The Application of Event Studies in Evaluation of Teaching Quality

Xiaoguang Liu, Gang Wang, Jing Liu
Department of CS,Nankai University
Nankai University
TianJin, P.R.China
E-mail:{liuxg74,wgzwp}yahoo.com.cn,jingliu@nankai.edu.cn

Abstract—**Computer science is integrated with the practice closely. It has become consensus of scholars that coding ability is important to computer science education, and programming exercises can improve teaching quality efficiently. However, this consensus more relies on the expert's experience and intuition. It's lack of quantifiable indicators or empirical data to support the consensus. In this paper, event studies are used to analysis the effect of programming exercises to teaching quality. The results show that the improvement of students' coding ability can effectively improve teaching quality of computer science courses.**

Keywords-Event Study; Teaching quality; Computer Science

I. INTRODUCTION

With the development of IT, the requirements to IT engineer increases at the same time. There are some challenges to computer science education in universities[1]. Firstly, the increasing requirements from society provide many opportunities to computer science. Secondly, the knowledge of IT changes so quickly that the courses in universities may not fit for the requirements of business.

In order to solve the problem, some advices are presented by many scholars [1,2,3,4]. The main ideas of these advices can be describes as three items. Firstly, updating the teaching content according to the latest advances in computer science. At the same time, more references are provided to students to enhance their independent learning ability. Secondly, in order to improve students' coding ability, special experimental courses should be set up. Thirdly, cooperating with enterprises to provide some practice training to students. In recent years, many policies are issued, such as tax incentives, to encourage enterprises to provide more opportunities to students.

In this paper, event studies are used to analysis the effect of programming exercises to teaching quality. Event study is an econometric method to evaluate the effect of event by comparing the data before and after the event occurred [5]. In Nankai University, a special programming exercise course for C++ language was set up in 2004. It's used as event to evaluate the effect of teaching quality in this paper.

II. OUR WORK

Usually, C++ language is the first programming language which a computer science undergraduate student will meet in university. Students will learn how to program in this course. Coding ability is basic skill to computer science students. This ability is also foundation of other computer science courses. The teaching quality of C++ language will directly affect teaching quality of other professional courses.

At the same time, students will know what computer science is firstly by learning C++ language. The course is important to students' understanding and interests to computer science. The students will also establish good coding style and the correct learning method by programming exercises.

In Nankai University, in order to improve students' coding ability, programming exercises was set as a special course for C++ language in 2004. In this course, students are told what a good program is, coding regulars and debugging skills as far. The main idea of this course is students should know how to program by more practical exercises.

In this paper, we take this change as 'event' to evaluate the effects of coding exercise. Because C++ language is basic skill to computer science, the teaching quality will reflect more in other computer science courses, such as database, data structure and operation system. So we only select the courses, which are taught after C++ language, as the samples in tests. By paired comparison the students' scores before and after the event, we can evaluate the effect of this event.

III. HYPOTHESES AND PRETREATMENTS

As an empirical research, our work is inevitably subject to data collection, statistical methods or other conditions. Some hypothesizes are given as follows,

Hypothesis **1:** The students, who enter the university before and after this event, don't show significant difference on intelligence and learning ability.

In this paper, the samples are the students who enter department of computer science in Nankai University in 2003 and 2004. It's just before and after the event, which means programming exercises was set as a special course. Because students enter the universities every year are almost same, *Hypothesis* 1 can be accepted from the intuition.

Hypothesis **2:** Except the event, there is no other factors can affect teaching quality significantly.

This work is partly sponsored by NSF of China (No.90612001), 863 Program(No.2008AA01Z401), the Ph.D. Programs Foundation of Ministry of Education of China(No. 20070055054),Science and Technology Development Plan of Tianjin(No. 08JCYBJC13000).)

This is a strict hypothesis. In order to eliminate the impact of other factors as much as possible. We do some pretreatment to these samples.

Pretreatment 1: In sample data, only the computer science courses, which are almost same to students enter university in 2003 and 2004, are selected. It means the courses, which are selected, have same teachers and other teaching materials.

Hypothesis **3**: The mean examination results can be used as indicator of teaching quality.

It's also a strict hypothesis. In real life, the evaluation of teaching quality should include many factors, such as examination result, understanding of knowledge and practical application of knowledge etc. Due to the examination result is the only quantitative indicator we can get, it's used in this paper.

During the analysis of sample data, we find some extreme data have more impact on mean score. For example, an extreme excellent student can increase the mean score when the number of student in the class is small and vice versa. In order to eliminate the effect, we do Pretreatment 2 as following.

Pretreatment 2: In sample data, the best and worst 10% items are eliminated, and only other 80% items are conserved.

Pretreatment 2 is acceptable according to teaching experience.

Firstly, the best and worst students cannot be used as indicator of teaching quality.

