

Model Parameters and Model Performance

Jan Smid¹, Jan Smid Jr.², Marek Obitko³, and Petr Volf⁴

¹ Morgan State University, Baltimore, MD, USA, jsmid@morgan.edu

² ERHS, Greenbelt, MD, USA, smid_j@hotmail.com

³ Dept. of Cybernetics, Czech Technical University, Prague, Czech Republic,
obitko@labe.felk.cvut.cz

⁴ UTIA, Prague, Czech Republic, volf@utia.cas.cz

Abstract. We have proposed and tested several versions of a mini-tutorial system. The system, called the Information Cycling and Diagnosing (ICD) system repeats, or cycles, information on the basis of expected or tested knowledge of the user. The goal of our system is to cycle information. We have compared several mini-systems, including one with a preset number of operations as well as a flexible version where the number of cycles depends on the user level and on the user answers. Information consists of a pool of definitions, examples, and quizzes. Cycling of the same type of information is provided until the user is able to successfully answer a series of questions. Scoring of the user knowledge and a mild remediation are the features that make the system adaptive. We have found that this type of micro-systems is useful in a classroom setting when used as an introductory homework assignment.

1 Introduction

Researchers in the Intelligent Tutoring System (ITS) community have been considering Bayesian Networks as a mechanism for student modeling (e.g. [3,2]) and knowledge representation (e.g. [1]). User profiling can be based on a pre-tutorial questionnaire and on-line data evaluation (for example ELM on-line system [8]). These and many other approaches to ITS (e.g. [7]) introduce models with parameters that are subject to user oriented adaptation.

Our goal is to develop a simple system that is effective in the average introductory college classroom. The average classroom is defined here as one made up of students who are required to take mathematics but do not grasp the deeper principles of mathematics such as proofs. The differences among users in a classroom setup is skill level rather than a background level. The targeted users for our ICD experiments were undergraduate students of mathematics and computer science at Morgan State University and senior high school students at Eleanor Roosevelt High School.

A user can be characterized by many parameters. We have limited our attention to only two adaptive parameters for each information unit. These parameters control the number of repetitions (cycles) and the test passing criterion. The knowledge domains we investigated were algorithms for solving linear algebraic equations using the Gauss-Jordan method, theoretical computer science such as Euler Cycle theorem and an elementary vocabulary for French.

2 System Architecture

The ICD system consists of units and nodes. We will use the terms unit and node interchangeably.

Each node represents a particular segment of knowledge. Furthermore, nodes can also have links to other dependant nodes. Nodes represent elements of information for a given knowledge domain and degree of detail or abstraction. Thus one domain can have more than one tree of nodes. For example, one tree can represent memorization of keywords for the Euler cycle theorem. The other tree may represent the complete proof of the theorem. This tree structure is similar to chapters containing sections, sections containing paragraphs and so on. Each node has definitions, procedures, and examples. Nodes are assigned numbers that represent user knowledge. The weights assigned to edges between nodes represent the amount of content of prerequisites. The more content the child node contributes to the parent node, the higher the corresponding weight. Many important domains can be structured as simple two or three layer trees with low in-degree (2 or 3) vertices.

The following are examples of different domains that use the same cycling principle:

2.1 Test Domain 1 — Gauss-Jordan

The Gauss-Jordan procedure for solving linear algebraic equations consists of several elimination steps applied to a matrix representing a system of linear algebraic equations. We can think of this as a unit having several dependent nodes. These nodes are individual elimination operations that are shown to the user.

2.2 Test Domain 2 — Euler Cycle Theorem

The Euler cycle theorem is a theorem from graph theory: 'A connected multi-graph has an Euler circuit if and only if each of its vertices has even degree.' This theorem consists of several definitions and a statement that is valid under these assumptions. One useful task is to memorize the necessary definitions of the theorem. The theorem is a node that has, as pre-requisites, a node of Euler cycle definition and a node of even degree definition and so on (see figure 1).

