文章编号：1672-5913(2018)12-0028-08 中图分类号：G642

DOI:10.16512/j.cnki.jsjjy.2018.12.005

Evaluating students' learning situations using "Four–quadrant law"

Jinze Liu, Dongyang Hu, Xunhui Zhang, Tao Wang, Yue Yu, Gang Yin

Abstract: Evaluation is widely acknowledged as a powerful means of improving the quality of education and it is a very important component of the education system. However, the current mainstream evaluation method is based on the examination as the common way, which has very limited help to instructors' teaching. In this paper, based on an online learning platform named "educoder", we design an evaluation method by using "Four-quadrant law", which divides students' learning situations into four types. In addition, we provide some suggestions for instructors about how to give targeted teaching to different students in each type. We verify that our method is useful by setting up experiments. The experiment results show that our method can effectively improve the quality of instructors' teaching.

Key words: evaluation; four-quadrant law; classification

1 Introduction

The traditional mode of education is being questioned in many aspects. More and more people think that instructor should focus on different teaching emphasis on different students. They think that the instructors' teaching arrangements should be dynamic adjustment according to the current learning situations of students, which can make the teaching activities more effective and targeted.

In order to realize this idea, instructors need to use evaluation to know the students' current learning situations, evaluation is widely acknowledged as a powerful means of improving the quality of education. Evaluation is a very important component of the education system, after the instructors have been teaching at the current stage, they need to get feedback from the students on their learning situations at that stage so that the instructors can make corresponding decisions. This kind of feedback is an evaluation of students' learning situations.

As for the evaluation of students' learning situations, the current mainstream method is based on the examination as the common way. By preparing test questions, instructors use the ranking of test scores as the result of evaluation. However, as time goes by, instructors find that this evaluation method has very limited help to their teaching. On the one hand, preparing test sheets and marking papers are so tedious that instructors cannot afford to have too many exams arranged. But if there are too few exams, the effect of the evaluation will be hard to guarantee. On the other hand, the ranking of test scores only reflects the order of students, for those students who ranked lower, instructors have no way to know why they cannot achieve good results, so that instructors cannot give

• Jinze Liu, Dongyang Hu, Xunhui Zhang, Tao Wang, Yue Yu and Gang Yin are with the College of Computer, National University of Defense Technology, Changsha 410073, China. E-mail: jinze_liu@qq.com.

• Manuscript received: 2018-01-15; revised: 2018-04-04; accepted: 2018-04-06.

targeted teaching on their deficiencies.

Therefore, we propose to apply the idea of "Four-quadrant law" to the evaluation of students' learning situations, which makes the result of evaluation no longer a simple ranking, but a classification of students' learning situations. After that, instructors can give more targeted teaching to different students according to the type of students.

This paper is conducted throughout an online learning platform named "educoder", which contains a large number of tasks, covering a very wide range of types, such as Java programming, Python programming, MySQL database, etc. Each task has several levels and each level has a question of programming for students to solve. On this platform, instructors can create a course and invite students to join, then the instructors can arrange the tasks on the platform according to their teaching arrangement for the students to complete. The design concept of "educoder" shows in Fig. 1 and Table 1.

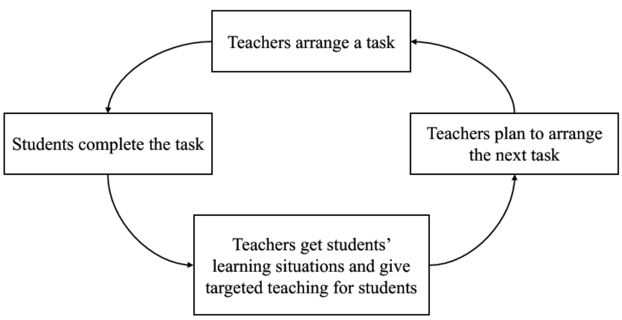


Fig. 1 Design concept of "educoder".

Table 1 Design concept of "educoder".

1.	Instructors arrange a task
2.	Students complete the task
3.	Instructors get students' learning situations and give targeted teaching for students
4.	Instructors plan to arrange the next task

As pass the 4-step procedure in Table 1, we build a loop which can promote the communication of instructors and students about teaching and learning. This paper addresses the issue of how instructors acquire students' learning situations in step 3.

