

Approximate Degrees of Similarity between a User's Knowledge and the Tutorial Systems' Knowledge Base

Namdar Mogharreban

Southern Illinois University, Carbondale, IL. USA

namdar@cs.siu.edu

Executive Summary

A typical tutorial system functions by means of interaction between four components: the expert knowledge base component, the inference engine component, the learner's knowledge component and the user interface component. In typical tutorial systems the interaction and the sequence of presentation as well as the mode of evaluation are predetermined and follow a somewhat linear sequence. This model was implemented in many of the early computer based trainings, computer assisted instruction systems and tutorial drill programs. However, by introducing artificial intelligence in the inference engine or by enhancing the expert system component (by means of including feedback), by improving the evaluation of the learners responses and facilitating interaction between these components one may provide a learning environment that more closely resembles a real teacher and student interaction. This approach is known as Intelligent Tutoring Systems (ITS). Various tutorial systems were developed based on this paradigm that proved useful in knowledge domain areas that are highly structured and relatively small (e.g. solving math problems or balancing chemical equations). The difficulty resides in the complexity involved in making the various components encompassing and complete in a knowledge area. For instance, understanding why learners commit a particular error and then assisting them is highly challenging since the cause of an error might be different for every learner. Variations on this ITS model have been employed with success in developing tutoring systems in less structured knowledge domains and more generic environments.

Another element that improves ITS functionality is the ability to deliver the correct and necessary granule of material for effective coverage and completion of the knowledge area. The question of where to start a learner in the tutorial system and how to choose the next step is difficult to delineate. In this paper we propose an approach based on the fuzzy set theory to determine the entry knowledge level possessed by a learner in a specific area of learning. Two relations between the knowledge area and the skill levels of a user are established. The first relation is created between the given behavior or knowledge and the mastery of the foundation skills required for it.

The second relation is between the given behavior or knowledge and the required exposure to the knowledge domain. The matrices are manipulated to drive a set of values. The resulting values reflect the amount of familiarity of a learner with the knowledge domain. These values can be utilized to provide an accurate starting place for the delivery of a training set or curriculum in a given

Material published as part of this journal, either on-line or in print, is copyrighted by the publisher of the Journal of Information Technology Education. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Editor@JITE.org to request redistribution permission.

domain of knowledge. Continuous evaluation and determination of the values can also be utilized to determine the granule of materials to be delivered for the most efficient progress through the knowledge area.

Keywords: tutoring systems, intelligent tutoring systems, learners' characteristics, model-tracing, fuzzy set

Introduction

The classical view of a typical intelligent tutoring system (ITS) involves at least four components. These include the knowledge base or the expert system component KC, the inference engine or the pedagogical component IC, the learner's knowledge component LC and the interaction or the user interface component UC. These various components interact with one another to support the learner in the tutorial environment (McArthur, Lewis, & Bishay, 1994). Figure 1 represents this conceptual model of an ITS. The expert system typically provides the intelligence of the system. It contains the knowledge base as well as the feedback information for errors and questions on the part of the learner. The inference engine contains the evaluation and decision making components of the system. It functions as the determinant of what is educationally sound and acceptable during the interaction. The learner component basically keeps track of the learner's behavior and responses. Finally the most visible component is the user interface, which attempts to provide the most user friendly and intuitive interaction to the learner (Katz, Lesgold, Eggen, & Greenberg, 1996). This component provides the means for inputting to the system and generating output from the system for the user.

Due to the complexity of human learning and the diversity of knowledge domains research is contentious in all of these areas to improve the development and utilization of ITSs. Improvements in graphical user interface, GUI, have enhanced interactions with the user while developments in artificial intelligence research have had great effects on the inference engine. New conceptions in categorizing problem solving tasks and ideas of generic tasks have improved knowledge acquisition and representation in tutorial systems (Chandrasekaran, 1986; El-Sheik & Sticklen, 2003).

In section two of this paper we will discuss the various types of ITSs and the different approaches to their development. In section three we propose an approach to determining the starting point for a tutorial system based on user's knowledge of the domain. Our approach uses the degree of similarity between a learner's amount of knowledge and time exposure to the knowledge area and the knowledge requirements and time exposure needed to master the knowledge domain. The resulting value is an indicator for the level of mastery an individual possesses. It thus can be utilized in determining the starting point in the tutorial for the learner or the amount of materials needed.

