

A heuristic method for performance evaluation of learning in an intelligent tutoring system

KIRAN MISHRA¹
R.B. MISHRA²

Banaras Hindu University
Department of Computer Engineering
Institute of Technology
Varanasi-221005- India

¹kiran.mishra.bhu@gmail.com

²ravibm@bhu.ac.in

Abstract. Evaluating student's performance is one of the important activities in intelligent tutoring system. This paper presents learning strategy for study and a heuristic method for performance evaluation of learning of student in intelligent tutoring system. A course has been divided into different interlinked courseware. The method is based on the number of attempts with and without hint for success in a courseware and student's performance in a courseware. Grading such as average, good, excellent has been assigned to the student based on his performance value index. A comparative view of different methods has been presented based on some characteristics such as: computing model, evaluation parameter and evaluation measure. A graphical view of comparison has also been presented.

Keywords: Intelligent tutoring system, performance evaluation, hint, attempt, performance index.

(Received November 17, 2009 / Accepted May 07, 2010)

1 Introduction

Many intelligent tutoring systems (ITS) have been developed for teaching of programming languages i.e. MENO-II [5], PROUST[9], LISP-Tutor [3], ELM-PE[17], C++ tutor [1] [10], JAVA tutor [15], [2]. These systems use different learning strategy. PROUST attempted to estimate student's plan intention while LISP tutor guided student's left to right, top down attempt by interpreting their code as a correct or buggy solution.

MENO-II and ELM-PE tutor analyze the student's solution to exercise and provide feedback to identify misconception or missing skills based on the analysis. [6] presented an expert tutoring system (E-TCL) for teaching computer programming languages through WWW. Their system consists of three agents representing server-client relationship, tutoring agent(TA) as a "server", personal assistant agent for teachers(PAA-T), and personal assistant agent for students (PAA-S) as

"clients". The PAA-S can communicate with the TA through the WWW to retrieve the tutoring dialog of the command(s) that a student wants to practice, and to access the experiences of other students in the blackboard module while the PAA-T communicates with the TA to add/modify semantic rules of computer programming languages and to check the correctness of the contents of the blackboard database.

[4] presented an ITS (Bits) for computer programming using Bayesian technology. Bits provided remote access to hypermedia-structured learning material which included instruction notes, tests, and examples. Unlike traditional web based education tools, Bits provided the learner with intelligent navigation support, recommendation, and integrated the features of an electronic hypermedia textbook with intelligent tutoring tactics. Bits proposed learning goals and guided users by generating reading sequences for them. [10] asks students to pre-

dict C++ program's output and identify semantic and runtime errors. [15] intelligently examines the student's submitted code and determines appropriate feedback based on a number of factors such as cognitive model of the student, the student's skill level, and problem details.

[13] described a tutoring system for learning computer programming based on a multiple domain multiple-agent environment. It provided a multiple domain learning environment for language learners. The aim of the system was to teach the target domain by using a supporting domain(s) to reinforce the learning of it. [2] developed an agent based intelligent tutoring system for parameter passing in Java programming. This system helps student better understanding of parameter passing mechanisms in Java using problem based technique. [1] provides intelligent feedback to the student and for this purpose it relies on a group of information: the problem statement, problem specification, student's code, established student's model, the C++ compilation and the result from C++ run time engine.

Different methods have been used for performance evaluation of student. [18] evaluated performance of student in four levels i.e. global assessment(an overall measurement of student's ability; procedure-level assessment(measurement for each problem the student is asked to solved); stage assessment(a measurement for each of three physiological stages in a problem); local assessment(a measurement for each variable that has been tutored). [16] presented an approach to hierarchical knowledge representation for the student's evaluation in propositional logic. The hierarchical evaluation consists in assessing the student's state of knowledge at several level of granularity.

[8] evaluated student's performance using evaluation module. The evaluation module evaluates the user's performance during a learning session, based on the interaction of the user with the system. Student's mark level ranges from 'low' to 'excellent' based on the number of times the user asked for assistance, the number of related example requested by the user and the number of answering attempts made by the user. This paper presents learning strategy for study and heuristic method for performance evaluation of learning of student in intelligent tutoring system. The method is based on the number of attempts for success in a courseware and student's performance in a courseware.