Many teachers consider that the best students can learn by themselves and the worst students never learn. It means the best students have prominent intelligence and formed their own learning method usually. The teacher's mission is to direct and correct them only. To the worst students, the intelligence isn't key factor mostly because they can pass the strict university entrance examination. Their failure caused by some private reasons, such as learning attitude, family and friendship etc.

Further more, the number of the best and worst students every year shows some randomness. This randomness will weaken the confidence of results. So we only study the remaining 80% items as sample.

Secondly, the purpose of education is to improve the ability of most students. Using middle 80% items can reflect the teaching quality more efficiently.

According to Pretreatment 2, *Hypothesis* 3 can be modified as follows,

Hypothesis **3A:** The mean examination results of middle level students can be used as indicator of teaching quality.

In the following sections, all data are treated according to Pretreatment 1 and 2.

IV. STATISTICAL TESTS

Z test is used to study the effect event in this paper. The samples are the examination results of students who enter university before and after this event.

The implicit assumption of Z test is the data, which Z test studied, are independent with the same distribution random variables. This assumption is acceptable.

The null hypothesis of Z test,

H_0: There is no significant difference between the mean examination results of students in 2003 and 2004.

The alternative hypothesis,

H_1: There is significant difference between the mean examination results of students in 2003 and 2004.

According to the rules of *Hypothesis 1, 2, 3A and* Pretreatment 1, 2, samples are selected. There are four computer science courses in the samples. They are Compiler Design, Data Structure, Operation system and Computer System Architecture.

The sample data are shown in following figures. In these figures, Y-coordinate is examination results and X-coordinate is the number of items. Here, the items are sorted according to ascending order.

