CHAPTER 12 –The Need for a Mathematical Model of Intelligent Tutoring

Robby Robson[1], Xiangen Hu[2], Donald M. Morrison[2], Zhiqiang Cai[2]
[1]Eduworks Corporation; [2]University of Memphis

Introduction

ITSs are a large family of technologies that attempt to replicate the experience and learning gains derived from one-on-one human tutoring (for recent reviews, see Durlach & Ray, 2011; Graesser, Conley & Olney, 2012). While these systems are notoriously diverse in their functionality and construction, several authors have created abstract models of ITSs that attempt to capture common components, generally accepted to be the learner model (sometimes called the “student model”), the domain model, the tutor model, and the user interface (Durlach & Ray, 2011, Elson-Cook, 1993; Graesser et al., 2012; Nkambou, Mizoguchi & Bourdeau, 2010; Psotka, Massey & Mutter, 1988; Sleeman & Brown 1982; VanLehn, 2006; Woolf, 2008). ITS have also been classified into categories such as model-tracing, example-tracing, constraint-based, and dialogue-based tutors. Finally, VanLehn has observed that in describing an ITS, it is useful to distinguish between an outer loop of a tutor, which focuses on task selection and macro-adaptation, and the inner loop which handles in-task interactions and micro-adaptation (VanLehn, 2006).

Gaps

These descriptions, however, are largely *qualitative* descriptions that help researchers understand how a given tutor works, but do not help answer *quantitative* questions about how well they work and ultimately how they can be improved. In this chapter, we consider what type of model is needed for answering quantitative questions such as the following:

- What is the gap between the behaviors exhibited by a student and by an expert?
- At what rate is an ITS closing the gap?
- How accurately does an ITS assess knowledge, skills, affective states, and other attributes of a learner?
- Among all possible strategies that an ITS has available, which are likely to lead to the most learning gains in a given situation?
- Given two strategies, which leads to desired learning outcomes in the least amount of time?
- Among the algorithms used by the inner and outer loops of a tutor, where will improvements have the greatest impact?

Most of these questions require measuring two basic quantities about an ITS: Its effectiveness in helping learners achieve learning goals and how efficiently it helps students achieve them. To date, effectiveness is usually measured in terms of the effect size (Cohen’s *d*) of learning gains. While aggregate, summative measures such as these are useful in determining how well a tutor is working, they do not tell us what caused the learning, what is happening as learners use the tutor, or how we can improve the tutor. Efficiency, if it is measured at all, is measured by comparing how long it takes to learn concepts or behaviors using an ITS to how long it takes to learn them using other instructional means. Once again, this does not help us make improvements because we don't know what in the tutor causes faster or slower learning, and it does not give us any information about an ideal efficiency against which we can measure how well a tutor is working. To answer these questions, we need models that enable us to measure learner progress and learner rate of progress towards a goal as the tutor operates. The word "measure" is key here and implies quantification.

Exploration of the General Model

Two Analogies

Before discussing the quantification problem, we present two analogies that may help explain why a formal mathematical model is so important. The first is mechanical and views the problem of tutoring a student as analogous to using an autopilot to guide an airplane. As with an ITS, an autopilot is a machine substitute for a human. Its goal is to steer an aircraft to a destination, much as the goal of an ITS is to guide a learner to a learning objective. The autopilot operates using real-time data on position, atmospheric conditions, trim of the aircraft, and other factors and attempts to follow a flight path, which, in many cases, has been calculated to maximize efficiency given constraints such as atmospheric conditions, terrain, the flight paths of other aircraft, the cost of flying at different altitudes, and the ultimate destination. It controls the path by adjusting the ailerons, elevators, rudder, and thrust.

We can think of the data that defines the current state of an aircraft as analogous to the learner model that defines the current state of the learner. But whereas an autopilot continually measures this state and follows a path, ITSs typically measure only the initial and end states, or possibly a small number of intermediate states. This is not sufficient to quantify the dynamics of an ITS or create the analog of a guidance system, which is what we want.