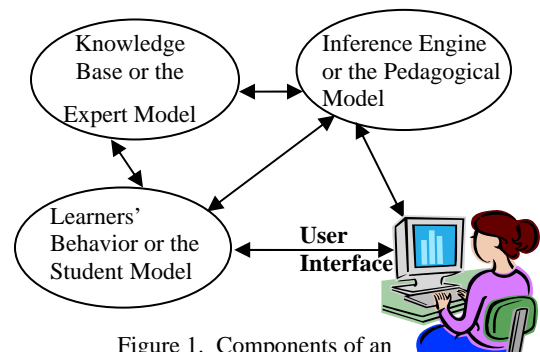


Figure 1. Components of an Intelligent Tutoring System.

Development of Tutoring Systems

Implementation of the four components, KC, IC, LC, and UC fall along a continuum with various amounts of intelligence and interactions occurring between the components of the system (Burns & Capps 1988). At one end of the continuum are the training systems, computer assisted instruction CAI, and to some extent computer based training CBT, software. Typically in this

group of software the level of intelligence is limited to conditional branching based on user response. The interaction is self-paced but predetermined and preprogrammed. The interaction typically is limited to a few cycles of feedback, in case of an incorrect response, followed by the correct answer. Little or no attempt is made at understanding the sources of error committed by the learner. The behavior recorded, responses by the learner, is limited to the number of right and

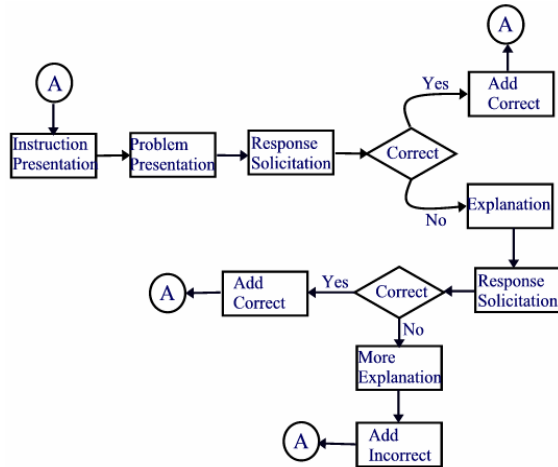


Figure 2. Typical interaction flow in a CAI.

wrong answers. Figure 2 depicts a typical interaction flow of a CAI. The designer has the responsibility of determining the interactions. Therefore, CAIs are useful in limited areas.

At the other end of the spectrum are the ITSs where the expert system component is “omniscience” and compares the behavior of the learner to that of the expert “teacher” and intervenes whenever these two depart. In these tutorials the responses are analyzed and attempts are made to determine the sources of error made by the learner and subsequent feedback is provided. The degree of intervention and mode of intervention is determined by the pedagogical model of the system. Anderson and Pelletier (1991) employ

this approach and refer to it as model-tracing methodology. Successful ITSs were created for solving algebra problems (McArthur & Stasz, 1990) or for tutoring programming languages, in particular LISP (Anderson & Reiser, 1985). The problems with this approach become apparent when attempts are made to develop ITSs that are not within a narrow and well-defined domain of knowledge.

The construction of an ideal student model in which all of the potential combinations of student responses and misgivings are represented is at best difficult if not unworkable (VanLehn, 1986). The problem is both theoretical and practical. However, research in task analysis and artificial intelligence has provided powerful technologies for creating an ideal learner and analyzing a user’s responses. These include such technologies as plan recognition, case-based reasoning, neural networks, fuzzy logic and rule-based problem solving (Anderson, Boyle, Corbett, & Lewis 1990). For example, the expert module of SYPROS consists of a knowledge base of goals and

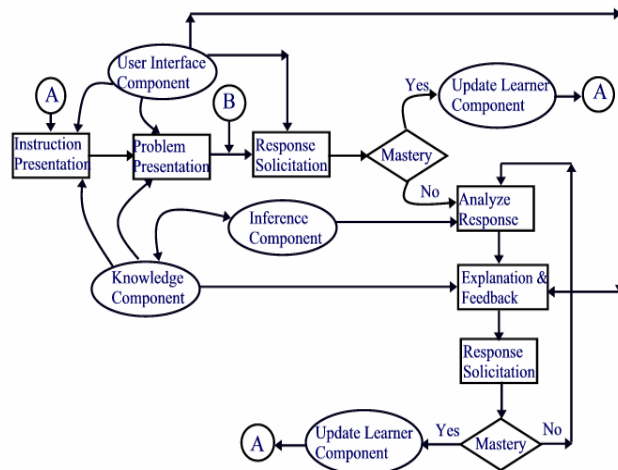


Figure 3. A typical interaction flow in an ITS

plans from which a goal-plan-tree can be generated for every task. This tree describes a correct way to solve the problem as well as frequently occurring mistakes in this task. (Rothenhofer & Herzog, 1993). However, incorporating these technologies into a functional and transferable ITS is very complex and resource intensive, both from a development stance and a user stance. Figure 3 represents a typical interaction flow in an ITS.

