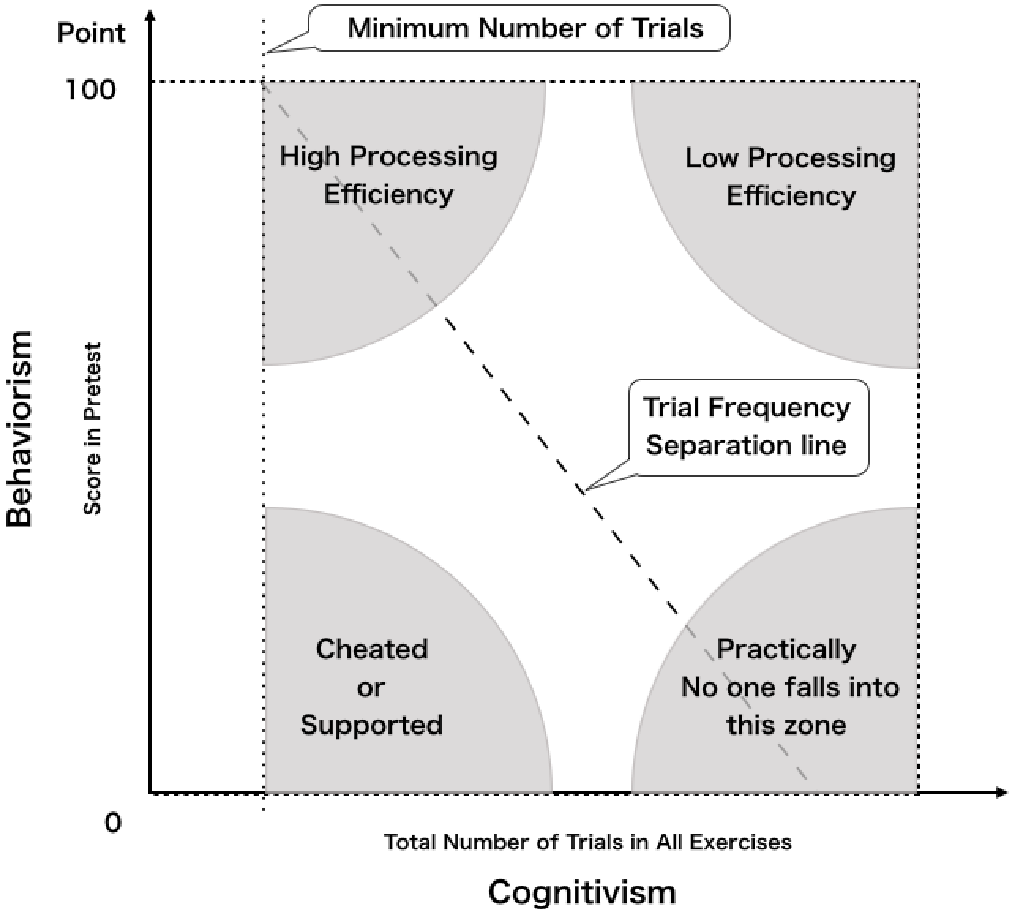

Figure 5. Trial frequency separation line.

The minimum number of trials is equal to the total number of challenges a learner performed on all the exercises. Therefore, the minimum number of trials is equal to the number of prepared exercises. For example, if 70 exercises are prepared for the course, learners have to go through at least 70 exercises, which makes the minimum number of trials 70.

If the students are found to be in the lower left area in Figure 5, it means their pretest score was extremely low and the number of trials they made on the exercises was extremely small. It is natural that learners with extremely low pretest scores are unlikely to clear exercises in a single attempt. Therefore, it is highly possible that they could have cheated or could have been assisted by a person who knew the correct answers. On the other hand, if students are found to be in the upper left area in Figure 5, it means their pretest score was very high and they cleared the exercises in only a few attempts. They are high performers in processing tasks. The students found in the upper right area in Figure 5 got high scores on the pretests, but they are learners who do not have strong ability to handle given tasks because they needed to try the given exercises many times to clear them.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

It is our assumption that we should be able to find a line like the trial frequency separation line described in Figure 5 and the learner distribution should appear in the upper right of this line. The reason is that Figure 5 corresponds to the gray part of Figure 6 which is drawn by reversing the left and right of the horizontal axis of Figure 1.

The upper part of the gray area in Figure 6 is aligned with a 100-point line of the pretest for knowledge assessment. The trial frequency separation line is aligned with the line that represents the behaviorism learning strategies (the line where the circles in Figure 6 are lined up). Also, the number of trials learners made to go through all the exercises is at least the number of given exercises, and it is not impossible to try the exercises as many times as learners want because the duration of the course is limited. Therefore, the processing level on the horizontal axis moves within a limited range. The gray area in Figure 6 is enclosed by this limited range on the horizontal axis and the range of knowledge levels is from 0 to 100 points on the vertical axis.

Since the learners who have knowledge beyond the knowledge tested on the pretest exist above the 100 point line in Figure 6, such learners appear in the gray area (pointed at by the solid down arrows).

