

A computer aided learn by example system

M. AIRES AND L. DESCALCO *

Departamento de Matemática, Universidade de Aveiro
Campus Universitário de Santiago, 3810-193 AVEIRO, Portugal
March 28, 2006

Abstract

We present a learn by example system being developed in the University of Aveiro. This system uses the software platform, PeA, of the project PmatE, which provides a way to store and run our example questions (more precisely question generator models). Several PeA models for the course “Cálculo III” (calculus with several variables), have been produced recently. The contents of the course is divided in several topics, that we associate with objectives. The system allows a student to answer questions within several objectives without a predetermined order. After each answer, a new objective is chosen based on the questions the student has provided so far and hence, on the knowledge the student has shown so far.

1 Introduction

The use of computers in education has been a subject of research (see for example [1], [3], [9]). We believe there are many benefits in a sensible use of computers as tool for learning mathematics.

In this paper we describe a computer system that allows a student to answer questions about several subjects without a pre-determined order. After each answer, a new subject is chosen based on the answers the student has provided so far.

One advantage of answering questions in a computer system is that feedback can be provided immediately: the student knows if his answer is right or wrong. But usually, when a set of questions is chosen, to be answered by several students, all students are supposed to progress at the same speed, independently of the knowledge they already have, or at most, the difficulty level is changed in a pre-established manner. We are working on a system that allows each student to progress at his own speed. If he does not answer correctly to some questions, then more basic question will be asked, which are the basis of those he does not know how to answer. When he answers correctly a

*The first author is a MSc student, working on the subject of this paper and the second is the supervisor, member of PmatE and coordinator of the course “Cálculo III”.

new topic is chosen and he progresses to more difficult subjects or those he has not yet answered correctly.

The paper is divided in five sections. In the second section we describe briefly the software platform of “Projecto Matemática Ensino (PmatE)” (see [4], [6]), in which we base the system. The following section describes the organization of the mathematical contents of the course “Cálculo III” (calculus with several variables) which constitutes the pedagogic material of the computer system. The fourth section describes the computer system itself, which allows a student to navigate through several topics of the course according to the correctness of his answers. Finally, in the last section, we refer to some issues both technical and pedagogical, and possible optimizations for the system, which is still being tested.

2 PmatE platform

The software platform of PmatE, named PeA, which stands for “Plataforma de Ensino Assistido” (Assisted Teaching Platform), was initiated in 1990, by David Vieira, Batel Anjo and Paula Carvalho, has grown with contributions from many people, and is presently available on the Internet, with contents from primary school to University courses. In the beginning it was used only for mathematics but it is now being used also for other subjects, including University courses from other departments, such as “Sistemas Digitais” (Digital Systems).

Basically, the PeA allows us to store and run question models and keep track of the students progress through a set of questions. A question model is a set of questions with some parameters. These parameters are instantiated randomly by the computer when the model is selected to be executed. So, instead of storing something as “Is $1+1 = 2$?” in PeA, we store “Is $a + b = c$?” together with the information “ $a, b \in \{0, 1\}$, $c \in \{0, 1, 2\}$ ”, for example.

A question model consists of four or more parameterized questions. When a model is executed by the computer, four parameterized questions are randomly selected, and then their parameters are randomly instantiated from the associated sets. So, the student sees four questions at a time, filling the computer screen, and must answer true or false to each of them. Hence, a single question model can generate a huge number of similar but different questions.

The idea is extremely simple but the usefulness of this system has shown to be enormous. It has been used as a diagnostic tool for students entering the University, as an assessment tool, in mathematical games and competitions engaging thousands of students with the purpose of motivating them for mathematics, etc.

3 Learning calculus with several variables

The contents of Cálculo III is the classic contents of calculus with several variables: topology, continuity, partial derivatives and differentiability, Taylor Formula, Lagrange multipliers and optimization, multiple integrals, line and surface integrals and partial differential equations. We note that the number of students of this course is larger than 1000 every year, which makes the use of automatic tools for teaching and assessment, like the PeA platform, very appealing.