In this paper, we first propose two classification indicators to characterize the learning situations of students. Second, we build a classification model of the learning situations of students, which divides students' learning situations into four categories according to the classification indicators and draws a classification diagram. At last, we separately select the most recent current task in different courses to evaluate students' learning situations. After Feeding back the evaluation results to the corresponding instructor, we track the students in these courses during the next task. By comparing these two learning situations, we verify that our method can make instructors know more about the learning situations of the students, so that instructors can achieve good effect on the targeted teaching of the students.

The key contributions of this study include the following:

• To the best of our knowledge, we apply the idea of "Four-quadrant law" to the evaluation of students' learning situations for the first time, which improves the role of the existing test-based evaluation methods in helping instructors.

• Making instructors know more about the learning situations of the students and helping instructors to give targeted teaching for students, which greatly improves the quality of instructors' teaching.

The rest of this paper is organized as follows. Section 2 introduces the research status of students' evaluation and the concept of "Four-quadrant law". Section 3 elucidates the approach of our study. Section 4 elaborates our experimental process and results. Section 5 concludes this paper and introduces future work.

2 Related Work

2.1 Evaluation of students

Evaluation of students has always been a subject of concern to many researchers in education. Study [1] outlines the principles for evaluating students' abilities and points out that the evaluation of student ability is an important means for educators to understand students. Study [2] introduces a self-evaluation of engineering education and it is points out that one of the key points of the evaluation is that the evaluation result should have an impact on the follow-up behavior of the evaluated person.

Study [3] uses statistical mixed-model methodologies to conduct multivariate, longitudinal analyses of student achievement to estimate the impact of various aspects of instructors' teaching methods on student achievement. The results show that the teaching model of instructors is the dominant factor affecting students' academic performance. Instructors' use of an effective teaching model is crucial to students' good performance.

Study [4] uses meta-analytic methodology to synthesize research on the relationship between student ratings of instruction and student achievement. Data comes from 41 independent validation studies and 68 standalone multi-sectoral courses that link student ratings with student grades. This shows that the traditional instructors' evaluation of student scores on the way to help students has little help. Instructors should look for better-performing methods of evaluation.

2.2 "Four-quadrant law"

"Four-quadrant law" is a time-management theory put forward by a American management expert named the Keynesian [5], which divides jobs into four types according to importance and urgency:

- Important and urgent.
- Important but not urgent.
- Urgent but not important.
- Not important and not urgent.

We can know more clearly about which type a specific job belongs to and how it should be done by using "Four-quadrant law". Therefore, we envisage that if the students are also classified according to the idea of the "Four-quadrant law", whether it will help instructors in teaching.

3 Approach

The goal of our work is providing a classification of students' learning situations in a task for instructors. As shown in Fig. 2, we first extract the classification indicators that characterize the learning situations of students. Then we build the classification model according to the classification indicators. Finally, we draw the classification diagram. In the following sections, we will elaborate each step in detail.

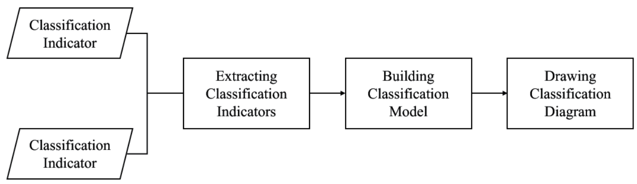


Fig. 2 Overall framework of our method.

3.1 Extracting classification indicators

Our work propose to describe students' learning situations in a task by students' programming ability and learning efficiency.

3.1.1 Programming ability

During completing the level in the task, students have to write code to achieve the requirements asked by the level and our platform will determine whether the code written by the students is correct. Students will get the level's scores and access into the next level until they submitted the correct code. During this time, students can submit their code for several times. The number of increased rows that each committed code compared with last version is called "code-increased-amount" and the number of deleted rows that each committed code compared with last version is called "code-deleted-amount":

$$\begin{aligned} Code\_inc_i &= \sum_{j=1}^{n} Commit\_inc_j \\ Code\_del_i &= \sum_{j=1}^{n} Commit\_del_i \end{aligned} \tag{1}$$

Where n is the total number of submission.