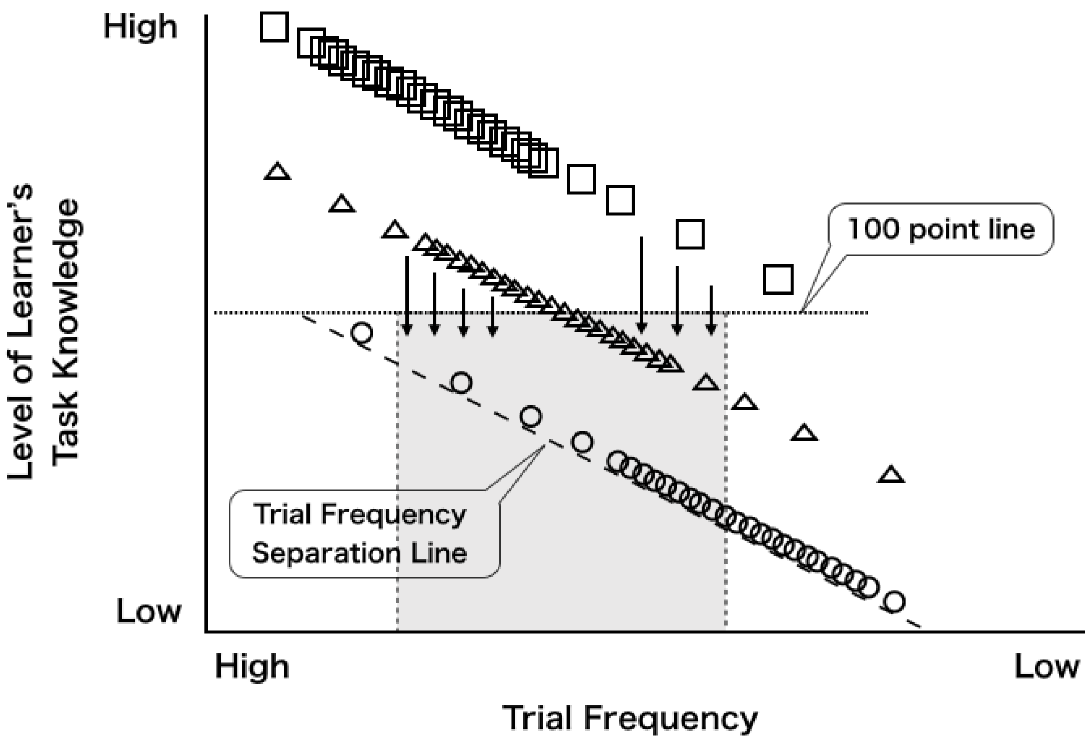

Figure 6. Trial frequency separation line in relation to Ertmer and Newby's Theory Line.

If a learner already knows answers for some reason, she or he will appear as a high performer in Figure 5. We cannot check whether they are cheating or not with this diagram alone. If checking for cheating is needed, it could be captured by monitoring other data collected by the system. For example, there is a tendency that the time

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

between exercises is extremely short or the number of mistakes is unnaturally few when such cheating occurs.

3.2 The relationship between the pretest score and the number of trials

When we designed the pretest for the knowledge assessment, the first problem was that we did not know what kind of questions should be asked to yield meaningful results which reflect the level of learners' knowledge. Since there was no sample pretest available, we decided to ask questions on basic mathematical knowledge and programming knowledge which frequently appear in our courses. The difficulty level of each question was set for the learners to be able to find the answer within a minute if they were already familiar with the question asked. Figure 7 shows some sample questions.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 0 \end{pmatrix} B = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$$

Calculate AB and answer the value (trace) that added the diagonal.

- ○ 1
- ○ 2
- ○ 7
- ○ 8
- ○ I don't know

Answer what's in c after the execution of the following code:

```
a = [3, 2, 0, 5]
c = a[1] + a[3]
print(c)
```

- ○ 3
- ○ 4
- ○ 7
- ○ 8
- ○ I don't know

Figure 7. Examples of pretest questions for mathematics and programming.

Figures 8 and 9 show the number of trials made by learners on all the exercises on the horizontal axis and the score of pretest on the vertical axis, on the Machine Learning course and Deep Learning course, respectively. On the Machine Learning course, there are 76 programming exercises in total. Therefore, a learner has to try to work on exercises at least 76 times. In other words, 76 is the minimum number of trials. Similarly, the minimum number of trials for the Deep Learning course is 77. In Figures 8 and 9 we can observe a similar line (dashed line in each figure), which corresponds to a trial number separation line. The fact that such a similar line is observed from the two different courses supports the idea that a pattern exists between the pretest score and the number of trials that the learners need to finish an exercise, as was suggested by Ertmer and Newby.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

