Can Distributed Practice Improve Students' Efficacy in Learning their First Programming Language?

Qiujie Zhang[b], Lishan Zhang[a,b*], Baoping Li[a,b], Ling Chen[a,b], I-Han Hsiao[c] & Fati Wu[b]
[a]*Beijing Advanced Innovation Center for Future Education, Beijing Normal University, China*
[b]*Faculty of Education, Beijing Normal University, China*
[c]*School of Computing, Informatics & Decision System Engineering, Arizona State University, USA*
*lishan@bnu.edu.cn

Abstract: Learning how to program has become required for many majors in higher education. However, programming is not easily learned, especially for non-engineering students. To improve students' learning efficiency, we applied distributed practice to a C programming class with 69 college students in first grade, but have students decide the space of practice by themselves. By mining the relationships between practice patterns and learning performance, we found that students who practiced with high frequency significantly outperformed those who practiced infrequently. The frequency of practice was a strong predictor of both homework grades (p=0.001) and midterm exam grades (p=0.023). By contrast, the total amount of practice had very little effect on learning performance. The result shows that distributed practice is a better learning strategy than massed practice in C programming language learning. But the optimized space of practice in this domain has not been completely revealed yet.

Keywords: distributed practice, massed practice, programming language learning, higher education

1. Introduction

Mastering a programming language has become a required skill for many students in higher education, because programming now is an essential skill for data analysis in majors like economics, chemistry, biology and the social sciences. However, programming is not easily learned, especially for the students whose majors are non-engineering. These students usually only have programming related courses once a week, and do not have enough stimulus to practice. Therefore, if students can be encouraged to practice between the weekly classes, they may learn more efficiently.

Distributed practice is widely used in memorization (Dempster 1988; Rohrer 2015). For this task, the optimal space of learning or recalling is around one month. In the domain of language learning and reading comprehension, the optimal space tends to be reduced to one week or even shorter (Glenberg 1976; Balota, Duchek, & Paullin 1989; Begg & Green 1988). Previous studies seldom applied distributed practice in the domains like mathematics that were full of procedural knowledge (Cepeda, Pashler, Vul, Wixted, & Rohrer 2006), because space seems not always to have a significantly positive effect on learning procedural knowledge. Is distributed practice still useful in improving programming language learning efficiency? This paper studies whether distributed practice can help students learn C programming language, a typical domain full of procedural knowledge.

A previous study (Alzaid, Trivedi, & Hsiao, In press) introduced a system called "Quizit" which provided one multiple choice question for students to practice each day. Its preliminary result showed that studying actively with the system could lead to higher learning performance, but it was not clear whether it was due to the higher volume of practice, appropriate practice pattern or both. To further understand the effect of distributed practice on programming language learning, and also to test whether this bite-sized distributed practice could also help Chinese students, we adopted Quizit in a C programming class which had 69 Chinese college students in their first grade. Moreover, we analyzed the relationship between students' system usage pattern and their scores in midterm exams to answer the following research question: What factors in distributed practice can affect students' learning performance? By knowing the answer to this research question, we can teach students how to use the system more efficiently in the future to improve their learning.

2. Experiment Design and Analysis Model

Quizit is a system that provides students a multiple choice question to practice every day. The user interface is very simple and straightforward. As soon as a student logs into the system, the question of the day will show up if the student has not finished it. Students can always retry or review the question of the day at their convenience. Students may not log into the system every day, so they do not always finish their questions on time. In this case, they can make up their unanswered questions by accessing the calendar view and navigating to the previous dates. All the multiple choice questions in the system are posted by the programming language instructor in advance. The system sends out the questions based on the order enforced by the instructor. A student can make comments on each question, and the comments can be seen by himself/herself immediately. But a student cannot see the comments made by others until he/she makes his/her own.