The number of changed rows when students have completed a level is called "code-changed-amount" of this level:

$$\text{Code\_chg}_i = \text{Code\_inc}_i + \text{Code\_del}_i \quad (2)$$

Study [6] pointed out that the students who are more capable of programming tend to complete levels with few code changes. It is not hard to understand that if a student' s programming ability is strong, his code does not need to repeatedly change to complete the level, sometimes even just need submit once to successfully pass. However, students with a weak programming ability need to repeatedly modify their code in order to pass the level.

Therefore, we use the logarithm of the ratio of the total levels' scores earned by students to the total "code-changed-amount" of all levels to describe students' programming ability (*PA*) in this task:

$$PA = \log\left(\frac{\sum_{i=1}^{k} \text{Score}_i}{\sum_{i=1}^{k} \text{Code\_chg}_i} + 0.01\right) \quad (3)$$

is the score of level and it is zero if students do not pass this level. is the total number of levels in this task. Plus 0.01 in order to prevent mathematical error.

There may be a situation where students who do not complete all the levels had the same *PA* as those who have passed all the levels. For example, when student-A passes all there levels with 30 "code-changed-amount" and get 300 points, student-B passes two levels with 20 "code-changed-amount" and get 200 points, they have the same *PA*.

This is obviously not in line with our expectation, and we should consider that students who have completed more levels have better programming ability. So we pull in a penalty factor of "code-changed-amount" in order to avoid this situation. We provide that if a student fails to complete a level, his "code-changed-amount" of this level is the maximum "code-changed-amount" of this level among all students in this course who have completed the level:

$$\text{Code\_chg\_punish}_i = \max\left(\text{Code\_chg}_i(t)\right) \quad (4)$$

3.1.2 Learning efficiency

We take the time difference between students starting a level and passing a level called "time-consuming-amount":

$$\text{Time\_con}_i = T_{\text{pass}}(i) - T_{\text{open}}(i) \quad (5)$$

Study [7] showed that a good learning attitude is the key to improving academic performance and a good learning attitude is reflected in the high learning efficiency. We have surveyed several courses' students and find that some of them lack perseverance in completing tasks. They are very easy to give up their mission to do other things, resulting in their high "time-consuming-amount". Therefore, we use the logarithm of the ratio of the total levels' scores earned by students to the total "time-consuming-amount" of all levels to describe students' learning efficiency (*LE*) in this task:

$$LE = \log\left(\frac{\sum_{i=1}^{k} \text{Score}_i}{\sum_{i=1}^{k} \text{Time\_con}_i} + 0.01\right) \quad (6)$$

Similar to the penalty factor of "code-changed-amount", we also pull in a penalty factor of "time-consuming-amount":

$$\text{Time\_con\_punish}_i = \max\left(\text{Time\_con}_i(t)\right) \quad (7)$$

3.2 Building classification model

For each task, instructors set the expected amount of code changes and completion time based on the difficulty of the task and the teaching experience of instructors, and we convert them to the threshold of *PA* (PA_{Th}) and the threshold of *LE* (LE_{Th}). Then for a course with students, we implement a threshold-based normalization of PA and LE of these students, so that the parts above or equal to the thresholds fall between[0,1] and the parts below the thresholds fall between[−1,0]:

$$Nor(PA(t)) = \begin{cases} \dfrac{PA(t) - PA_{Th}}{\max(PA) - PA_{Th}} & PA(t) \geqslant PA_{Th} \\ \dfrac{PA(t) - \min(PA)}{PA_{Th} - \min(PA)} - 1 & PA(t) < PA_{Th} \end{cases}$$
$$Nor(LE(t)) = \begin{cases} \dfrac{LE(t) - LE_{Th}}{\max(LE) - LE_{Th}} & LE(t) \geqslant LE_{Th} \\ \dfrac{LE(t) - \min(LE)}{LE_{Th} - \min(LE)} - 1 & LE(t) < LE_{Th} \end{cases} \quad (8)$$

After normalization, all the *PA* and *LE* fall between [−1,1]and 0 is where the thresholds are. So we select 0 as a cut-off point between the strong programming ability and the weak programming ability, and the high learning efficiency and the low learning efficiency, which divides students' learning situations into four types.