The theoretical foundations in cognitive science and learning theories lag behind our technological advancements. Some

have even debated whether the cost of constructing very detailed, complex user models that are runnable and have to be continually maintained is worthwhile in terms of the gain in teaching efficiency. A more moderate approach to the construction of a student model is proposed by Self (1990). He advocates building the student model through the interaction of the student with the system. As such the ideal model need not be omniscient and represents more of what the student actually knows and believes as opposed to what he should know and believe. With such a conception the role of the ITS, then is more of collaborative interaction than evaluative critique.

Yet another approach to development of ITS, which might prove more viable is an environment where learners are provided with a variety of stimulation in an exploratory setting. The difficulty and complexity of representing and coding such systems are also enormous. Virtual wondering is also a problem as (Katz et al., 1996) point out, "exploratory environments provide opportunities that might stimulate learning ... but students often travel down a "garden path", and quit before they have a chance to learn from their mistakes." A shared approach where the instructionist strategies and constructivist strategies are combined has proven effective in development of ITSs (Nussbam, Rosas, Peirano, & Cardenas, 2001). In this approach the learner constructs what is to be done but the expert instructs how it is to be done.

Finally another area that is receiving special attention as a functional as well as feasible approach to development of ITS is the identification of generic tasks that can be applied to all types of problem solving. Using such an approach, a set of reusable blocks that are task related can be developed and utilize to improve and cut down on development time for an ITS (El-Sheik & Sticklen, 2003).

A Starting Point for the Tutorial

Tutorial systems, by definition, are meant to train skills that have been previously taught and not necessarily teach original content. As such users come to tutorial systems with great degree of difference in their background. Where to start the tutorial and how to proceed from one content to another is the subject of much research. One approach, for instance, is to present the student with a question or problem that is selected from a curriculum script. AutoTutor which is an ITS designed to assist college students learn about computer literacy applies such an approach (Person, Graesser, Kreuz, & Pomeroy, 2001). The learner is moved along in AutoTutor by means of a fuzzy production rules that utilizes the data provided by a latent semantic analysis module.

In this paper, we are proposing a general approach to evaluation of the learner's knowledge for starting a tutorial as well as guiding the learner through the learning materials. This approach is based on fuzzy set theory where several values are evaluated to determine the entry skill/knowledge of a learner or to determine the next step or content to be delivered. This can provide a more accurate estimation of the learner's knowledge and skill level without an extensive coding of traditional logical considerations. The resulting value can then be used for the delivery of training materials or curriculum sets in a given knowledge domain. Which factors are ideal to be used for consideration are subject to further research. Here we are using mastery level requirements and exposure to a skill as the factors to use for evaluation. Other factors such as learner's motivation or learning style, task difficulty or importance can be added and evaluated to further refine the value generated for selection of materials to be delivered or to determine the next step to be taken.

Our approach is based on two sets of fuzzy relations. The first relation is created between the given capability or knowledge (K) and the mastery of the foundation skills (F) required for it. We call this R_p relation. The second relation is between the given ability or knowledge and the required familiarity to the knowledge domain. We call this R_e . In the first relation, R_p , the degree of mastery required for a given set of foundation skills will be given a range of values between 0

and 1. The value 0 indicates the least amount of requirements of the foundation skills mastery for a behavior or knowledge, i.e. least essential. The value 1 indicates the most degree of mastery required i.e. the most essential. In the second relation R_E the degree of skill as measured by amount of experience with the knowledge domain or ability is given a range of values between 0 and 1. The value 0 indicates no experience needed and value 1 indicates maximum experience needed.

Following is an example of implementation of our proposal.