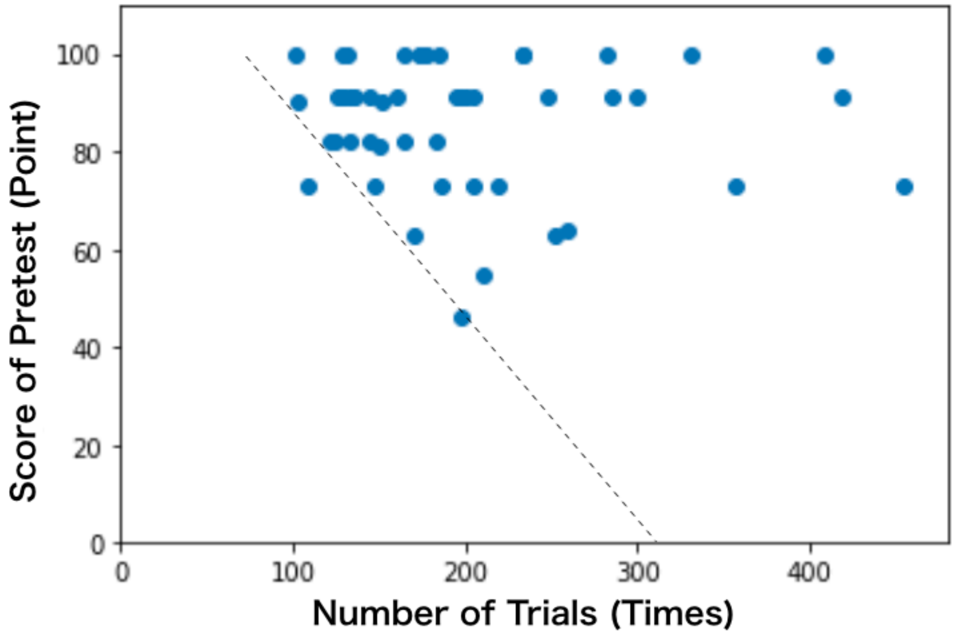

Figure 8. Trial frequency separation line for the Machine Learning course

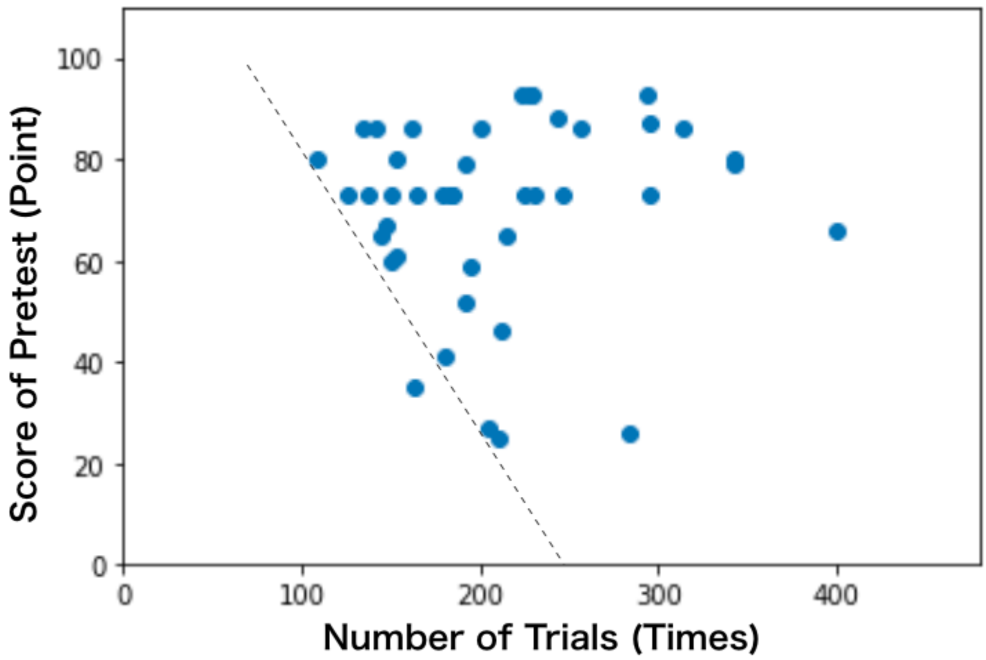

Figure 9. Trial frequency separation line for the Deep Learning course.

3.3 Possibilities of Self-Achievement in Online Learning

A personal support channel was provided by email for the learners who could not work on the exercises by themselves. We counted the number of personal support requests by each learner. Support inquiries not related to the exercises were also received via this support channel, but we did not include them in the count of the number of personal support requests. As a result, the distribution of learners who needed support for the exercises became clear as each of them appeared as a colored square mark in Figures

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

10 and 11 for the Machine Learning course and the Deep Learning course, respectively. The size of a square reflects the number of questions made by the corresponding learner.

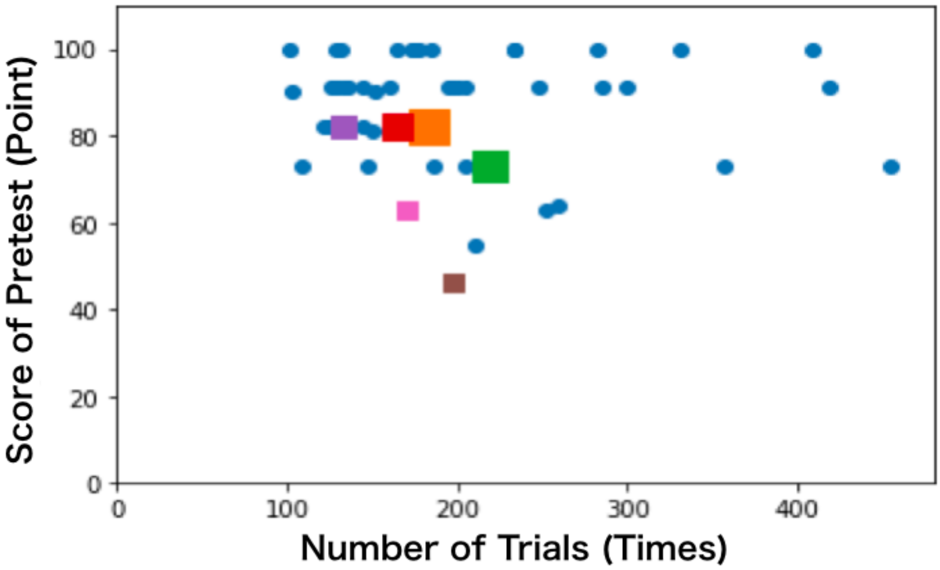

Figure 10. Number of exercises and actual support requests needed in the Machine Learning course.