Table 2 Classification of students' learning situations.

Type	Description
Type 1	Strong programming ability High learning efficiency
Type 2	Strong programming ability Low learning efficiency
Type 3	Weak programming ability High learning efficiency
Type 4	Weak programming ability Low learning efficiency

Table 3 describes some suggestions for instructors about how to give targeted teaching to different students in each type.

Table 3 Suggestions for instructors about targeted teaching.

Type	Suggestion
Type 1	Such students are very outstanding, suggest that instructors set them as role models and call on other students to learn from them.
Type 2	Such students lack of correct learning attitude, concentration and perseverance in completion of task, which is very detrimental to their study. Suggest that instructors pay more attention to their learning attitudes and regularly urge them to conscientiously complete the task.
Type 3	Such students have a high enthusiasm of learning. Although they need to repeatedly modify their code to pass the level, they are actively modifying. As for programming ability, suggest that instructors can pay attention to whether they are not familiar with this programming language, or the requirements of the level is not enough understanding.
Type 4	Such students need more care, suggest that instructors should take into account the suggestions in Type2 and Type3.

3.3 Drawing classification diagram

We use *PA* as *X* axis and *LE* as *Y* axis. Then we use *X* axis and *Y* axis as the quadrant boundaries to draw classification diagram.

4 Experiment

4.1 Experimental setup

We conduct experiments in several courses, and select two courses with more students to introduce(as shown in Table 4).

Table 4 Introduction of experimental course.

Courses	Course ID	Number of students
Course 1	1105	91
Course 2	1139	90

The Task ID of the most recent current task for students in Course 1 is 65 and for students in Course 2 is 61. We evaluate the students' learning situations in these two tasks and calculate the proportion of each type of students in the total number of students:

$$P_{\text{Type }(n)} = \frac{N_{\text{Type }(n)}}{N} \tag{9}$$

Where N is total number of students.

We reflect the evaluation results to the corresponding instructors and then continue to track the next task in both courses and also evaluate their learning situations during the next task. Finally, we respectively compare the evaluation results in these two tasks.

4.2 Experimental result

Figure 3 shows the evaluation result of learning situations of students in Course 1 & Task ID: 65.

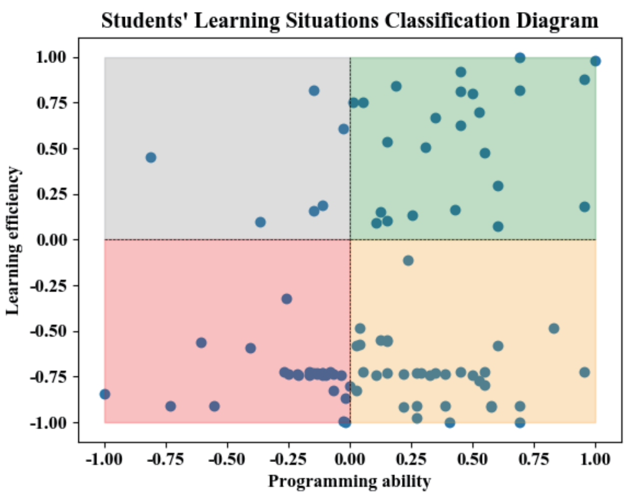


Fig. 3 Evaluation result in course 1 & task ID: 65.

In the figure, the green area represents students in Type 1, the yellow area represents students in Type 2, the grey area represents students in Type 3 and the red area represents students in Type 4.

Table 5 shows the proportion of each type students in Course 1 & Task ID: 65.

Table 5 Proportion of students in Course 1 & Task ID: 65.

Type	Proportion
Type 1	26.37%
Type 2	38.46%
Type 3	6.59%
Type 4	28.58%

Figure 4 shows the evaluation result of learning situations of students in Course 2 & Task ID: 61.