Every interaction made by students is recorded by the system, and each interaction is associated with a timestamp. The recorded interactions for the follow-up analysis include: correct attempt, incorrect attempt, question review (only look at the question and its correct answer), question retry (re-answer the question), comment on a question.

We conducted the experiment in a classroom scale. The class was C programming language learning, and had 69 college students in their first year, majoring in Psychology. According to power analysis (Faul, Erdfelder, Buchner, & Lang 2009), our sample size (N=69) is big enough (Power=0.849) to find out a correlation whose coefficient is 0.35 with two-tailed significance (p=0.05). The experiment lasted 41 days, which entailed that students were required to answer 41 questions in total. Although students had one question to practice per day, they attended the class only once a week. The class lasted 210 minutes every time, with 10 minutes break each 45 minutes. This was the very first class in computer programming for most of the students. Students were required to use the system along with class study, but the instructor did not enforce how often they should practice with the questions.

To mine the relationship between students' system usage pattern and their learning performance, we designed nine indicators that represented different system usage behaviors and took the scores of four homework assignments as well as the midterm to quantify learning performance. With these indicators, we used Pearson correlation to find the relationships. The indicators are introduced rest of this section, and the analysis result is reported in the next section.

Because students can re-access their already answered questions, we calculated the measures to describe how they reviewed the questions. Specifically, two indicators were calculated for each student: the ratio of reviews, and the ratio of retries. Because retry is presumably more constructive than review, we expected that a higher number of retries may lead to better performance. We also explored students' levels of system usage from two different dimensions: the aggregated system usages and system usage frequency. The total number of questions a student answered was used to approximate the aggregated system usages of the student. On the other hand, we used a counter to record how many days a student practiced with the system. Every time a student logged into the system and answered a question, the counter was increased by 1. The counter could only increase at most by 1 in a single day, no matter how many questions the student practiced. The interval of system usage for each student then could be calculated as the total number of days divided by the counter. The lower the interval of system usage was, the higher the system usage frequency was. In order to know in which situations students prefer to make comments, especially comments relevant to the question, we calculated five related indicators for each question, which included difficulty, total number of comments, number of meaningless comments, number of irrelevant comments, and number of relevant comments. Difficulty was calculated as [1 – percentage of correct attempts] with the range between 0 and 1, and it did not matter if the attempts were the first or not. If a question was never answered correctly by anyone, its difficulty was 1. If a question was always answered correctly, its difficulty was 0. The other four indicators had their meaning labeled explicitly with their names. In addition, students' performance using the system was quantified with the percentage of first attempt correctness and the overall percentage of correct attempts. The percentage of first attempt correctness represented a student's base competence level regarding the related topic. By contrast, the overall percentage of correct attempts also took students' performance in retrial into consideration.

3. Results

Besides the data collected from the system, we also had four homework grades and the midterm score for each student. These five scores were used to measure a student's learning outcome. In this section, a descriptive result of the indicators is first reported, then the factors that affected learning performance are examined, and at last how a question's features affected the comments elicited by it is explored.

3.1 Descriptive Results

In terms of treating already answered questions, most students like to review (93.87%) the questions, instead of retry (6.13%). Students did not log into the system to practice very often. On average, students used the system every 9.04 (SD=8.69) days. The mean value was heavily affected by 4 extreme students who never logged into the system at all. The medium interval of system usage was 6.83. This means most of the students practiced with the system at least once a week. The questions were not too easy or too difficult. The percentage of first check correctness was 0.601 (SD=0.165). The overall percentage of correctness was 0.691(SD=0.135). The 69 students only made 406 comments for all the 41 questions. Out of the 406 comments, there were 271 meaningless comments, 125 comments relevant to the questions, and 10 irrelevant comments. Students made many meaningless comments probably to check the comments made by others.