Of course, a learner is not a machine, and an autopilot does not teach a plane to fly. A better analogy, in some ways, is that of a doctor treating a patient. Unlike the autopilot, the doctor, even with modern imaging technologies, cannot observe the precise state of a system as complex as the human body. At best the doctor can talk to the patient, take the patient's history (or look it up), conduct a physical examination, if necessary order some lab tests, and then, using knowledge of the relationship between clusters of symptoms and disease processes, make some guess about what is likely going on. If the doctor finds that the patient is unhealthy, the goal is to find a course of treatment, which, if followed, will return the patient to a state of health, but since the doctor cannot know the actual state of the patient, in practice, the doctor can, at best, monitor the patient's symptoms, with the expectation they will go away. If this is achieved, the patient is presumed to be healthy.

Although an ITS operates more like an autopilot in the sense that it uses a guidance system to steer learners towards a goal, it operates more like a doctor in that it cannot measure the precise state of learner and must rely on models of generic learners to interpret the measurements made of any specific individual. The ITS must therefore rely on a set of measurements and models of how humans learn to infer the state of the learner and prescribe interventions that change the state to a desired one, usually the state of an "expert." However, to discover which interventions should be prescribed, the ITS must have a means to observe how (and how much) the inferred learner state changes in response to specific interventions. This requires a well-defined mapping between what an ITS can measure and a model of the learner state. In today's practice, it is often difficult to specify what measurements an ITS is taking and how those are being translated into a model whose distance from an ideal model can be measured.

A Conceptual Model for Tutoring

Although ITS developers employ different approaches to the problem of guiding human learning, there is a great deal of similarity in how their systems function. Conceptually, each ITS uses interactions between the student and the system (verbal, written, haptic, biometric, etc.) to infer and alter what the learner knows or can do. The ITS observes these interactions and translates them into machine-readable data. These data, and possibly other data that are known to the ITS, are used to estimate the learner's current state of knowledge and skill. Knowledge states are typically represented as mastery levels of concepts or skills, knowledge components (Koedinger, Corbett & Perfetti, 2012), or a similar set of parameters. In addition to using estimates derived from interaction data, some ITSs use the structure of the knowledge domain and history of learner's interactions to make inferences about knowledge states, e.g., by inferring that if a learner has demonstrated mastery of concept C or successfully performed task T several times, then the learner has also mastered concept D and can perform task U. In some instances, affective states are estimated by the ITS in analogous ways. Once the state of the learner is estimated, the ITS uses this state to determine what interaction or interactions will next take place.

It is useful to separate the foregoing description of an ITS into two parts:

1. *State modeling and estimation*: Through the use of data collection devices and strategies such as emotion sensors, observation of game performance, responses to test questions, and direct questions (in the case of dialog-based systems), the system estimates where the learner currently lies in a multidimensional model of possible cognitive and affective states.

2. *Evaluation and decision making*: Given the model of possible states and the estimate of the learner's current state, the system decides that either (1) the learner's state is optimal with respect to normative expectations, in which case it moves to the next step in the outer loop; or (2) the learner's state is suboptimal, in which case it selects and enacts what it considers to be the best intervention in the inner loop.

State modeling and estimation is the general form of learner modeling, and the second step is the functional view of how an ITS uses expert, domain, and pedagogical models to direct its operation based on a state model. The cycle of estimation, evaluation, and decision making is repeated until the outer loop is exited.

Ultimately, we want to evaluate how well the tutor is working and how to improve it by improving its ability to estimate, evaluate, and change the learner's state. For this purpose, a slightly more formal formulation is needed and can be given as follows:

1. A learner model (the learner's present state) can be represented by a (finite) set of state variables. Each ITS represents a learner's state by the values of these variables as they vary over time. In more formal terms, there is a space S that represents all possible states of a learner. These states can ostensibly include motivational, affective, cognitive, and social factors ranging from mastery levels of domain concepts to frustration and motivation levels and certain individual or cultural beliefs, which might affect how a learner approaches a task (e.g., see Arroyo et al., 2009; D'Mello & Graesser, 2010). In the autopilot analogy, S is the position of the aircraft and in the patient analogy, S is the actual state of the patient (which can only be inferred and not directly measured).