Suppose we have a knowledge domain that has the following relations for performance mastery R_P and exposure requirements R_E to a set of foundation skills.

| | R_P | | | | R_E | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| | K_1 | K_2 | K_3 | | K_1 | K_2 | K_3 |
| F_1 | .8 | .3 | .5 | F_1 | .7 | .3 | .6 |
| F_2 | .5 | .8 | .6 | F_2 | .5 | .5 | .3 |
| F_3 | .4 | .7 | .5 | F_3 | .4 | .7 | .8 |

Given a learner with specific values for each foundation skill

- 1) Using equation 1 and 2, a degree of similarity between the individual vector and the R_P and R_E relations is obtained.
- 2) A table of evidence for each knowledge domain is obtained and
- 3) Using equation 3 and 4 the relations are normalized.
- 4) A joint belief value is calculated by multiplying the normalized value and
- 5) The belief relation is normalized again and ordered.

The ordered values are the degree to which the learner is familiar with the knowledge area. Thus the training requirements can be determined by subtracting the value from one.

A numerical example is provided below.

Suppose we have a learner with the following set of values for knowledge:

| | | | |
|---------|----|----|-----|
| | F1 | F2 | F3 |
| R_n (| .4 | .5 | .3) |

and exposure time.

| | | | |
|---------|----|----|-----|
| | F1 | F2 | F3 |
| R_t (| .2 | .8 | .4) |

Each value in R_n represents the amount of knowledge the learner has of each foundation skill. The values in R_t represent the amount of exposure the learner has had to each foundation skill. These values can be obtained by means of a set of questions asked by the system prior to the start of the tutorial. They may also be selected from a set of values by the user.

Degrees of similarity (RS) between the learner's values and the two relations are calculated using the following formula where m is the number of Foundation Skills and t is the number of knowledge domain:

Approximate Degrees of Similarity

$$RS(R_n, R_{p,K_i}) = 1 - 1/m \sum_{j=1}^m |R_n(F_j) - R_p(F_j, K_i)| \quad \text{for } i=1 \text{ to } t. \quad \text{(Equation 1)}$$

where if $R_n(F_j) - R_p(F_j, K_i) < 0$ then $R_n(F_j) - R_p(F_j, K_i) = 0$

and

$$RS(R_t, R_{E,K_i}) = 1 - 1/m \sum_{j=1}^m |R_t(F_j) - R_E(F_j, K_i)| \quad \text{for } i=1 \text{ to } t. \quad \text{(Equation 2)}$$

where if $R_t(F_j) - R_E(F_j, K_i) < 0$ then $R_t(F_j) - R_E(F_j, K_i) = 0$

From the example above equation 1 will yield the following:

$$\begin{aligned} RS(R_n, R_{p,K_1}) &= 1 - 1/3 (|.4 - .8| + |.5 - .5| + |.3 - .4|) = .83 \\ &= 1 - 1/3 (|.4 - .3| + |.5 - .8| + |.3 - .7|) = .73 \\ &= 1 - 1/3 (|.4 - .5| + |.5 - .6| + |.3 - .5|) = .87 \end{aligned}$$

From the example above equation 2 will yield the following:

$$\begin{aligned} RS(R_t, R_{E,K_1}) &= 1 - 1/3 (|.2 - .7| + |.8 - .5| + |.4 - .4|) = .73 \\ &= 1 - 1/3 (|.2 - .3| + |.8 - .5| + |.4 - .7|) = .77 \\ &= 1 - 1/3 (|.2 - .6| + |.8 - .3| + |.4 - .8|) = .57 \end{aligned}$$

Degree of similarity calculation will yield the following evidence table:

| | e_n | e_t |
|-------|-------|-------|
| K_1 | .83 | .73 |
| K_2 | .73 | .77 |
| K_3 | .87 | .57 |

Each value is normalized according to:

$$e_n(K_i) = \frac{e_n(K_i)}{\sum_{m=1}^q e_n(k_m)}, \quad \text{for } i=1 \text{ to } q \quad \text{(Equation 3)}$$

where q is the number of rows in the degree-of-evidence table and

$$e_t(K_i) = \frac{e_t(K_i)}{\sum_{m=1}^q e_t(k_m)}, \quad \text{for } i=1 \text{ to } q \quad \text{(Equation 4)}$$

| | Normalized e_n | e_t | Multiply $e_n * e_t$ | Normalized |
|-------|---------------------|-------|-------------------------|------------|
| K_1 | .34 | .35 | .12 | .37 |
| K_2 | .30 | .37 | .11 | .34 |
| K_3 | .36 | .27 | .10 | .30 |

The normalized calculated value for each domain is the degree to which the learner is familiar with the knowledge area. Thus the training requirements can be determined by subtracting the value from one.