Rest of the paper has been organised as follows: Apart from introduction in section 1, section 2 deals with modeling the courseware. Section 3 represents learning strategy. Performance evaluation has been represented in section 4. Section 5 contains experimen-

tation. section 6 contains result, section 7 represents comparison and finally section 8 represents conclusion.

2 Modeling the courseware

We have been taken C++ programming language as subject model. The course has been divided for the whole curriculum for C++ into seven coursewares i.e. CW1-classes and objects, CW2-constructor, CW3-operator overloading, CW4-inheritance, CW5-virtual functions, CW6-managing console io operations, CW7-working with files. The curriculum has been divided into courseware in such a way that CW1 is prerequisite for CW2; CW1 and CW2 are prerequisite for CW3; CW1, CW2 and CW3 are prerequisite for CW4; CW1, CW2, CW3 and CW4 are prerequisite for CW5; CW1, CW2, CW3, CW4 and CW5 are prerequisite for CW6; and CW1, CW2, CW3, CW4, CW5, and CW6 are prerequisite for CW7. For example, classes and objects are necessary to understand constructor and classes and objects and constructor are necessary to understand operator overloading.

3 Learning strategy:

The flow chart for learning strategy has been shown in figure 1 which includes the following steps.

1. Student studies first courseware CW[1]
2. After study his performance is evaluated.
3. If his performance is up to level he can go to CW[2]. If performance is not up to level he has to repeat same courseware (he will get up to three attempts. In 2nd and 3rd attempt he will get hint related to question. If performance is up to level, he will go to next courseware. If performance is not up to level he will repeat same courseware. After three attempts if performance is not up to level he will get specific treatment)
4. In CW[2] his performance is evaluated.
5. If his performance is up to level he can go to CW[3]. If performance is not up to level he has to repeat same courseware (he will get up to three attempts. In 2nd and 3rd attempt he will get hint related to question. If performance is up to level, he will go to next courseware. If performance is not up to level he will repeat same courseware. After three attempts if performance is not up to level he will get specific treatment)
6. In CW[3] his performance is evaluated.