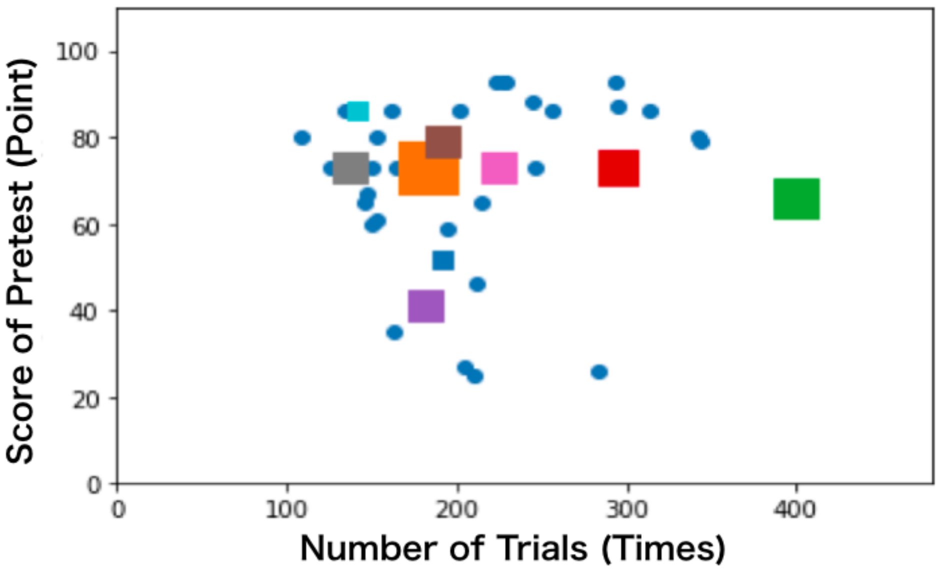

Figure 11. Number of exercises and actual support requests needed in the Deep Learning course.

In both courses, it is clear that the number of questions from high scorers (above 90 on the pretest) was zero. In other words, all the high scorers on the pretest finished the course by themselves without any support. On the other hand, learners who did not get high scores on the pretest, which was designed to check the basic knowledge of mathematics and programming skill, found themselves in trouble completing the exercises on their own. In fact, there is a sudden increase in the number of questions from the learners whose pretest scores were lower than 80. We observe this tendency in the pretest score range between 60 and 80 in Figures 9 and 10. Learners in this score

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

range might have reached a marginal level of knowledge to work on the exercises with some personal support.

Many learners want to know, before course entry, whether they have a sufficient level of knowledge to complete the courses on their own. During course development, we could not predict what kind of pretest would be useful to predict if learners were ready to take the courses. The data gathered through the course offerings suggest a pretest that measures learner's basic mathematics knowledge and programming skill seems to be a good way to predict whether learners can finish the course by themselves.

3.4 Effectiveness of the Presence of Study Peers

Several learners finished the course without any personal support even though their pretest scores were low. The reason for this phenomenon became clear through the interviews with the learners who took these courses; such learners took courses in groups of two or more people and had opportunities to have some personal support from their peers. Groups of more than two people who did not use the support channel always included high scorers on the pretest.

In the Machine Learning course, we found that a single learner, who took the course alone, needed about 18 times more support than a group learner, who took the course in a group (Figure 12). The same pattern was observed in the Deep Learning course. However, this difference is more prominent in the Deep Learning course than in the Machine Learning course as a single learner needed about 26 times more support than a group learner (see Figure 13).

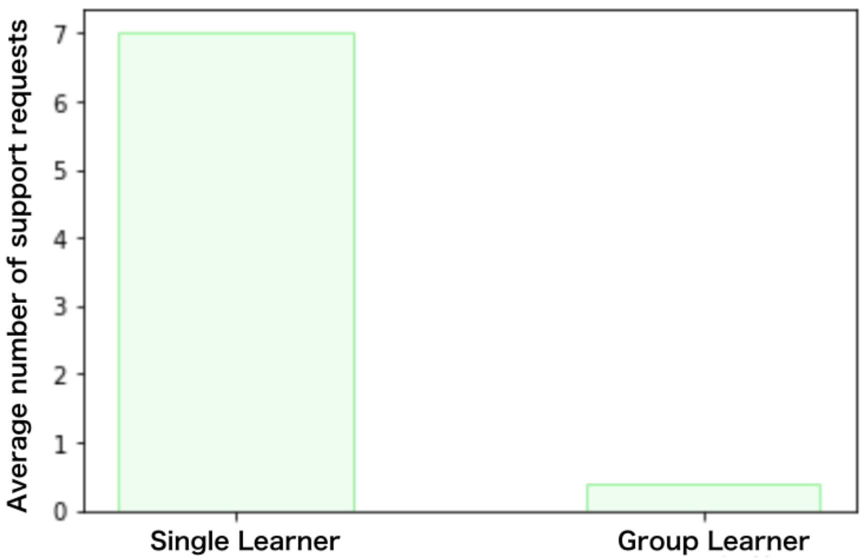

Figure 12. Average number of support requests in the Machine Learning course.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