2.3 Test Domain 3 — Foreign Language

In a foreign language vocabulary builder the task is to memorize a pool of words and be able to form those into sentences. A sentence or several sentences form one node and prerequisite nodes are individual verbs, nouns, adjectives, and other parts of the sentence. Each node is associated with a set of tests. These tests are multiple choice tests and free-form tests. The free-form tests use the asterisk style (described below). The scoring function is defined based on the

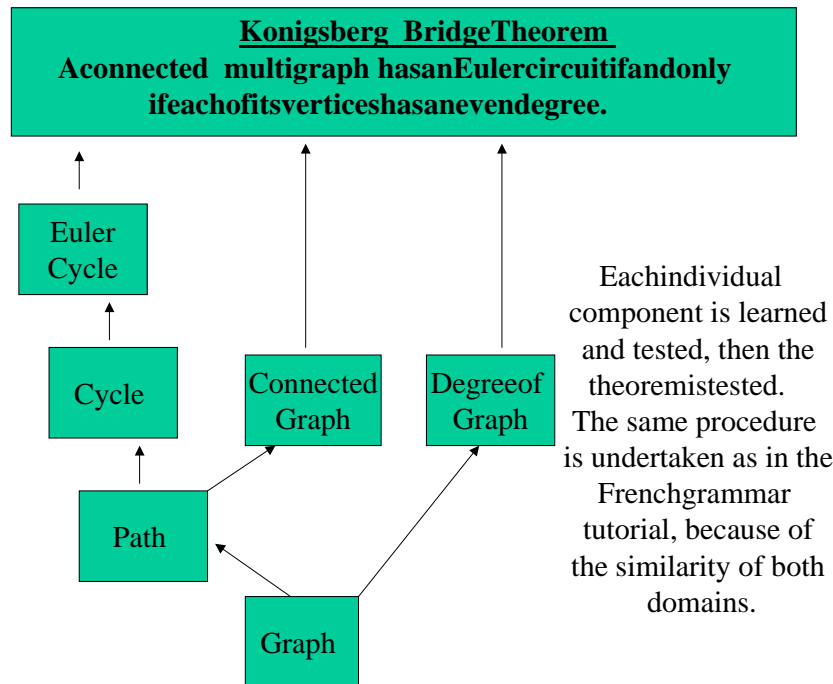


Fig. 1. Definitions-Theorem Mini-Domain

number of asterisks. In a fixed structure sentence, the user is tutored to be able to translate or fill in missing words. The nodes for language consists of individual words (layer 1), pairs of words (layer 2), and fixed structure sentences (noun, verb, adjective) (layer 3). Free-text formatted questions were chosen as the most desirable format because guessing is decreased when the user lacks a list of possible answers as in the multiple-choice tests. Thus, the probability of selecting a right answer with little or no knowledge is eliminated. Hints are given to help a user that is having much difficulty performing a certain task. These hints are considered very valuable because it maintains the user's interest without revealing away the answer. Examples of hints can be coded as a series of strings. The "*" is used to disguise the answer. For example, *****, *p***, *pp*e, are used as hints or test patterns for the word apple (see figure 2).

3 Tutoring Strategy

The domain can be structured as several different sequences of steps. Each sequence has units of different size reflecting skills and needs of the user. The author can generally devise tutorial units with sufficiently fine granularity. The size of the unit guarantees that the system can accommodate users of varying