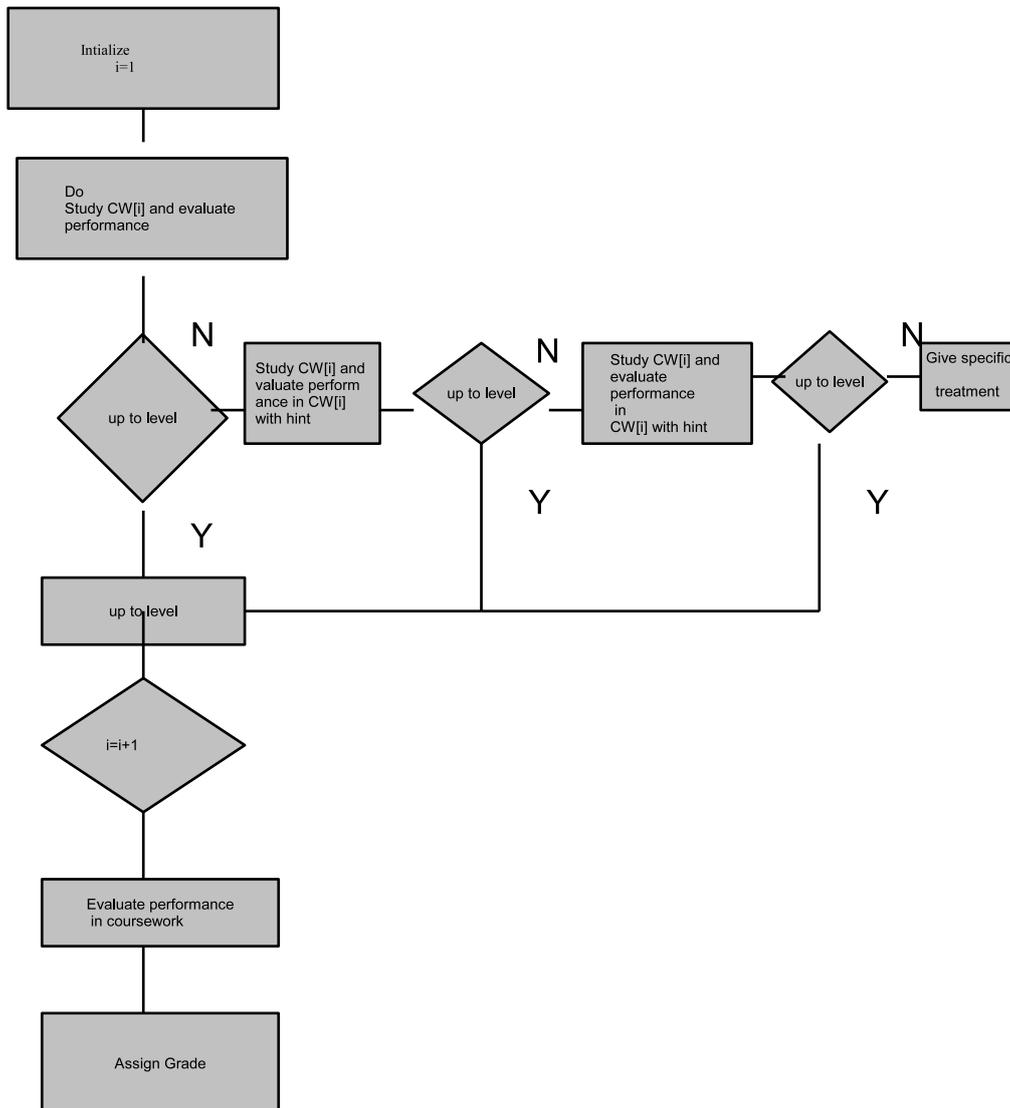


Figure 1: Flow chart for learning strategy.

7. If his performance is up to level he can go to CW[4]. If performance is not up to level he has to repeat same courseware (he will get up to three attempts. In 2nd and 3rd attempt he will get hint related to question. If performance is up to level, he will go to next courseware. If performance is not up to level he will repeat same courseware. After three attempts if performance is not up to level he will get specific treatment)
8. In CW[4] his performance is evaluated.
9. If his performance is up to level he can go to CW[5]. If performance is not up to level he has to repeat same courseware (he will get up to three attempts. In 2nd and 3rd attempt he will get hint related to question. If performance is up to level, he will go to next courseware. If performance is not up to level he will repeat same courseware. After three attempts if performance is not up to level he will get specific treatment)
10. In CW[5] his performance is evaluated.
11. If his performance is up to level he can go to CW[6]. If performance is not up to level he has to repeat same courseware (he will get up to three attempts. In 2nd and 3rd attempt he will get hint related to question. If performance is up to level, he will go to next courseware. If performance is not up to level he will repeat same courseware. After three attempts if performance is not up to level he will get specific treatment)

Table 1: Examination data of student.

Student	Performance in CW1 (A+R+D)	Performance in CW2 (A+R+D)	Performance in CW3 (A+R+D)	Performance in CW4 (A+R+D)	Performance in CW5 (A+R+D)	Performance in CW6 (A+R+D)	Performance in CW7 (A+R+D)	Performance Index(PI)	Grade
1	6(1st)	5(2nd)	4(1st)	3(1st)	5(3rd)	4(1st)	3(1st)	3.5	good
2	5(2nd)	6(1st)	3(1st)	4(1st)	5(2nd)	6(1st)	4(1st)	4.2	good
3	3(1st)	5(1st)	4(1st)	5(2nd)	3(1st)	4(1st)	4(1st)	3.7	good
4	5(1st)	4(1st)	5(1st)	6(2nd)	4(1st)	3(1st)	4(1st)	4.1	good
5	6(1st)	6(2nd)	6(2nd)	6(1st)	6(2nd)	6(1st)	6(2nd)	4.8	Excellent

he will repeat same courseware. After three attempts if performance is not up to level he will get specific treatment)

12. In CW[6] his performance is evaluated.

13. If his performance is up to level he can go to CW[7]. If performance is not up to level he has to repeat same courseware (he will get up to three attempts. In 2nd and 3rd attempt he will get hint related to question. If performance is up to level, he will go to next courseware. If performance is not up to level he will repeat same courseware. After three attempts if performance is not up to level he will get specific treatment)

14. Performance in CW[7] is evaluated. If performance is up to level his final score will be calculated based on the equation (1) given in following section. If performance is not up to level he will repeat CW[7] up to three attempts. After three attempts if performance is not up to level he will get specific treatment.