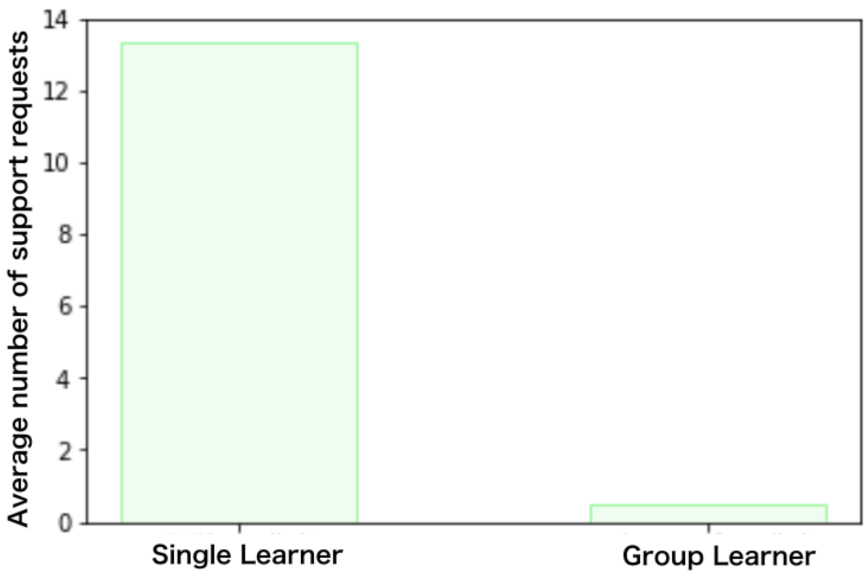

Figure 13. Average number of support requests in the Deep Learning course

These results indicate that low scorers on the pretest need a personal support environment. For example, it might be effective to use an online forum where peers can help each other.

3.5 Time Interval Between Exercises

High scorers on the pretest had a tendency to challenge exercises more than low scorers. We initially thought the time spent on one exercise would be shorter with high scorers than with low scorers. If this assumption is true, high scorers would finish the exercises sooner than low scorers. Since the time spent on one exercise could not be measured directly, the average time intervals between two adjacent exercises were investigated, and we found the difference of the time intervals was less than a minute among learners regardless of their scores on the pretest in the Machine Learning course (Figures 14) and was also less than a minute among learners whose pretest scores were between 40 to 100 in the Deep Learning course (Figure 15). It seems odd to find the low pretest scorers (between 20 and 40) have shorter intervals than high scorers (Figure 15). However, this aligns with the fact that they were supported by the learning peers or the support channel to find the answers for difficult exercises without struggling to find the solutions by themselves.

When using a computational notebook like Jupyter Notebook as an exercise environment, the finding that the interval between two adjacent exercises is almost constant regardless of the level of the learners suggests that it is difficult for the learners to shorten their time spent on the exercises. However, it also suggests that the minimum time needed to finish all the exercises can be estimated with quite a high accuracy.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

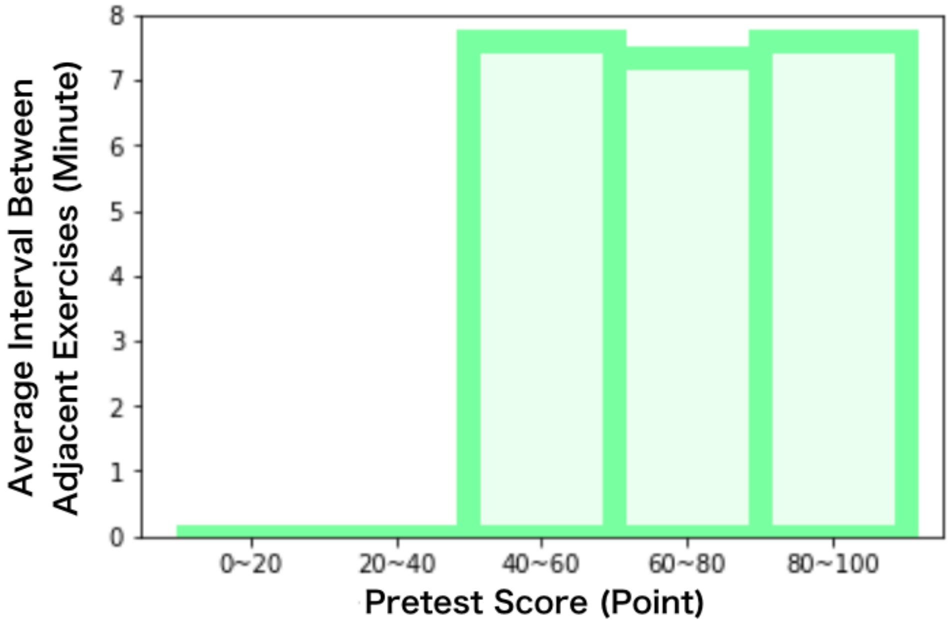

Figure 14. Average interval between adjacent exercises in the Machine Learning course.