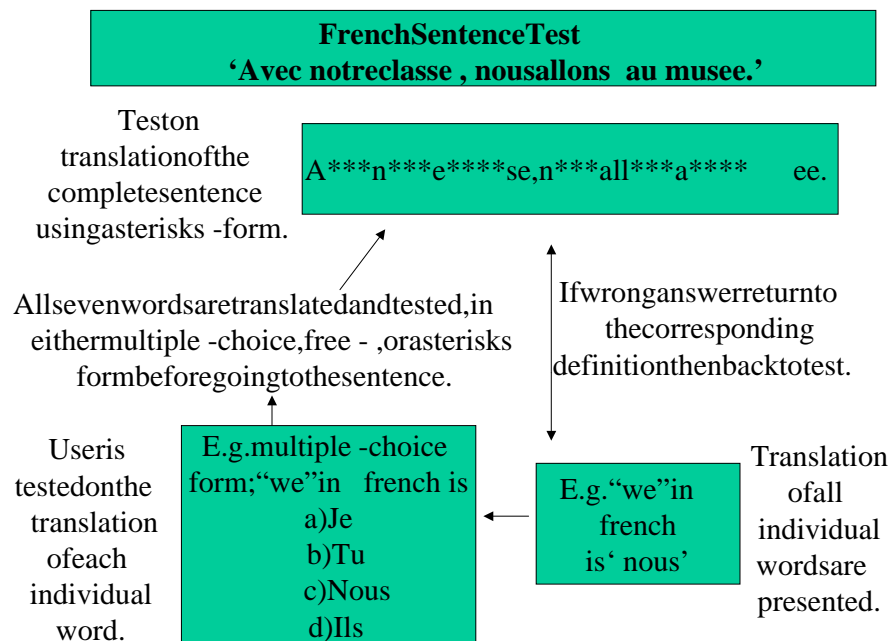


Fig. 2. Words-Sentence Mini-Domain

instructional needs. Thus, the objective of the tutorial is to cycle a sequence of steps of the appropriate size for the user until certain scoring criterion is satisfied. Cycling can be done with the same information (e.g. same definition), different numerical parameters (e.g. row1: $2x + 3y = 5$ instead of row1: $x + 4y = 5$) or different number of asterisks (e.g. ***le instead of *pple).

Similar to the classroom environment, the number of presentations needed for an information unit differs from unit to unit and from user to user. The teaching strategy that is used can be described as follows:

1. The unit that caused the failure of a test is presented. For example, incorrectly translated word in a sentence is provided.
2. The unit with the smallest score (the least learned) is presented.
3. The unit that is likely to account for a failed test is re-visited. This can be achieved by Bayesian calculations provided data is available for the reliable description of dependencies among units. If the data or expert knowledge supporting Bayesian decisions is unavailable we can execute the least scored unit (2).

The overall tutoring goal is to achieve a certain score level for each unit that will guarantee an improvement on the post-test from the pre-test.

4 Adaptation Parameters

Every user and each unit is associated with parameters of a scoring function. A scoring function can have many parameters. We assume only one parameter scoring function where the parameter controls the total number of presentations of the associated unit. For example in the case of computer basics we have used a scoring function family was a two-parameter function:

$$\begin{aligned} f(s) &= \min(0.9, s + p_1) \text{ for a presentation,} \\ f(s) &= 1 \text{ if } s = 0.9 \text{ else } f(s) = 0.9 \text{ for a positive test,} \\ f(s) &= \min(0.9, s + p_2) \text{ for a negative test,} \end{aligned}$$

where s is the previous and $f(s)$ is the updated score.

Typical numerical values are $p_1 = 0.3$, $p_2 = 0.1$, initial $s = 0$. Thus, the parameters p_1 , p_2 control frequency of repetition of an information units for the user. The parameters p_1 , p_2 can have a wide range of values for different users and different units.

5 Remediation

In order to understand information that has prerequisite nodes, the children nodes must be taught first. Only after dependent nodes have collected sufficient scores is the parent node cycled and tested. If the parent unit tests are failed then the user returns to instruction of the prerequisites. The units that are retaught and retested are those that either have identifiable weak components or have the most number of repetitions in previous instruction. For example, in the French tutorial the sentence "Je vais nager" (I am going to swim) is cycled. Let us assume that the user's answer for a translation is "Je vais dormir" (I am going to sleep) then the user is returned back to the identifiable weak node for the verb "nager." Alternatively, if a user is presented with a graph and fails to identify it as one with an Euler Cycle then there is no direct indication of which component was misunderstood. A choice can be made on the basis of the Bayes technique, in a framework of Bayesian network scheme (BN) [4]. The primary scores from ICD are updated after a unit or a test is completed, and new evidence from primary scoring is propagated using conditional probabilities. In such a way a secondary scoring is obtained which provides a basis for remediation (see also [5]).