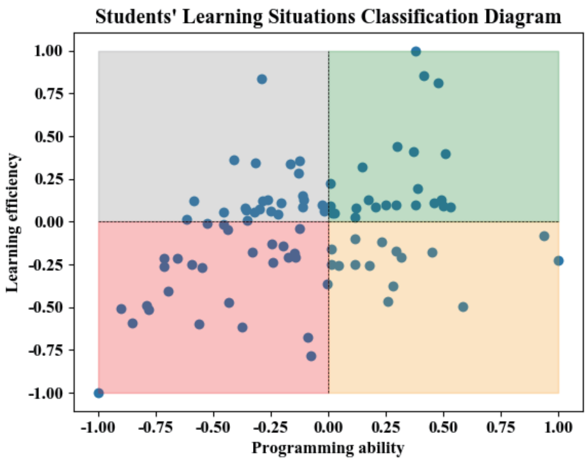


Fig. 4 Evaluation result in course 2 & task ID: 61.

Table 6 shows the proportion of each type students in Course 2 & Task ID: 61.

Table 6 Proportion of students in Course 2 & Task ID: 61.

Type	Proportion/%
Type 1	25.56
Type 2	16.67
Type 3	26.67
Type 4	31.10

The Task ID of the next task for students in Course 1 is 79 and for students in Course 2 is 85.

Figure 5 shows the evaluation result of learning situations of students in Course 1 & Task ID: 79.

Table 7 shows the proportion of each type students in Course 1 & Task ID: 79.

Figure 6 shows the evaluation result of learning situations of students in Course 2 & Task ID: 85.

Table 8 shows the proportion of each type students in Course 2 & Task ID: 85.

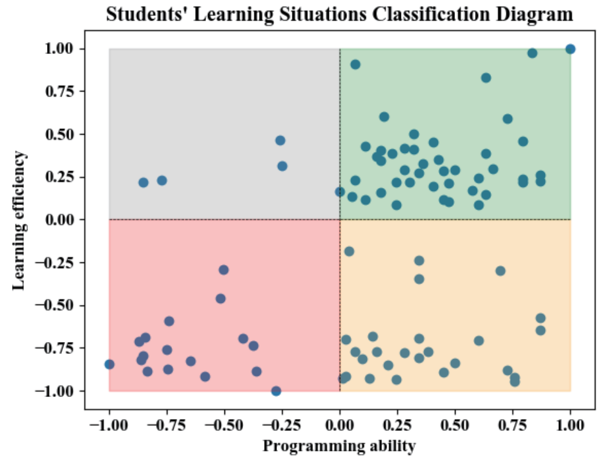


Fig. 5 Evaluation result in course 1 & T-ID: 79.

Table 7 Proportion of students in course 1 & task ID: 79.

Type	Proportion/%
Type 1	48.35
Type 2	28.57
Type 3	4.40
Type 4	18.68

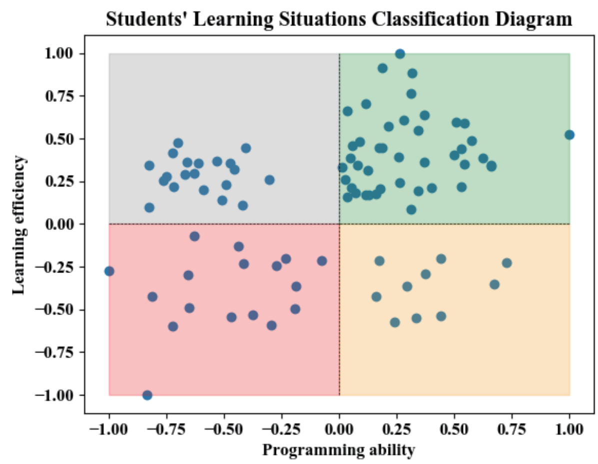


Fig. 6 Evaluation result in course 2 & task ID: 85.

Table 8 Proportion of students in course 2 & task ID: 85.

Type	Proportion/%
Type 1	47.78
Type 2	11.11
Type 3	22.22
Type 4	18.89

As shown in Table 5 and Table 7, in Course 1, the proportion of students of Type 1 increases 21.98 percent from the first task to the next. And the proportion of students of Type 2, Type 3 and Type 4 respectively decreases 9.89 percent, 2.19 percent and 9.90 percent.