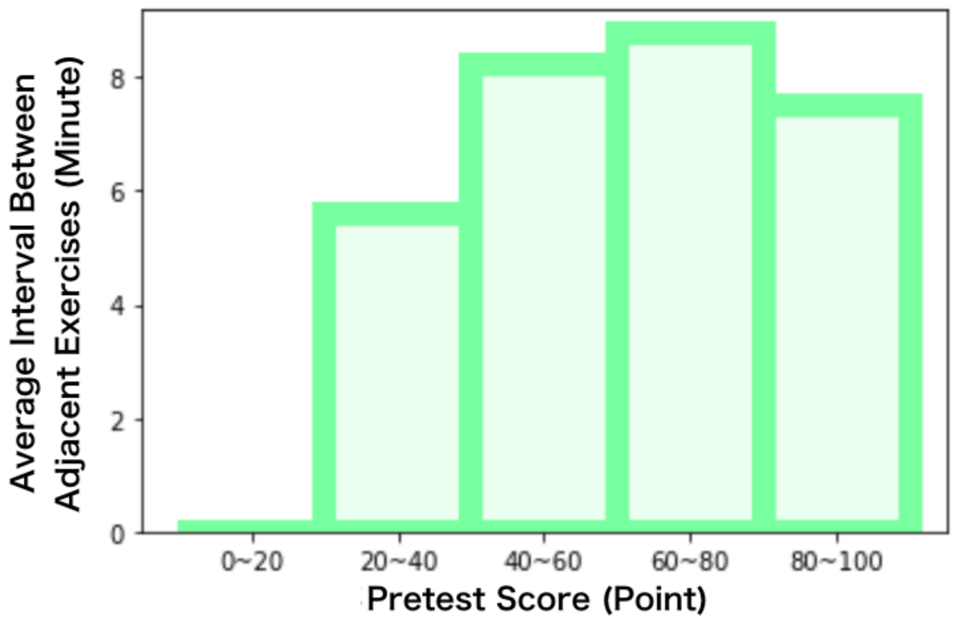

Figure 15. Average interval between adjacent exercises in the Deep Learning course.

4. IMPROVING COURSES

We cannot rule out possible problems originating from teaching methods and learning materials as the reason why learners cannot proceed with their learning effectively. Improving the courses in terms of teaching methods and learning materials helps to lower the support costs not only for the learners but also for the course providers.

4.1 Identifying the Exercises to be Improved

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

The exercises which need to be improved first are the ones that many learners actually needed help with through a support channel. However, even though the learners did not need much support, it might be necessary to review the exercises if many learners required many attempts to complete them.

Figure 16 is a scatter diagram of the Machine Learning course, with the horizontal axis representing the number of actual support requests, the vertical axis representing the number of trials on each exercise, and the size of a circle representing the variance of the number of trials on each exercise. The exercises which should be reviewed for improvement first are the ones with a large number of actual support requests and a large number of trials. Therefore, the exercises that are located in the upper right area in Figure 16 should be improved first. Normally it is difficult to improve all the exercises because it would cost too much. For that reason, it is helpful to determine an improvement control limit line, shown as a dotted line in Figure 16, and try to improve the exercises which appear to the upper right of the line. However, in practice, not all online courses can provide a support channel because it costs too much, so it is not always possible to create scatter plots like Figures 16 and 17. Therefore, it is useful if we can identify which exercises need to be improved without drawing an improvement control limit line.

The exercises mapped to the upper right area of the improvement control limit line as shown in Figure 16 have a large variance in the number of trials (i.e., the size of a circle is large). The same pattern is found for the Deep Learning course, as shown in Figure 17. These results suggest that the exercises which need to be improved can be identified simply by checking if the variance of the number of trials on each exercise exceeds predetermined criteria.

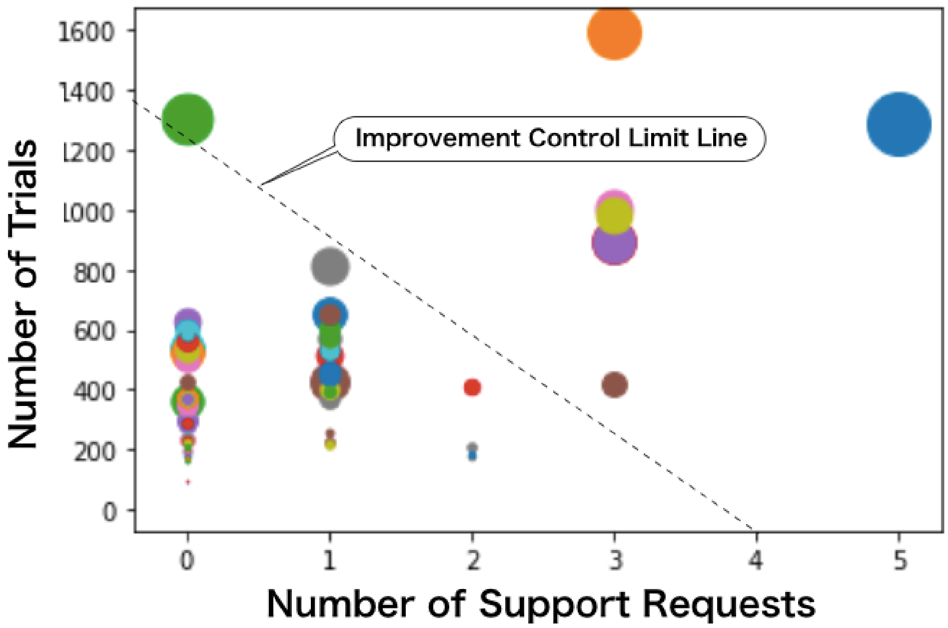

Figure 16. Improvement control limit line for the Machine Learning course.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