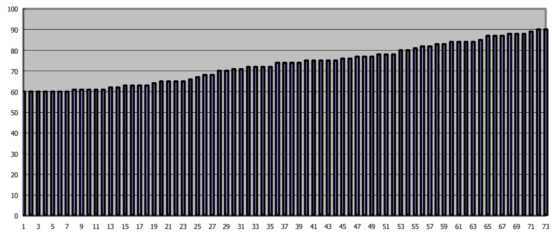

Figure 1. 2003 students' scores in Compiler Design

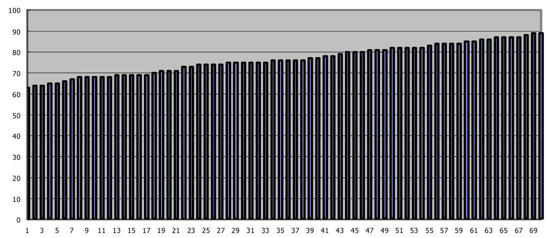

Figure 2. 2004 students' scores in Compiler Design

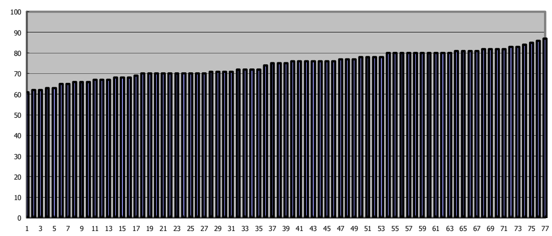

Figure 3. 2003 students' scores in Data Structure

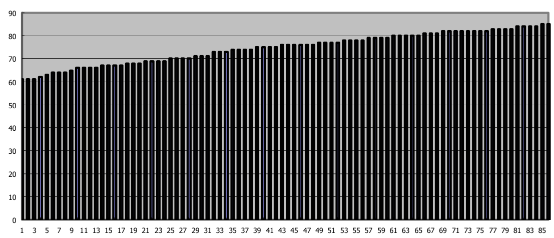

Figure 4. 2004 students' scores in Data Structure

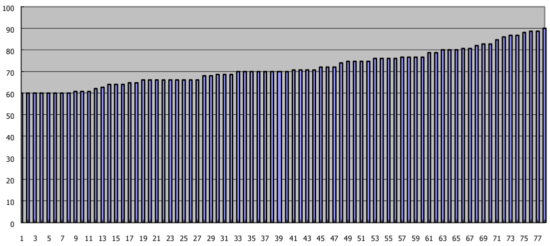

Figure 5. 2003 students' scores in Operation System

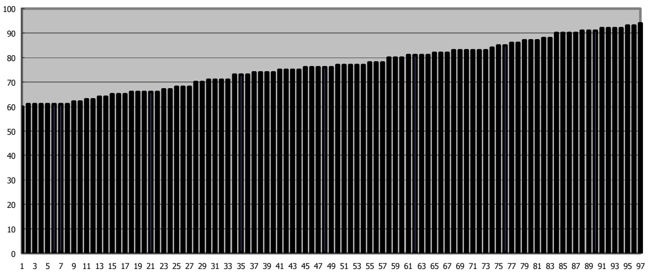

Figure 6. 2004 students' scores in Operation System

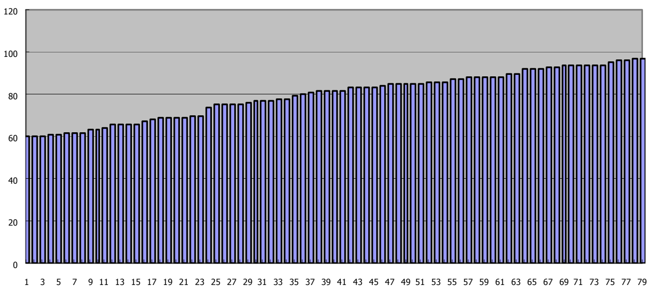

Figure 7. 2004 students' scores in Computer System Architecture

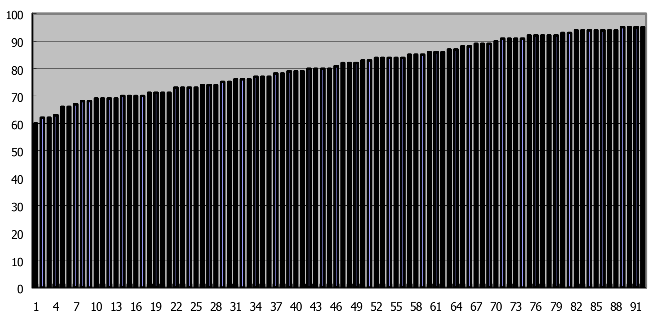

Figure 8. 2004 students' scores in Computer System Architecture

The equation of Z test is :

$$Z = \frac{\text{observed value - expected value}}{\text{Difference of SE}} = \frac{\text{mean score of 2004 - mean score of 2003}}{\text{Difference of SE}} \quad (1)$$

The test results are shown in table 1.

As shown in table 1, to all four courses, the mean scores of students in 2004 are higher than that of students in 2003. Specifically, the P value of Compiler Design is 1.69%. It means the probability of random error is 1.69% to Compiler Design course. Therefore, H_0 is rejected to Compiler Design course. Using the same method, H_0 is rejected to Operation System course in term of its P value is 0.07%. As the results, we can say the mean scores of students in 2004 are significant higher than students in 2003.

At the same time, the P value of Data Structure and is 76.42%. It means H_0 cannot be rejected, because the probability of random error is much higher. Although the mean scores of students in 2004 are higher than that of students in 2003 to Data Structure, we cannot determine the results are caused by random error whether or not. To Computer System Architecture, we get same conclusion in term of its P value is 48.39%.

V. CONCLUSIONS

In this paper, event studies are used to evaluate the teaching quality. According to the idea of *Hypothesis 1, 2 ,3A and after data pretreatment 1 and 2,* four computer science courses, Compiler Design, Data Structure, Operation System and Computer System Architecture, are selected as samples. The event in the test is the setup of special programming exercises course for C++ language in 2004. The test results show that the mean scores of students in 2004 are higher than that of students in 2003 to all four courses. Furthermore, the statistical results are significant to Data Structure and Operation System.

According to the results, we can conclude that the setup of special programming exercises course for C++ language has a positive impact on teaching quality of computer science courses.

The conclusion shows that a high-quality programming exercises course can not only help students develop good programming habit, but also be conducive to understand computer science correctly and establish good learning methods. All of these will help students better understand the knowledge of computer science.

We also find some interesting results in the tests. To some courses which don't belong to computer science, such as Cryptography and Game Theory, the mean examination results of students, who enter university before and after this event, don't show significant difference. These results may imply programming exercises course is only effective in computer science field.

TABLE I. THE RESULTS OF Z TEST

COURSES NAME	**STUDENTS IN 2003**			**STUDENTS IN 2004**			**RESULTS**	
	Mean	***SE***	***Number of Sample***	***Mean***	***SE***	***Number of Sample***	***Z***	***P***
Compiler Design	73.123	1.101	73	76.443	0.849	70	2.3869	1.69%
Data Structure	74.272	0.731	77	74.581	0.727	86	0.2993	76.42%
Operation System	71.821	0.939	78	76.474	0.993	97	3.4024	0.07%
Computer System Architecture	79.721	1.257	79	80.858	0.998	92	0.7080	48.39%

REFERENCES

[1] Judy Kay; Michael Barg,etc, Problem-Based Learning for Foundation Computer Science Courses, Computer Science Education, Vol. 10(2), 2000 , pp 109 – 128

[2] Said Hadjerrouit, A constructivist approach to object-oriented design and programming, ACM SIGCSE Bulletin, Vol.31(3), 1999,pp.171-174.

[3] Jim McKeown, The use of a multimedia lesson to increase novice programmers' understanding of programming array concepts, Journal of Computing Sciences in Colleges, Vol19(4),2004, pp39-50.

[4] Sarah Matzko , Timothy Davis, Pair design in undergraduate labs, Journal of Computing Sciences in Colleges, Vol.22(2),2006,pp123-130.

[5] Campbell,J., The Econometrics of Financial Markets, Princeton University Press, US, 1997.