Table 1: The correlation between system usage indicators and learning outcomes (N=69)

Outcome Indicator	Homework 1	Homework 2	Homework 3	Homework 4	Mean score of the 4 homework	Midterm
Ratio of reviews	r=-0.219 p=0.071	r=0.116 p=0.343	r=0.189 p=0.12	r=0.067 p=0.582	r=0.087 p=0.478	r=0.21 p=0.083
Ratio of retries	r=-0.025 p=0.837	r=-0.085 p=0.488	r=0.113 p=0.357	r=0.094 p=0.444	r=0.035 p=0.775	r=0.042 p=0.732
Total number of attempts	r=-0.182 p=0.134	r=0.234 p=0.053	r=0.233 p=0.053	r=0.017 p=0.888	r=0.153 p=0.208	r=0.135 p=0.268
Interval of system usage	r=0.046 p=0.71	r=-0.205 p=0.091	**r=-0.332**** **p=0.005**	**r=-0.247*** **p=0.041**	**r=-0.303*** **p=0.011**	**r=-0.273*** **p=0.023**
Percentage of correct attempts	r=0.011 p=0.927	r=0.011 p=0.931	r=-0.071 p=0.564	r=0.056 p=0.648	r=-0.004 p=0.972	r=0.109 p=0.372
Percentage of first attempts being correct	r=0.078 p=0.522	r=0.003 p=0.98	r=0.035 p=0.778	r=0.149 p=0.223	r=0.090 p=0.461	r=0.156 p=0.201
Ratio of meaningless comments	r=-0.012 p=0.92	r=-0.074 p=0.544	r=-0.186 p=0.126	r=-0.142 p=0.244	r=-0.165 p=0.176	r=0.133 p=0.275
Ratio of irrelevant comments	r=0.052 p=0.673	r=-0.125 p=0.306	r=0.065 p=0.594	r=0.04 p=0.742	r=0.003 p=0.979	r=-0.104 p=0.397
Ratio of relevant comments	r=-0.004 p=0.974	r=-0.138 p=0.259	r=0.104 p=0.397	r=0.113 p=0.357	r=0.021 p=0.866	r=0.18 p=0.14

3.2 Which Factors Affected Learning?

A student's learning performance was evaluated from two aspects: homework grades and midterm scores. One student had four homework grades before taking the midterm exam. The average score of the four homework grades for each student was calculated. It ended up with six performance measures in total. Out of the six measures, a mean score of the four homework grades and the midterm score were considered to be the two most important measures. Note that students could help each other while doing homework but they had to work on their own while taking the midterm exam.

We calculated the Pearson correlation coefficient between the indicators and the six outcome scores respectively. The result is described in Table 1. It showed that all but one indicator had no correlation with the performance outcome. Interval of system usage was the only exception. It did not correlate to the grades of the first two homework assignments, but strongly correlated to all the other performance outcome measures. Considering that early homework just had students review basic concepts, the interval of system usage was believed to be an important factor affecting learning performance. Neither the way to re-access the answered questions nor distribution of different types of comments had an effect on students' learning performance.

It is also interesting to see how the two different dimensions (i.e. close-book & close-discussion vs. open-book & open-discussion) of learning performance measures are correlated. The result of Pearson correlation is described in Table 2. Surprisingly, the scores of Homework 4 was the only indicator that was significantly correlated to midterm scores.

Table 2: The correlation between homework scores and midterm scores (N=69)

	Homework 1	Homework 2	Homework 3	Homework 4	Mean score of the 4 homework
Midterm scores	r=0.147 p=0.227	r=0.071 p=0.56	r=0.117 p=0.337	**r=0.321**** **p=0.007**	r=0.235 p=0.052

3.3 When do Students Want to Make Comments?

Pearson correlation coefficient was calculated between question difficulty and the number of their three different types of corresponding comments respectively. The correlation between question difficulty and the total number of corresponding comments was also calculated. The result is summarized in Table 3. The number of meaningless comments and total number of comments significantly correlated to question difficulty. At the same time, students were also actively making significantly more relevant comments to ask for help or share their thoughts. It appeared that students tended to refer to the comments for help when they had difficulty in answering questions correctly.