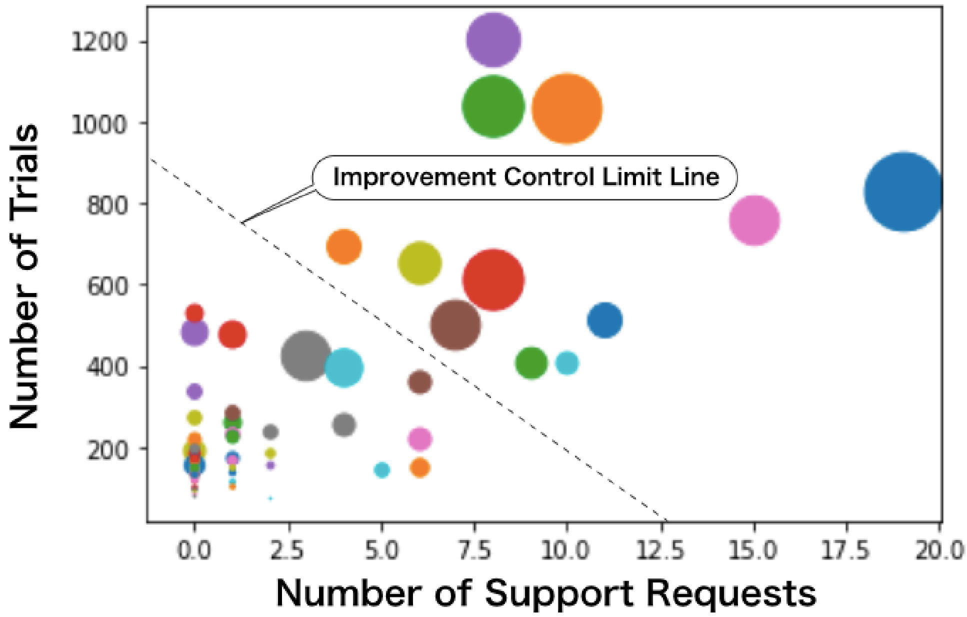

Figure 17. Improvement control limit line for the Deep Learning course.

4.2 Improving the Teaching Method

Since the learning data are collected on the premise of a specific teaching method that the learning platform adopted, in principle it is impossible to identify areas for improvement in the teaching method from the data. Therefore, we investigated areas for improvement of the teaching method through feedback from the learners who took our courses. The feedback indicated that many experienced a gap when transitioning from the behaviorism learning approach to the cognitivism learning approach.

In order to address this problem, we modified the teaching method so that learners could practice basic versions of the practical programming exercises. The basic versions only provided the essence of the exercises on the LMS, which plays the role of the knowledge learning management system in our system (Figure 18).

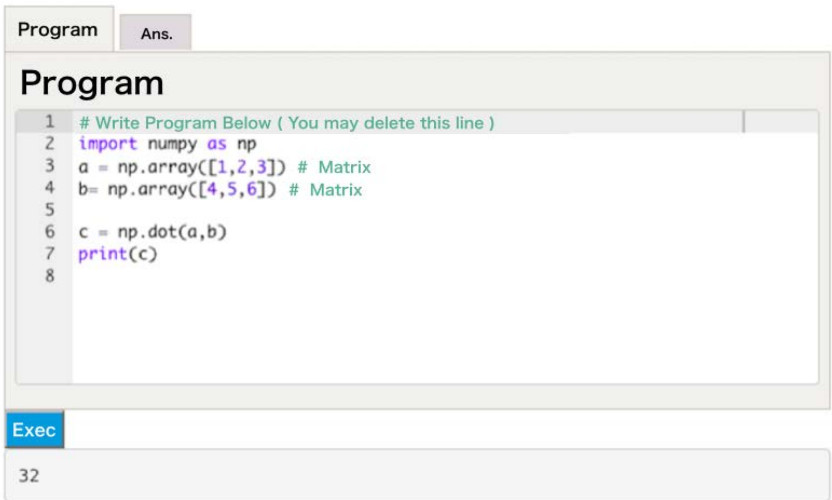

Figure 18. Programming window implemented on the LMS.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

By introducing such a programming exercise environment on the knowledge management system side, we found, through the interview after the completion of the courses, about 39% of learners enjoyed this feature.

Common open source learning management system (LMS) packages such as Moodle ("Moodle - Open-source learning platform | Moodle.org," n.d.), which have been adapted in many higher educational institutions, do not provide a programming environment as default. Therefore a special implementation must be carried out to provide a programming environment (Rodríguez-del-Pino et al., 2012). It would trigger the expansion of advanced technology online courses if popular LMS packages were to support such a programming environment by default in the future.