4 Performance Evaluation

In [11] We have made analytical (A), reasoning (R) and descriptive(D) type questions from each courseware for evaluating the performance of student. Although there has been more 4 divisions of the type of questions such as AR(Analytical-Reasoning) , RD (Reasoning - Descriptive), AD (Analytical - Descriptive), and ARD (Analytical - Reasoning - Descriptive).

Student's performance is up to level, if he gets greater or equal to 1(1 is threshold value) in analytical(A),and in reasoning(R) and in descriptive(D) and his average of marks in A,R and D is greater or equal to 1. If this condition is not met his performance is not up to level. Based on this performance and number of attempts performance index can be calculated using following formula i.e. equation(1).

$$PI = 1/q * \sum_{i=1}^7 Pi * W_n \quad (1)$$

In equation (1) PI is performance index.

q= number of courseware, here q=7 because there are seven courseware.

Pi is performance of the student in ith courseware. Performance is summation of marks obtained by student in analytical, reasoning and descriptive type questionnaire. For example if student got 2 in analytical, 2 in reasoning and 2 in descriptive in two attempts in a courseware, his performance will be 2+2+2=6 and Wn (weight of marks) will be 2/3.

Wn is the weight of marks obtained by student. It can be 1, 2/3 or 1/3. It depends on the number of attempts in which student's performance is upto level. If his performance is up to level in first attempt, Wn will be 1. If performance is up to level in 2nd attempt Wn will be 2/3. If his performance is up to level in three attempts Wn will be 1/3.

Based on the value of performance index "excellent", "good", and "average" grade have been given to the student. Following rules have been used for grading in the scale of 1 to 6:

- Rule 1: If PI is between 1 to 2.5 grade= "Average"
- Rule 2: If PI is between 2.51 to 4.5 grade= "good"
- Rule 3: If PI is between 4.51 to 6 grade= "excellent"

5 Experimentation

In the experimentation we have taken the students of BTech I and II semesters of our department. Questionnaire of subject topic category i.e. analytical (A), reasoning(R), descriptive (D), was given to the student in each courseware. Examination data have been given in table 1.

Table 2: Comparative view of our method with other's method.

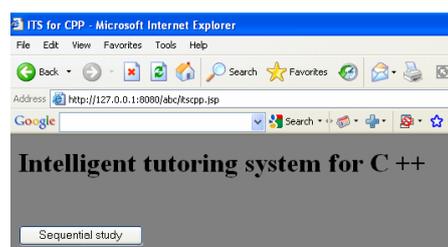
Author	Computing Model	Evaluating Parameter	Evaluation measure	Implementation
Our method	Heuristic method	Number of attempts, Hints	Average, good, excellent	JSP
Zhou et al.(1999)[18]	Heuristic method	Number of errors the student made while solving the problem, Hints	Poor, good	LISP
El-Khouly et al.(2000)[6]	Heuristic method	number of questions which had been asked to the student, correct answer	Not specify	Java , HTML
Tchetagni et al.(2002)[16]	Mathematical method	number of solved exercises, average time to solve exercises, number of tutor interventions(hint)	Not specify	Not specify
Hatzilygeroudis et al.,2004 [8]	Algorithmic method	number of times the user asked for assistance (hint), the number of related examples requested by the user, the number of answering attempts made by the user	Low, high, average, excellent	Not specify
Sykes(2004)[15]	Algorithmic method	problems solved, problems attempted, number of attempts for each problem, Hint	Not specify	JSP
Piramuthu(2005) [12]	Heuristic method	The amount of time spent per lesson, the amount of (uninterrupted) time spent per session, the number of times the student went back, the frequency of help requests(hint), average time spent on a given "page" during a lesson plan	Learning effective(yes/no)	Not specify
Ferguson et al.(2006)[7]	Mathematical method	number of problems the students answered correctly, the number of problem the student attempted to answer (did not skip), Hint	Not specify	Not specify
Stankov et al.(2007)[14]	Algorithmic method	problem difficulty, Hint	insufficient mark, sufficient mark, good mark, very good mark and excellent mark.	ASP, ODBC etc.
Abu Naser(2008)[2],[1]	Mathematical method	Problems attempted problems solved, number of attempts on a problem , problem difficulty, Hint	Not specify	Not specify