Table 3: The correlation between question difficulty and different types of comments (N=41)

	Number of meaningless comments	Number of relevant comments	Number of irrelevant comments	Total number of comments

Question difficulty	**r=0.401****
	p=0.009
		r=0.355*
		p=0.023
			r=0.107
			p=0.504
				r=0.467**
				p=0.002

4. Discussion

The result reveals several interesting findings. In terms of the learning factors, we found that neither number of practices, which was represented by total number of attempts, nor percentage of correct attempts was significantly related to learning outcomes. This seems inconsistent with the well-known Learning Factors Analysis (LFA) theory (Cen, Koedinger, & Junker 2006), which quantifies learning outcome with the number of correct attempts and incorrect attempts, and claims that learning outcomes should be improved with practice. However, different from the traditional setting of LFA, which provided more than enough practice questions, students in our experiment had a fixed number

(41) of questions and all the students answered the 41 questions at least once due to the requirement from the instructor. Students could practice with the same question as many times as they wanted, which lad to the difference in the number of attempts. This experiment setting provides us an opportunity to see how to better use the fixed size of practice questions for learning. From the result, it did not matter how a student reviewed the questions (just look at it or do it again). This was probably because all the questions were multiple choice questions. We suspected that type of review might play a more important role for the questions that required student to do actual programming. As mentioned above, the number of practices also did not matter. The only significant factor was the interval of practice. The lower the interval was, the better the learning performance was. This again seems inconsistent with the previous conclusions about spacing effect and distributed practice (Cepeda et al. 2009; Cepeda, Vul, Rohrer, Wixted, & Pashler 2008). But it is not the case. Traditional spacing effect experiments fixed the number of opportunities that one could practice and gave participants the same amount of practice every time. But we provided the fixed size of practice resources for each student, and let students decide the distribution of their practice. If a student's interval of practice was one day, he/she practiced one question per day, which was actually adding one-day space between each two questions. If a student's interval of practice was seven days, he/she would practice seven questions at a time. So the student added no space between the seven questions, but a seven -day space between every seven questions. The result was consistent with the finding in comparing massed practice to distributed practice (Dempster 1988; Hopkins, Lyle, Hieb, & Ralston 2016).

We had face-to-face interviews with six students after the midterm exam to understand what prevented some students from doing exercises in the system frequently, and why distributed practice helped some of them. The six students covered four different types: frequent system usage with high midterm score, infrequent system usage with high midterm score, frequent system usage with low midterm score, and infrequent system usage with low midterm score. All the students claimed that time was the biggest issue that prevented them from practicing the exercises every day. They were busy with many other classes. If they did have time, all but one student would like to practice every other day. The lowest-performing student simply felt bothered logging into the system and answering questions. It was not a surprise that the student got a very low score on the midterm. The reason why students did not want to practice every day was due to the cost of mental context switching (Kieffaber & Hetrick 2005). They often needed to prepare themselves for answering the questions, especially for the questions containing programs. They said that it usually just took three to five minutes to answer one question, so it was not worthwhile to answer just one question after the students spent an equally long time to prepare themselves. On the other hand, because answering one or two questions should only take less than 10 minutes, students should be able to spare this amount of time if they are somehow encouraged to do so. How to encourage students is one of our next projects. We could not explain well why some students still got a low grade on the midterm although they practiced frequently. It could be because practicing exercises in the system was the only thing they did for studying in this class. Indeed, there were many other variables that affected their learning outcome that we could not capture. Also, the experiment was just enough to claim that relatively shorter interval of practice could lead to better learning outcomes in C programming language learning. But we cannot conclude the optimized space of distributed practice for this course work.