2. The state of a learner at any time t is estimated based on a set O of observable variables. These are obtained through interactions with the ITS or through data communicated to the ITS from other systems, e.g., from a LMS or a game operating in a multi-system framework such as GIFT

(Sottilare, 2012). The ITS contains an algorithm a that maps the history of observations about a learner to the learner's current state. This is a function $a{:}O \times T \rightarrow S$, where T is the time interval during which observations are available. The variables in O can be thought of as the symptoms and measurements taken by the doctor, and a is the process the doctor uses to infer the state of the patient. (Note: The function a may, in practice, use past values of a in computing the present value of a. In other words, the history of estimates of the learner's state may be used to estimate the current state.)

3. Within S there are target states, and the goal of the ITS is to move the learner's current state to a target state. In the analogies, these are the destination of the aircraft and the state of health of the patient, which is a range of states. In an ITS, these may be defined by an expert model or the expected behavior of a learner at a particular developmental stage.

4. The ITS functions by doing the following:

    a. Interacting with the learner.

    b. Measuring the values of variables in O.

    c. Applying a to estimate the learner's state in S.

    d. Selecting a strategy and associated actions that will move the learner to the target state.

    e. Implementing those actions through (and only through) a set of interactions with the learner.

Observations about this Model and Questions Raised

All Tutors Trace a Learner Model: Our first observation is that this model is an abstraction of model-tracing tutors but applies to almost all ITSs. For example, the constraint-based tutors described by (Mitrovic, Mayo, Suraweera & Martin, 2001) analyze student answers against a set of constraints to determine the student state and take appropriate actions. The example-tracing tutors of (Aleven, Mclaren, Sewall & Koedinger, 2009) compare student input to correct and incorrect problem-solving behaviors. These constraints and reference examples define points in S that the tutor tries to steer toward or away from. Autotutor Lite (Hu et al., 2009) constructs a "learner's characteristic curve" based on semantic comparisons of student input to text that represents expected answers (Robson & Ray, 2012) calculated over a series of turns in the inner cycle. Even if an ITS does not have an explicit expert model or domain model, it computes some set of state variables and takes actions to move the learner state to a desired state. *Conceptually, every ITS is a"learner model tracing" tutor*. Even if different tutors estimate the learner state in different ways and take different actions to change the state, the existence of a parameterized state space makes it feasible to quantify and compare tutors on the basis of how learners move through this space.

Optimal Paths: Our second observation is that the ideal ITS moves a learner along a path in S that minimizes "cost" (e.g., time to mastery or actual cost of running a simulation) and maximizes "benefit" (e.g., how close the learner is to the target state and how long the learner will retain that position.) A good test of whether a particular S (i.e., a particular set of parameters used to model the learner state) is viable is whether its properties allow for the optimization of paths between any two states with respect to a suitable utility function that reflects cost and benefits. This raises the question of whether existing tutors have either explicit or implicit state models that are sufficient to do this. In most cases, the answer is likely no, and we see this as a limiting factor to making progress in the area.

We observe that most existing tutors focus on cognitive (and possibly affective) state variables that describe learning goals. These state variables are often represented as levels of competency with respect to a set of objectives, knowledge components, or similar constructs, and are implicitly considered to be observable (with error) through assessment. In reality, the situation is more complex, more akin to the patient analogy than the aircraft analogy.

If we picture S as a higher dimensional object, then the typical tutor works in a projection of S onto a discrete structure (see Figure 12-1). This results in loss of information and makes it impossible to measure the distance between two learner states because there are many points in S that correspond to each point in the discrete structure. Moreover, the observability assumption means that, as far as the ITS is concerned learners never stop between observable states. For example, an ITS might observe through assessment that a learner knows or does not know a fact but cannot observe where the learner is in the process of learning the fact. No intermediate state between "not knowing" and "knowing" exists. Even if the discrete structure could be used to estimate distances between states in S, this further loss of information makes it hard to determine the learner's path through S. ITS authors can program the inner and outer loops of a tutor, and they can empirically determine the effects of this programming on learning outcomes, but they cannot measure the effects on a utility function and cannot determine whether the tutor is close to ideal.