The basic BN problem can be formulated as to calculate node probabilities $P(A, B, C, \dots)$ given a set of conditional probabilities and a new evidence in a node e.g. B . New evidence comes from a scoring function that in turn can be optimized. A serious problem is how to initialize the conditional probabilities. This problem will require systematic data collection and analysis.

6 Statistical Challenge

A substantial challenge is posed by the parameter estimation problem. Scoring function parameters for each unit and the weights between units need to be set

up. Scoring will depend on both the algorithm complexity and the user abilities. Complexity of algorithms can be estimated using a set of base algorithms. For example, in the case of GJ procedure, three different complexities can be represented by simple keyword memorization, matrix multiplication, and Gauss elimination. Regardless of how we set up scoring and the weights, numerical values need to be adjusted when dealing with user data. Properly set up scoring and weights will tell the user about his/her knowledge of the domain. The user should be allowed to modify scoring function parameters in order to increase or decrease the frequency of presentations. For example, the user may pass a quiz on a verb but desires to solidify the current knowledge.

7 Experiments

Simple definitions from computer science and the Euler circuit theorem were implemented and tested in classrooms as quizzes in the free- and multiple-choice form. These two test are structured essentially the same as the vocabulary builder domain. Free-form style showed better results of the two. For the simplest domains the scoring function parameters can be set at the same value for all users. Typical results are shown in figure 3. The two curves show the number of mistakes for users before and after a tutorial. There was an obvious improvement in the post-test scores using the simplest scoring function (one presentation, and testing until a correct answer) This behavior has been seen over a class of keyword-type of tutorial. We refer to these domains as low sensitivity domains because the user test results do not improve considerably with higher scoring parameters values.

Experiments where the scoring function is set excessively high does not yield a better post-test result and the users generally do not select use this system if given a choice. However, the users indicated that they would use the system with correctly set scoring function.

Measuring the number of operations was implemented in the Gauss-Jordan elimination experiment [6]. The Gauss-Jordan procedure showed a dramatic difference in the number of operations used by different users (see figure 4). The GJ procedure requires 10 – 15 operations for the average user, word memorization 1 – 2 operations for the average user. We were able to find a model for prediction of the number of operation necessary for the individual user. This model was a regression model where the input was a window of first n user answers. Good regression fits were obtained for numerical values $n = 9, 10$. This is promising because the results indicate that system may be able to adapt to on-line data analysis.

Figure 4 shows the number of operations requested by the user before a quiz was passed. This experiment showed that in complicated domains the sensitivity of a scoring function with respect to its parameters is high and should be adapted to the user.

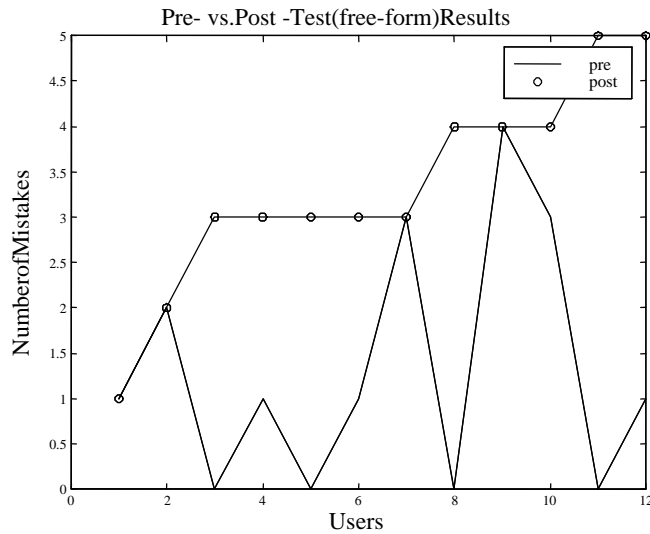


Fig. 3. Typical Results for Domains with Low Sensitivity to Scoring Functions

8 Conclusion And Future Work

What to implement in the future? The number of issues ranges from mixed initiative parameter adjustment (system and user adjustments allowed) to authoring and interface concerns. We can also expand the number of new parameters. A tutoring micro-strategy could be another user-adjusted parameter. For example some users like an abstract definition presented first and examples later, other users might prefer the reverse sequence. This problem could be approached using planning techniques from AI.