There are 10 columns and 6 rows in table 1. Column 1 represents student number, column 2, 3, 4, 5, 6, 7 and 8 represents performance of student in CW1, CW2, CW3, CW4, CW5, CW6 and CW7 respectively. Information i.e. 1st, 2nd and 3rd in brackets in these columns represents attempts in which student's performance is up to level in the courseware. Column 9 represents PI(Performance Index) of the student calculated by the system using equation (1). Column 10 represents grading assigned based on PI by the system to the student.

6 Result

Results and intermediate results of the system have been shown in this section in figures 2, 3, and 4. Figure 2 shows that student starts his study from CW1 by clicking on sequential study. Figure 3 shows student's performance in analytical, reasoning and descriptive and average performance in a courseware. Figure 4 shows student's overall performance (performance in CW1 to

CW7), performance index and grade obtained by the student. Column 1 in figure 4 shows courseware, column 2 shows number of attempts in which student's performance is up to level, column 3, 4 and 5 shows marks obtained by student in analytical, reasoning and descriptive respectively. Total marks obtained by student based on number of attempts have been shown in column 6.

**Figure 2:** Student starts his study by clicking on sequential study.

student	Analytical	Reasoning	Descriptive	Average
student	2.0	2.0	2.0	2.0

Figure 3: Student's performance in a courseware

courseware	Success in number of attempts	Marks in Analytical	Marks in Reasoning	Marks in Descriptive	Total Marks obtained
C/W1	1	2	2	2	6
C/W2	2	2	2	2	4
C/W3	2	2	2	2	4
C/W4	1	2	2	2	6
C/W5	2	2	2	2	4
C/W6	1	2	2	2	6
C/W7	2	2	2	2	4

Performance index: 4.8
Grade: Excellent

Figure 4: Student's overall performance evaluated by system

7 Comparison

In this section we have represented a comparative view of our method with other's methods. A comparative view of our method with other's method has been shown in table 2.

Figure 5 represents a qualitative evaluation of intelligent tutoring systems. We have been taken 10 ITSS out of which 40 percent ITSS have been used heuristic method, 30 percent ITSS have been used mathematical method and 30 percent ITSS have been used algorithmic method for performance evaluation. 90 percent ITSS have been used hint and 40 percent ITSS have been used number of attempts as parameter for performance evaluation. 40 percent ITSS have been used qualitative evaluation measure (Q Measure) i.e. low, good, and excellent for performance evaluation. One ITSS has been used other measure (Learning effective Yes/No) while 50 percent ITSS have not been used (No measure) any qualitative evaluation measure for performance evaluation.