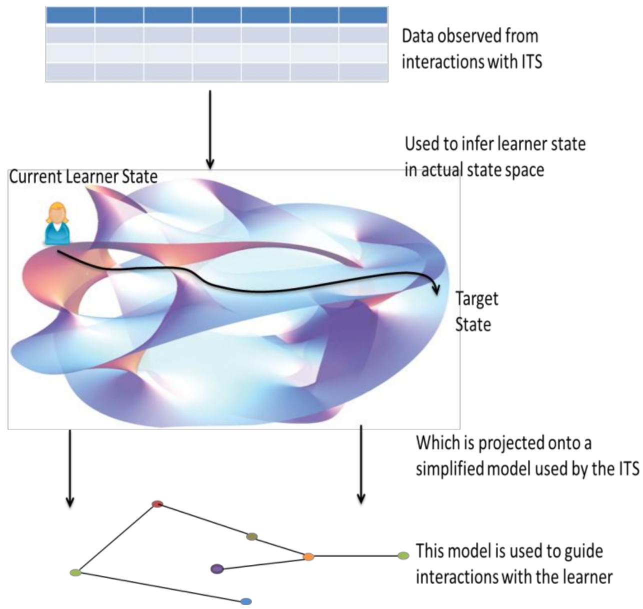

Figure 12-1. How an ITS works*

Universal Components: An obvious question, whose answer has many implications, is whether there is a universal state model. In other words, is there a set of state variables that can be used across multiple tutors? Even if a tutor has a different state space, could it be reduced to a common space without suffering a significant loss in effectiveness? Or, phrased differently, is there universal learner model that can and should be programmed into a framework such as GIFT? As argued in Robson & Barr (Chapter 2 in this volume), this is a key question, and it has been asked in various forms by Durlach (2012), Goldberg, Holden, Brawner & Sottilare (2011), and others.

It is doubtful that a universal state space exists if one includes the cognitive dimension since concepts and knowledge constructs differ from domain to domain. Even should an "ontology of everything" be achievable, it may not be practical to maintain this for all systems at all times. However, if we separate these parameters into a domain model (as is done in GIFT), then it is reasonable to ask whether there is a formulation of domain models that can be used as a template for any S and, more importantly, whether motivational, affective and social parameters are sufficiently domain-independent to allow for a manageable set of associated state variables that can be effectively used across most ITSs.

We do not know the answer to these questions, but we observe that affective, motivational, and social components are increasingly being inferred from sensor data and in-system responses (Arroyo et al.,

* Surface image from http://en.wikipedia.org/wiki/File:Calabi_yau.jpg.

2009; Calvo & D'Mello, 2010; D'Mello & Graesser, 2010; Robison, McQuiggan & Lester, 2009; Woolf et al., 2009). Since target states in S are usually defined in the cognitive or behavioral domain, these components are used primarily as control variables. A universal representation of these components could significantly improve the ability of the ITS to use them to effectively guide learning along optimal paths.

Estimation: The algorithm a maps ITS observations to learner states. In real-world examples, a operates on data ranging from simulation and game data (Engineering & Computer Simulations, 2013) to data generated by LSA (Wiemer-Hastings, Graesser & Harter, 1998) and by using production rules to analyze student answers (Blessing, Gilbert, Ourada & Ritter, 2009). The challenge of transforming observables into a state model may be equal to (or greater than) the challenge of determining the state model and is just as critical. The ability of an ITS to follow an optimal path is limited by its ability to detect learner state.

Intuitively, an ITS that continually tracks observable state changes along numerous dimensions, including multiple micro-adaptations, has a better chance at accurately estimating the learner's state than an ITS which relies on static models, discrete measurements, and only on macro-adaptations. However, a relatively small number of observables may be sufficient to account for most of the variance in learner state. This has been observed in affect detection (Graesser, Rus, D'Mello & Jackson, 2008) and is a fertile area of research that can be supported by GIFT.