Use Case Example

An example of implementation of the concept can be in the area of computer hardware. For instance, in a tutorial that is designed to teach system configuration and expansion. There are number of skills and knowledge areas that are required for a successful understanding of the system expansion, such as adding a modem or a sound card. By means of a set of questions that the user can respond to the system can determine the skill level and the knowledge level of the user and determine the appropriate starting point for the tutorial.

Summary and Conclusions

Two relations between the knowledge area and the skill levels of a user are established. The matrices are manipulated to drive a set of values. The resulting values reflect the amount of familiarity of a learner with the knowledge domain. Larger the value, the more experience the learner has had thus she is more familiar. The smaller the value, the less exposure to the knowledge domain the learner has had and thus is less familiar. Based on this value the tutorial system can determine what granule of material is to be delivered. Determining these values at the start of the tutorial provides a more accurate estimation of the learner's knowledge and ability for the system thus leading to more appropriate opening interactions. Users who are engaged at an appropriate level to their knowledge and ability are more likely to continue with the tutorial. If the tutorial appears too easy or too difficult a user may discontinue the use without giving the tutorial sufficient time to produce results.

Further more, providing the system with a value that would represent the maximum knowledge level requirements can further enhance the interaction between the learner and the system. For instance, if 85% knowledge level and familiarity is set as the cut off point, then based on the values obtained from the above matrices, the system will generate and deliver a more tailored granule of material to get an individual to the desired level and then move on to the next knowledge area. Therefore, continuous evaluation will move the learner through the materials in the most expeditious and parsimonious way. This will enhance the efficiency of the tutorial system while avoiding the problems of inadequate or excessive content material presentation.

References

- Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, 42, 7-49.
- Anderson, J. R. & Pelletier, R. (1991). A development system for model-tracing tutors. *Proceedings of the International Conference on the Learning Sciences* (pp. 1-8), Evanston, IL.
- Anderson, J. R., & Reiser, B. J. (1985). The Lisp tutor. *Byte*, 10 (4), 159-175.
- Burns, H. L. & Capps, C. G. (1988). Foundations of intelligent tutoring systems: An introduction. *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Chandrasekaran, B. (1986). Generic tasks in knowledge-based reasoning: high level building blocks for expert system design. *IEEE Expert*, 1 (3), 23-30.
- El-Sheik, E., & Sticklen, J. (2003). Using hierarchical classification-based expert systems to support tutoring. *Proceedings of the International Conference on Artificial Intelligence*, Vol 2, p 636 – 642.

Approximate Degrees of Similarity

- Katz, S., Lesgold, A., Eggan, G., & Greenberg, L. (1996). Towards the design of more effective advisor for learning-by-doing systems. *Proceedings, Third International Conference, ITS '96*. Montreal, Canada, June 12-14.
- McArthur, D., & Stasz, C. (1990). An intelligent tutor for basic algebra. R-3811- NSF, RAND Corporation, Santa Monica, CA.
- McArthur, D., Lewis, M., & Bishay, M. (1994). The role of artificial intelligence in education: Current progress and future prospects. RAND Corporation, Santa Monica, CA.
- Nussbaum, M., Rosas, R., Peirano, I. & Cardenas, F., (2001). Development of intelligent tutoring systems using knowledge structures. *Computer & Education*, 36, 15-32.
- Person, N. K., Graesser, A. C., Kreuz, R. J., & Pomeroy, V. (2001). Simulating human tutor dialog moves in AutoTutor. *International Journal of Artificial Intelligence in Education*, 12.
- Regian, J. W., & Shute, V. J. (1992). Automated instruction as an approach to individualization. In J. W. Regian & V. J. Shute (Eds), *Cognitive approaches to automated instruction* (pp. 1-13). Hillsdale, New Jersey.
- Self, J. A. (1990). Bypassing the intractable problem of student modeling. In C. Frasson & G. Gauthier (Eds), *Intelligent tutoring systems: At the crossroad of artificial intelligence and education* (pp.107-123). Norwood, New Jersey
- Rothenhofer, D., & Herzog, C. (1993). SYPROS -- An intelligent tutoring system for parallel programming. In T. W. Chan (Ed.), *Proceedings of ICCE93* (pp. 300—305), Taiwan.
- VanLehn, K. (1986). Student modeling in intelligent systems. *Proceedings of the Research Planning Forum for Intelligent Tutoring Systems*. San Antonio, TX: Air Force Human Resources Laboratory.

Biography



Namdar Mogharreban received his PhD in Computer Based Education in 1989. He has taught computing courses in Management Department, Technical School and most recently in a Department of Computer Science. The primary area of research interest is end-user computing user training and application development.