First of all, we have the contents of the course divided into several objectives. Inside each objective we have our question models. We imagine the contents as a graph, where nodes are objectives. Hanged in each node we have the models associated with the corresponding objective. The edges of the graph can be seen as connections between the objectives. For example, a student must be able to compute partial derivatives for computing Taylor Formulae and so we have an arrow from the objective “Compute partial derivative” to the objective “Compute Taylor Formula”.

The idea is that a student, studying calculus, wishing to solve exercises, uses our system as tool for learning. The system should select a first objective, randomly from a chosen subset of objectives, corresponding to a node in the graph. The student starts answering questions in the selected node. Then, he starts moving from node to node according to the knowledge he is showing with his answers. The process goes on until some established knowledge within the contents of the course is achieved.

These transitions between nodes are supposed to be intelligent. For example, if a student does not answer correctly about Taylor Formula it is more probable that he will be answering questions about partial derivatives in the future. If his answers are correct, then the probability of moving to a more complex objective, such as using the Taylor Formula to study function extremes, increases.

4 Computer aided learning

The computer system is inspired by the concept of probabilistic automata (see [5], [7], [8]). To perform transitions between the nodes of the graph, representing the contents of our course, we assume the graph is complete (each two nodes are connected by a directed edge) and has probabilities associated with the edges. The sum of probabilities of all arrows leaving a node is one. These probabilities are computed each time the student answers a question.

We represent all information by using four matrices, B , D , K and C , which are used to compute probabilities after each answer by the student. Matrix B (Basis) gives the structure of a tree to our objective nodes, reflecting the dependence between the objectives (the graph with the objectives is represented in Figure 1). If a student fails in some objective, the probability of moving to objectives it depends on should increase; hence we need matrix B . Matrix D (Distance) has information about of the distance

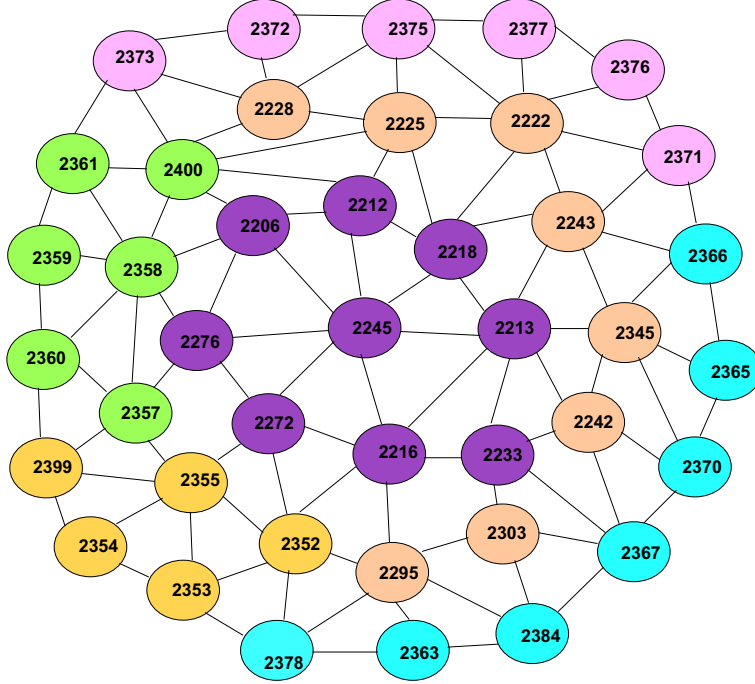


Figure 1: Graph of objectives

between objectives. This is used to avoid or allow jumping to subjects that are not closely related. These two matrices constitute the information that will not change during the execution of the program: the organization of the course contents we need. Matrix K (Knowledge), which is in fact a vector, contains the knowledge shown by the student in each objective. Matrix C (Counter) is a vector of counters containing the number of questions the student has answered about each objective. These two vectors are updated after each student's answer.

Presently, there are $n = 41$ objectives and 68 models for the course. When the program starts, the n positions of vector K are at 50% and those of vector C have zeros. The vector of probabilities, P , is initialized based on B , D , K and C . Then, by using P , the first node is selected. The initial values of P are such that the first selected node does not correspond to a very advanced objective. A model hanged on the selected node is randomly chosen for execution and the student answers its questions.