As shown in Table 6 and Table 8, In Course 2, the proportion of students of Type 1 increases 22.22 percent from the first task to the next. And the proportion of students of Type 2, Type 3 and Type 4 respectively decreases 5.56 percent, 4.45 percent and 12.21 percent.

Therefore, the experiment results show that after instructors use our evaluation result to guide their teaching, the grades of students have been greatly improved, which verifies our approach can effectively improve the quality of instructors' teaching.

5 Conclusion & Future Work

In this paper, we first introduce the importance of evaluation to education and analyze why the current mainstream evaluation methods are difficult to meet the instructor's needs. Then we propose out own approach, which is applying the idea of "Four-quadrant law" to the evaluation and dividing students' learning situations into four types. Also, we provide some suggestions for instructors about how to give targeted teaching to different students in each type. At last, we verify our method can effectively improve the quality of instructors' teaching by experiment.

Our work is working on the off-line computer at the moment and in the future we plan to integrate this evaluation function into "educoder" to provide better service to all instructors using this online learning platform.

References

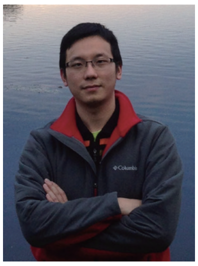
[1] Schrader D E. Assessing student learning and development: a guide to the principles, goals, and methods of determining college outcomes[J]. The Journal of Higher Education, 1992, 63(4): 463-465.
[2] Staniškis J K, Katiliūtė E. Complex evaluation of sustainability in engineering education: case & analysis[J]. Journal of Cleaner Production, 2016, 120: 13-20.
[3] Sanders W L, Wright S P, Horn S P. Teacher and classroom context effects on student achievement: implications for teacher evaluation[J]. Journal of Personnel Evaluation in Education, 1997, 11(1): 57-67.
[4] Cohen P A. Student ratings of instruction and student achievement: a meta-analysis of multisection validity studies[J]. Review of educational Research, 1981, 51(3): 281-309.
[5] Warr P, Bindl U K, Parker S K, et al. Four-quadrant investigation of job-related affects and behaviours[J]. European Journal of Work and Organizational Psychology, 2014, 23(3): 342-363.
[6] DeClue T H. Pair programming and pair trading: effects on learning and motivation in a CS2 course[J]. Journal of Computing Sciences in Colleges, 2003, 18(5): 49-56.
[7] Swing S R, Peterson P L. The relationship of student ability and small-group interaction to student achievement[J]. American Educational Research Journal, 1982, 19(2): 259-274.

Jinze Liu received his B.S. in communication engineering from Centre South University in 2017. He is now a M.S. candidate in computer science and technology, National University of Defense Technology. His work interests include open source software engineering, data mining, and social coding networks.

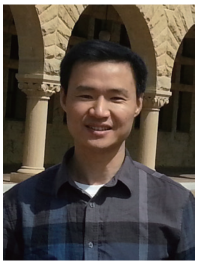
Dongyang Hu received his B.S. in information security from Shanghai Jiao Tong University, in 2017. He is now a M.S. candidate in computer science and technology, National University of Defense Technology. His work interests include open source software engineering and data mining.

Xunhui Zhang received his B.S. in Computer Science from Sichuan University in 2015. He is now a M.S. candidate in software engineering, National University of Defense Technology. His work interests include open source software engineering, data mining, and recommendation system.

Tao Wang received both his M.S. and Ph.D. degrees in computer science from National University of Defense Technology (NUDT), in 2010 and 2014, respectively. His work interests include open source software engineering, machine learning, data mining and knowledge discovering in open source software.



Yue Yu received his Ph.D. degree in computer science from National University of Defense Technology in 2016. His current research interests include software engineering, spanning from mining software repositories and analyzing social coding networks.



Gang Yin received both his M.S. and Ph.D. degrees in Computer Science from National University of Defense Technology. He is now a member of Software Engineering Committee of CCF. His work interests include distribution calculation, software engineering, and online education.

(Publishing Editor:Yiming Sun)