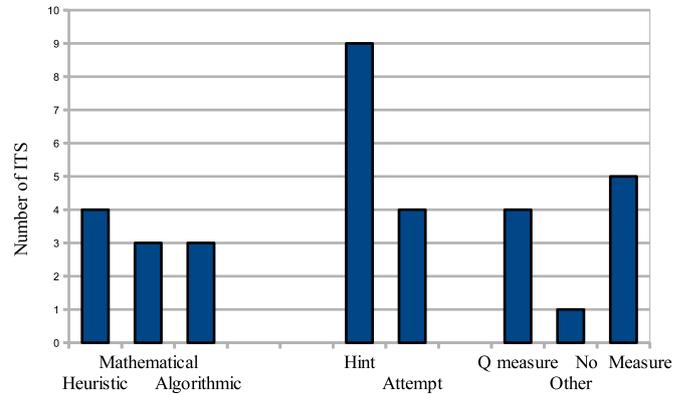


Figure 5: Qualitative evaluation of intelligent tutoring systems

8 Conclusion

A learning strategy for study and a heuristic method for performance evaluation of learning in an intelligent tutoring system has been developed and presented in this paper. It is observed from the table2 and the graph (figure 5) that heuristic methods have been deployed highly in comparison to the mathematical and algorithmic methods. Number of attempts with hint is more favorable and recommendable to the students. Qualitative measure such as average, good and excellent can also be put in the category of very low, low, medium, high, very high mapping in the range of 0-10. Most of the ITSS are web based services deploying JSP, ODBC, HTML etc. Taking into account much more related researches, and number of students in experimentation the table and the graph can be extended. There is a scope of further research using ANN and Bayesian network.

References

- [1] Abu Naser, S. S. An agent based intelligent tutoring system for parameter passing in java programming. *Journal of Theoretical and Applied Information Technology*, 4:585–589, 2008.
- [2] Abu Naser, S. S. Developing an intelligent tutoring system for students learning to program in c++. *Information technology journal*, 7:1055–1060, 2008.
- [3] Anderson, J. R. and Skwarecki, E. The automated tutoring of introductory computer programming. *Communications of the ACM*, 29:842–849, 1986.
- [4] Butz, C., Hua, S., and Maguire, R. Bits: a bayesian intelligent tutoring system for computer programming.

- <http://www.cs.ubc.ca/wccce/program04/Papers/butz.pdf> accessed on 4/11/09,2001.
- [5] E., S., Robin, E., Woolf, B., and Johnson, J., W.L. and Bonar. Meno ii :an ai based programming tutor. *Journal of computer based Instruction*, 10:20–34, 1983.
- [6] El-Khouly, M., Far, B., and Koono, Z. Expert tutoring system for teaching computer programming languages. *Expert Systems with Applications*, 18:27–32, 2000.
- [7] Ferguson Kimberly, M. S. W. B., Arroyo Ivon and Andy, B. Improving intelligent tutoring systems: Using expectation maximization to learn student skill levels. 4053:453–462, 2006.
- [8] Hatzilygeroudisa, I. and Prentzas, J. Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. *Expert Systems with Applications*, 26:477–492, 2004.
- [9] Johnson, W. and Proust, S. Knowledge based program understanding. *IEEE Trans. software engineering*, 11:267–275, 1985.
- [10] Kumar, A. Model-based reasoning for domain modeling in a web-based intelligent tutoring system to help students learn to debug c++ programs. *Intel. Tutoring Syst*, pages 792–801, 2002.
- [11] Mishra, K. and Mishra, R. Multiagent based selection of tutor-subject-student paradigm in an intelligent tutoring system. *International Journal of Intelligent Information Technology*, 6:46–70, Jan 2010.
- [12] Piramuthu, S. Knowledge-based web-enabled agents and intelligent tutoring systems. *IEEE transactions on education*, 48:750–756, 2005.
- [13] Sierra, E., Hossian, A., Britos, P., Rodriguez, D., and Garcia-Martinez, R. A multi-agent intelligent tutoring system for learning computer programming. In *Electronics, Robotics and Automotive Mechanics Conference*, pages 382– 385. IEEE Computer Society Press, 2007.
- [14] Stankov, S., Rosic, M., Zitko, B., and Grubisic, A. Tex-sys model for building intelligent tutoring systems. *Computers and Education*, 51:1017–1036, 2008.
- [15] Sykes, E. R. Qualitative evaluation of the java intelligent tutoring system. *systemics, cybernetics and informatics*, 3:49–60, 2004.
- [16] Tchetagni, J. M. P. and Nkambou, R. Hierarchical representation and evaluation of the student in an intelligent tutoring system. *Lecture notes in computer science*, 2002.
- [17] Weber, G. and Mollenberg, A. Elm programming environment: A tutoring system for lisp beginners. *Cognition and computer programming*, pages 373–408, 1995.
- [18] Zhou, Y. and Evens, M. W. A practical student model in an intelligent tutoring system. pages 13–18, Chicago, 1999.