After the student's answer, vectors K and C are updated. If he answers correctly to more than half the questions, from a model with objective i , then K_i is increased. Otherwise it stays the same or decreases. The counter C_i is incremented by one. Then vector P is updated by using the formula

$$P_j = \frac{w_B \bar{B}_j + w_D \bar{D}_j + w_K \bar{K}_j + w_C \bar{C}_j}{w_B \sum_{i=1}^n \bar{B}_i + w_D \sum_{i=1}^n \bar{D}_i + w_K \sum_{i=1}^n \bar{K}_i + w_C \sum_{i=1}^n \bar{C}_i} \quad (j = 1 \dots n)$$

where each \bar{A} is a vector of positive numbers corresponding to the contribution of matrix A , and w_B, w_D, w_K, w_C are positive constants. Finally, by using P , another node is selected and a model hanged in the node is randomly chosen and executed. This process continues until “vector K is good enough”.

5 Conclusion

This simple system is supposed to make a student travel in the graph of the course contents and help him learn calculus at his own speed. The system tries to guess the student difficulties by sending him to questions about subjects that constitute a basis for the subject about which he does not answer correctly. If the student has already shown some knowledge about a subject then it is less probable he will be questioned about it. This is the general idea, but there are many possibilities for improvements of this basic system.

In the description of the algorithm we have omitted several important details that affect the behavior of the system. For example, the amount to increase or decrease vector K , according to the number of correct answers. The stopping condition, “vector K is good enough”, is also adjustable. Presently, we are imposing that a minimum number of positions in K are above a certain number, and we are assuming the amount to increase for four correct answers should be three times the amount for three correct answers. These quantities, together with the stopping condition, determine the quantity of correct answer a student must give within each objective, and the number of objectives he has to satisfy.

The values of the positive constants w_B, w_D, w_K and w_C have crucial importance. We have these numbers as parameters to be adjusted. The values of these constants must be changed during the execution of the program. For example, constant w_D should be higher when the program starts to make the student stay for a while near the first chosen subject. But after a certain point, when the knowledge shown is high in many topics, it is more important that w_C is high so that a topic where he has not been before is chosen. If, for example, a student has answered many questions and most of his answers are wrong, then the value of w_B should be increased, since the most important in this situation is to make the student acquire basic knowledge.

From the pedagogical point of view, the quality of the modules, and the information in matrices B and D is very important. New modules are still being created and carefully tested which include both theoretical and applied questions.

A program in C programming language is presently being tested, to simulate the paths of the students trough the graph, in order to adjust the values of the parameters to be used in the first test with students.

We observe that a system of this kind does not depend on the course neither on platform PeA. It could perfectly be used in any other subject and with other kind of questions, not only questions with true or false answers.

Our main goal is to improve this basic system and to create a useful system allowing a student to answer questions about subjects in an order that is not pre-determined. These subjects are automatically selected by the system taking into account the revealed difficulties of the student. Hence, it is possible for him, to travel over the several topics according to the correctness of his answers and, most of all, at his own speed and needs.

References

- [1] Esther R. Steinberg, *Teaching Computers to Teach*. Lawrence Erlbaum Associates. Hillsdale (1991), NJ. Publication.
- [2] Paulo Mateus, *Universal aspects of probabilistic automata*. Math. Structures Comput. Sci. 12, no. 4 (2002), 481-512.
- [3] G. McMeen, *Modern technology in education: From teaching machine to microcomputers and student response systems*. Educational Technology 16 (1986), 20-24.
- [4] Maria Paula de Sousa Oliveira and Sabrina Vieira da Silva, *An overview of PmatE: developing software for all degrees of teaching*, ICMSE 2006.
- [5] Azaria Paz, *Introduction to probabilistic automata*. Academic Press, New York-London (1971) xxv+228 pp. 94.45.
- [6] Projecto Matemática Ensino (PmatE): <http://pmate.mat.ua.pt/>
- [7] M. O. Rabin, *Probabilistic automata*, Information and Control 6 (1963), 230-245.
- [8] Mariëlle Stoelinga, *An introduction to probabilistic automata*. Bull. Eur. Assoc. Theor. Comput. Sci. EATCS No. 78 (2002), 176-198.
- [9] J. Whiting, *Student opinion of tutorial CAL*. Computers in Education 10(2) (1986), 281-292.