Domain as a dependency tree and scoring based on similarity measure for algorithms (scoring is correlated with the number of steps of the procedure) is another issue that should be solved using mixed initiative (automatic data analysis in collaboration with user adjusted parameters).

The technology is available to use natural language processing to facilitate communication between the user and the system. The system could parse and interpret the user input and return a tutorial specific to the user with the requested detail and set scoring function parameters. In this sense the next development of user modeling may lead to system-user negotiation. An important component to tutoring strategy success is the typical user request to show a relationship between two concepts. Planning a tutoring strategy can be phrased as a search for a path linking two or more concepts given by the user. The information is available in a (electronic) book but a system search can create a desirable short cut. For example the user can ask how to reduce one matrix to an upper triangular matrix, or about the relationship between the degree of a node and the Euler cycle. Parameters for scoring functions and other parameters can be

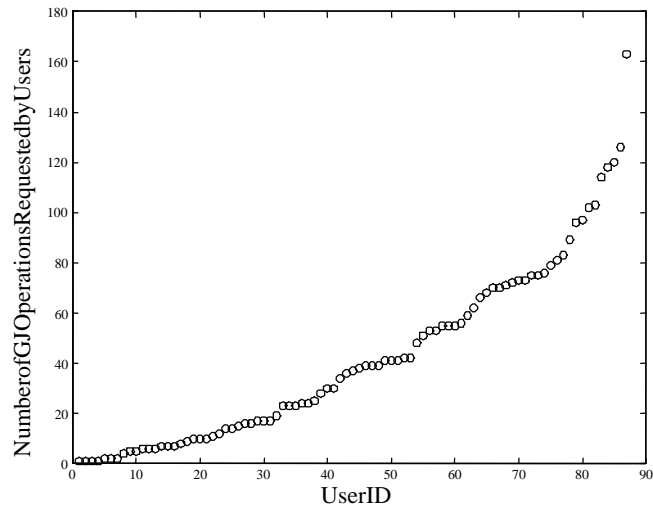


Fig. 4. Domain with High Sensitivity to Scoring Functions

set and modified automatically but this option is demanding on data. ICD can minimize the number of operations required to master information units. In the manual mode well adapted scoring can provide valuable self-guided study tool. The network of knowledge with assigned scores and weights will provide a new type of “textbooks”.

References

1. F. de Rossis, S. Pizzutilo, A. Ruso, D. Berry, F. Molina: Modeling the User Knowledge by Belief Networks. *User Modeling and User Adapted Interaction* 2:367-388, 1992.
2. G. Gauthier, C. Frasson and K. VanLehn: *Intelligent tutoring systems*, 5th international conference, ITS 2000, Montreal, 2000
3. Anthony Jameson: Numerical Uncertainty Management in User and Student Modeling. *User Modeling and User-Adapted Interaction* 5, pp193-251, 1996
4. Finn V. Jensen: *An Introduction to Bayesian Networks*. Springer 1996
5. P. Volf and J. Smid: Randomized Evaluation of States of an ITS. *Modeling and Simulation of Systems Conference MOSIS 01*, MARQ Ostrava, CZ, 2001, 227-232.
6. J. Smid, P. Svacek and J. Smid Jr.: Processing User Data for Intelligent Tutoring Models. Workshop within KDD-2000: *Post-Processing in Machine Learning and Data Mining: Interpretation, Visualization, Integration, and Related Topics*. Boston, MA, August, 2000
7. UM99 - User Modeling. *Proceedings of the Seventh International Conference*. Edited by Judy Kay. Springer Verlag 1999.
8. Weber G., Specht M. (1997) User modeling and adaptive navigation support in WWW-based tutoring systems. <http://www.psychologie.uni-trier.de:8000/projects/ELM/elmart.html>