Intervention Selection: Our final observation is that if we understood how different interactions affected the trajectory of the learner in S, it would seem relatively straightforward to design algorithms for selecting the best interactions. In other words, assuming we can estimate and evaluate reasonably, empirical experimentation can be used to come up with a potential set of interventions for which selection should be straightforward. Frameworks such as GIFT that can be used to integrate disparate types of interventions are ideal for developing this type of understanding, assuming we a reasonable model of S and estimation function a.

Future Research and Recommendations for GIFT

To be of general significance, the model presented in this chapter must be more precisely formulated and tested for its ability to model and provide useful insight into existing ITSs. This requires understanding what motivational, affective, cognitive, and social environment data is likely to be represented in S, which is perhaps the central problem addressed in this volume. As stated in (Goldberg et al., 2011:10), "*While we intuitively know that it is better to have more information when we are making decisions to tailor instructional feedback and content to individual trainee needs, the influence of specific trainee attributes on instructional decisions can be debated. Additional experimentation is needed to quantify the impact of trainee attributes.*"

This chapter suggests that as this central question is addressed, it will be important to observe the paths traced through the learner state data collected by GIFT and not just the data themselves, and that it will be important to test whether the parameters in a learner model can be used to answer optimization questions. In addition, this chapter points out that GIFT can be used to empirically evaluate the effects of individual interventions, probably at a more granular level than the typical complete ITS, and that a lot of thought should be given to observables. Experiments that investigate what data is required to sufficiently determine affective states provide good models for analyzing data coming out of simulations, games, and sensors with regard to their ability to determine the parameters in S.

References

Aleven, V., Mclaren, B. M., Sewall, J. & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 105-154.

Anderson, J.R. (1988). The expert model. In Polson, M. C. & Richardson, J. J. (Eds.). *Foundations of intelligent tutoring systems*. Lawrence Erlbaum. 21-53.

Blessing, S. B., Gilbert, S. B., Ourada, S. & Ritter, S. (2009). Authoring model-tracing cognitive tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 189-210.

Durlach, P. J. (2012). Vanilla, Chocolate, or Chunky Monkey: Flavors of Adaptation in Instructional Technology. *iFest 2012*, from http://www.adlnet.gov/wp-content/uploads/2012/08/Durlach_Adaption_in_IT_iFest-2012.pdf

Durlach, P. J. & Ray, J. M. (2011). Designing adaptive instructional environments: Insights from empirical evidence *Army Research Institute Report*. Arlington, VA.

Engineering & Computer Simulations. (2013). Student Information Models for Integrated Learning Environment (SIMILE) Retrieved February, 2013, from http://www.ecsorl.com/solutions/simile

Goldberg, B. S., Holden, H. K., Brawner, K. W. & Sottilare, R. A. (2011). *Enhancing Performance through Pedagogy and Feedback: Domain Considerations for ITSs*. Paper presented at the Interservice Interindustry Simulation Education and Training, Orlando, FL. https://litelab.arl.army.mil/system/files/IITSEC2011_Goldberg_etal_Enhancing%20Performance%20Through%20Pedagogy%20and%20Feedback.pdf

Graesser, A. C., Rus, V., D'Mello, S. K. & Jackson, G. T. (2008). AutoTutor: Learning through natural language dialogue that adapts to the cognitive and affective states of the learner. In D. H. Robinson & G. Schraw (Eds.), *Recent innovations in educational technology that facilitate student learning* . pp. 95-125. Information Age Publishing.

Hu, X., Cai, Z., Han, L., Craig, S. D., Wang, T. & Graesser, A. C. (2009). *AutoTutor Lite*.

Mitrovic, A., Mayo, M., Suraweera, P. & Martin, B. (2001). Constraint-based tutors: a success story. *Engineering of Intelligent Systems*, 931-940.

Robson, R. & Ray, F. (2012). *Applying Semantic Analysis to Training, Education, and Immersive Learning*. Paper presented at the Interservice/Industry Training, Simulation, and Education Conference, Orlando, FL.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). *The Generalized Intelligent Framework for Tutoring (GIFT)*. Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).

VanLehn, K. (2006). The behavior of tutoring systems. *International journal of artificial intelligence in education*, *16*(3), 227-265.

Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Burlington, MA: Morgan Kaufmann.