5. CONCLUSION

The purpose of this research was to identify the issues faced by learners in completing fully automated online courses on Machine Learning and Deep Learning on their own with a good understanding of both AI theory and its implementation. Five issues were found:

First, basic mathematics knowledge and programming skill need to be checked to predict whether learners can finish online courses on their own. Second, low scorers in a basic pretest need to have some personal support. Third, it is difficult for learners to shorten their time spent on the exercises because the time spent between the exercises is almost the same among the learners regardless of their pretest score. Fourth, it is necessary to pay attention to the size of the variance of the number of trials on each exercise in order to judge which exercises need to be improved. Fifth, in order to make a smooth transition between AI knowledge and practical exercises, basic versions of the practical programming exercises in advance helps learners.

The gap between AI knowledge and practical exercises corresponds to the gap between the abstraction and its implementation. Much research has tried to address this gap. Kobayashi et al. (2017) discussed this gap in the context of the gap among vision, strategy, business process, and IT system and tried to solve this issue by the assurance case method, while we did not investigate the gap in this context. Seya and Shirasaka (2016) tried to minimize the gap related to the learner's knowledge and experience level by utilizing the Open-Closed Principle (Meyer, 1988). Future research needs to identify the differences between existing teaching methods in order to utilize each approach appropriately and effectively.

It is difficult to supply the large number of technology-ready people needed by the emerging technology sector like AI only by direct teaching methods in classes typically conducted in the educational fields such as universities and graduate schools. In order to supply tens of thousands of leading-edge technical professionals required by the emerging technology sector, future research on the five issues identified by this research needs to be carried out, because online education is one of the promising methods to solve this problem.

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

Besides, as digital technology becomes increasingly influential, it is important to find an effective teaching method not only for technical people but also for non-technical people (Seya et al., 2019). Future research needs to reveal how different teaching methods and approaches can solve unique issues for different types of learner trying to understand Machine Learning and Deep Learning.

ACKNOWLEDGEMENT

This research was supported by the Keio University Doctorate Student Grant-in-Aid Program.

REFERENCES

[1] Ertmer, P.A., Newby, T.J., 1993. Behaviorism, Cognitivism, Constructivism: Comparing Critical Features from an Instructional Design Perspective. Performance Improvement Quarterly 6, 50–72.

[2] Felder, R.M., Brent, R., 2005. Understanding Student Differences. Journal of Engineering Education 94, 57–72. https://doi.org/10.1002/j.2168-9830.2005.tb00829.x

[3] Gartner Survey Finds Talent Shortage Considered A Top Risk Among Executives [WWW Document], n.d. MarketWatch. URL https://www.marketwatch.com/press-release/gartner-survey-finds-talent-shortage-considered-a-top-risk-among-executives-2018-10-02 (accessed 3.24.19).

[4] Japan Deep Learning Association [WWW Document], 2018. . Japan Deep Learning Association. URL http://www.jdla.org/en (accessed 3.23.19).

[5] Kobayashi, N., Nakamoto, A., Kawase, M., Sussan, F., Shirasaka, S., 2017. What Model(s) of Assurance Cases Will Increase the Feasibility of Accomplishing Both Vision and Strategy?, Review of Integrative Business and Economics Research, 7 (2), 1-17.

[6] Machine Learning | ML (Machine Learning) at Georgia Tech [WWW Document], n.d. URL http://ml.gatech.edu/ (accessed 3.23.19).

[7] Machine Learning Engineer | Udacity [WWW Document], n.d. URL https://www.udacity.com/course/machine-learning-engineer-nanodegree--nd009t (accessed 3.23.19).

[8] Machine Learning [WWW Document], n.d. . Coursera. URL https://ja.coursera.org/learn/machine-learning (accessed 3.23.19).

[9] Maplesoft - Software for Mathematics, Online Learning, Engineering [WWW Document], n.d. URL https://www.maplesoft.com/ (accessed 3.24.19).

[10] Meyer, B., 1988. Object-Oriented Software Construction, 1st ed. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[11] Moodle - Open-source learning platform | Moodle.org [WWW Document], n.d. URL https://moodle.org/ (accessed 3.24.19).

[12] Project Jupyter [WWW Document], n.d. URL https://www.jupyter.org (accessed 3.13.19).

[13] Rodríguez-del-Pino, J.C., Rubio-Royo, E., Hernández-Figueroa, Z.J., 2012. A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features.

[14] Seya, K., Shirasaka, S., 2016. Building Robots and Story-Based Learning to Teach Cyber-Physical Systems Effectively to Any Class of Audiences ranging from

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)

Elementary School Students to University Graduates. INCOSE International Symposium 26, 124–136. https://doi.org/10.1002/j.2334-5837.2016.00319.x
[15] Seya, K., Kobayashi, N., Shirasaka, S.: Effectiveness of Story-based Visual and Agile Teaching Method for Non-technical Adult Learners who want to understand Artificial Intelligence, 4th International Conference on Business Management of Technology (BMOT 2019)
[16] Wolfram Mathematica: Modern Technical Computing [WWW Document], n.d. URL https://www.wolfram.com/mathematica/ (accessed 3.23.19).

ISSN: 2304-1013 (Online); 2304-1269 (CDROM); 2414-6722 (Print)