We were expecting students who made more relevant comments would achieve higher learning performance, but this was not the case. It was probably because the comment function was not well introduced and recognized by students. After all, there were only 125 relevant comments in total. Students used this function only when they met difficulties. As the class goes on, questions will become more and more challenging, so we are expecting to see more comments in the future. For initiating students' discussions around the questions, we also considered compelling students to make comments.

Students spent less time answering simple questions and also did not have enough stimulus to post comments. By contrast, student had to spend a longer time dealing with complex questions, including answering and posting comments. This finding provides us an adaptive question recommendation mechanism. Based on an individual's time schedule, the system could choose to post a simple question for a busy day, and post a complex question for a leisure day. The utility of a question can be calculated in terms of time schedule and question difficulty, and the utility value can be used for the question recommendation (Zhang & Vanlehn 2016).

5. Conclusion

By having students decide the space of practice by themselves, we found that when all the students had the same amount of questions to practice, the students who practice frequently significantly outperformed those who practice infrequently in both homework and midterm exam. Neither the number of times re-practicing nor the way of re-visiting the answered questions affected students' learning performance. Many students claimed that they were short of time in practicing frequently, but the fact was that solving one simple multiple choice question usually only took less than five minutes. Complex questions usually need longer time to be solved, and students also tend to make more comments about these questions. So we should not expect students to spend equal amounts of time on all of the questions. In future work, we will consider how to encourage students and provide adaptive question recommendations based on students' time schedules.

Acknowledgments

This work was supported by The Chinese Ministry of Education Humanities and Social Science Foundation under projects number 14YJC880025, and China Postdoctoral Science Foundation under projects number 2017M610054.

References

Alzaid, M., Trivedi, D., & Hsiao, I. The Effects of Bite-size Distributed Practices for Programming Novices. Paper presented in *Frontiers in Education Conference* (FIE), 2017

Balota, D. A., Duchek, J. M., & Paullin, R. (1989). Age-related differences in the impact of spacing, lag, and retention interval. *Psychology & Aging*, 4(1), 3

Begg, I., & Green, C. (1988). Repetition and trace interaction: Superadditivity. *Memory & Cognition*, 16(3), 232-242

Cen, H., Koedinger, K., & Junker, B. (2006). *Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement*: Springer Berlin Heidelberg.

Cepeda, N. J., Coburn, N., Rohrer, D., Wixted, J. T., Mozer, M. C.,... Pashler, H. (2009). Optimizing distributed practice: Theoretical analysis and practical implications. *Experimental Psychology*, 56(4), 236-246

Cepeda, N. J., Vul, E., Rohrer, D., Wixted, J. T., & Pashler, H. (2008). Spacing effects in learning a temporal ridgeline of optimal retention. *Psychological science*, 19(11), 1095-1102

Dempster, F. N. (1988). The spacing effect: A case study in the failure to apply the results of psychological research. *American Psychologist*, 43(43), 627-634

Faul, F., Erdfelder, E., Buchner, A., & Lang, A. G. (2009). *Statistical power analyses using G*Power 3.1: Tests for correlation and regression analyses*: Springer-Verlag.

Glenberg, A. M. (1976). Monotonic and nonmonotonic lag effects in paired-associate and recognition memory paradigms. *Journal of Verbal Learning & Verbal Behavior*, 15(1), 1-16

Hopkins, R. F., Lyle, K. B., Hieb, J. L., & Ralston, P. A. S. (2016). Spaced Retrieval Practice Increases College Students' Short- and Long-Term Retention of Mathematics Knowledge. *Educational Psychology Review*, 28(4), 853-873

Kieffaber, P. D., & Hetrick, W. P. (2005). Event-related potential correlates of task switching and switch costs. *Psychophysiology*, 42(1), 56-71

Rohrer, D. (2015). Student Instruction Should Be Distributed Over Long Time Periods. *Educational Psychology Review*, 27(4), 635-643

Zhang, L., & Vanlehn, K. (2016). Adaptively selecting biology questions generated from a semantic network. *Interactive Learning Environments*, 1-19