Chapter 1

Introduction

Computers have been used in education since 1960 (O'Shea & Self, 1983). The purpose of using computers in education systems is to help students learn more efficiently. Traditional education systems instructing via computers are called Computer Assisted Instruction (CAI) systems. CAI systems presented the material to the student in a static "storyboard" fashion, where every student received the same material, although they may have had some control over how they navigated through the curriculum. While traditional CAI systems may be somewhat effective in helping learners, they are restrictive in that they do not consider the diversity of students' knowledge states and their particular needs (Brusilovsky & Maybury, 2002; Yao & Yao, 2003). Such systems do not generate flexible instructional plans. Instead, they follow a prespecified and fixed plan. Moreover, CAI systems are not adaptive and cannot dynamically provide the same kind of individualized attention that students would receive from human teachers (Bennett, 1999). Later systems included "branching", where the system's response to a student's answer differed according to what their response was. However, because these CAI systems lacked any knowledge of the domain being taught, specific feedback was difficult, because it had to be handcrafted for each problem. As a consequence, the system's response was often limited to indications of right/wrong or presentation of the correct answer, and so the problems posed usually required only yes/no, multi-choice or short (e.g. numeric) answers.

This drawback has prompted a promising direction in the application of Artificial Intelligence techniques in education known as Intelligent Tutoring Systems (Burns & Capps,
Intelligent Tutoring Systems are computer-based programs that present educational materials in a flexible and personalized way which is similar to one-to-one tutoring (Brusilovsky, 1999). In particular, Intelligent Tutoring Systems have the ability to provide learners with tailored instructions and feedback. The basic underlying idea of Intelligent Tutoring Systems is to realize that each student is unique. If the computer could be programmed to *emulate* a human tutor (teacher), then one-to-one tutoring for all students would become a possibility.

To emulate a teacher, the computer would have to be programmed with domain knowledge and pedagogy. Natural Language Processing (NLP) would be used to communicate with the student in a similar way in which a student communicates with a teacher. However, this initial approach of emulating a human teacher has proven to be an overly ambitious goal. Firstly, NLP is yet to be sufficiently effective to support a natural language conversation between a student and a tutor. Secondly, it has been realized that a computer program can never replace a human tutor. There will always be students for whom the software is not suitable. The role of computers in teaching has, therefore, been just complementing the teacher. Intelligent Tutoring System can be *used* as a complement to classroom teaching in that students are able to learn at their own pace. By using Intelligent Tutoring Systems as support tools for learning, teachers will have more time to focus on one-to-one tutoring for students who are struggling.

In this chapter, an overview of this thesis is presented. In section 1.1, the major components of all the Intelligent Tutoring Systems are presented and explained. Various problems associated with the current Intelligent Tutoring Systems are pointed out in section 1.2. The scope of this work is summarized in section 1.3. In section 1.4, demonstration of BiTutor is explained, and the guide to the thesis is provided in section 1.5.
1.1 Components of Intelligent Tutoring System

Early CAI systems were not modular (Woo, 1991). This unfavorable structure caused problems when a system required modification, and it was sometimes necessary to restructure the whole system. Researchers typically separate an Intelligent Tutoring System into several different parts, and each part plays an individual function. Beck et al. (1997) have identified five major components: student model, knowledge domain, expert model, pedagogical model, and user interface, as illustrated in Figure 1.1.

![Diagram of Intelligent Tutoring System Components](image)

Fig 1.1: The major components of Intelligent Tutoring Systems.

**Student Model:** The student model stores information that is specific to each individual learner and enables the system to identify different users. Usually, this information reflects the system’s understanding of learner’s current knowledge state. Thus, the student model can track a student’s understanding and particular need. Without an explicit student model, the teaching strategies component is unable to make decisions to adapt instructional content and guidance and is forced to treat all students similarly. Other factors such as motivation can also be modeled, though this is done less frequently.

**Knowledge Domain:** The knowledge domain stores learning materials which the students are required to study for the topic or curriculum being taught, and it is an important component since without it, there would be nothing to teach to the student. The quality of
the domain knowledge is entirely variable; it can range from expert system-quality in which
enough knowledge is represented to enable the Intelligent Tutoring System to solve
problems itself, through a subset of the knowledge that is just enough for the purpose of
teaching. The domain model represents the course being taught in such a way that the
system can use it for reasoning. There are many possible representations, including
semantic networks, production rules and constraints. What representation is adopted
depends partly on how it will be used. It supports other functions such as information
selection and representation, problem selection, and feedback generation. Mitrovic et al.
(1996) point out that in an ideal case, if an Intelligent Tutoring System is required for a
different domain, the domain knowledge model is the only component of the system that
needs to be changed.

**Expert Model:** The expert model uses the domain knowledge to advise other parts of
the system. It may indicate the relative difficulty of curriculum sections or problems, such
that the pedagogical model can select the next task. The expert model is able to provide the
source for the knowledge in explaining student's responses, and serve as a standard for
evaluating student's performance. In cognitive tutors, it identifies whether or not the
student's current solution is on track and, if not, what has gone wrong? It may also be able
to run the domain model to solve the problem from a given state.

**Pedagogical Model:** The pedagogical model decides what to present to the learner. It
uses information from the student, domain and expert models to arrive at each decision. In
effect, this module models the "teaching style" to be applied.

The pedagogical model component refers to instructional techniques for teaching. It is
the component that makes decisions about the teaching of the domain. Pedagogical
decisions include, but are certainly not limited to, next problem selection, and next topic
selection, adaptive presentation of error messages, and selective highlighting or hiding of text. Typically, this component will take into account both the current student model and the domain knowledge when making a decision. The assessment result of the student model is the input to this component, so the system’s pedagogical decisions reflect differing needs of students. Thus, this component needs to take appropriate actions to manage one-on-one tutoring, such as switching teaching strategies and using a variety of teaching approaches at the appropriate time according to the student’s particular needs and problems.

**User Interface:** The user interface decides how the system interacts with a user. The focus of this component is on how the material should be presented to the student in the most effective way. The interface should be able to provide the motivation for the student to continue. If the student lacks the desire to use the system, the Intelligent Tutoring System will not be effective. Then, the interface can improve learning significantly by reducing the cognitive load. If only one component of a problem is the focus of teaching, then the rest of the problem can be “externally stored” within the user interface. This means the student does not need to remember extraneous details and can instead target the component of problem solving that is of importance.

Although all the components of Intelligent Tutoring Systems are important, it should be clearly obvious that the student model is the most critically important. If the student model is not able to even approximately describe the traits of the current student, then the quality of decisions made will not be good. This is regardless of the quality of the tutoring strategy itself. Considerable research, therefore, has been invested specifically in student modeling.
1.2 The Problems in Intelligent Tutoring System

In this section, the central problems, particularly, the types and effects of the uncertainty inherent in an Intelligent Tutoring System and its consequences are discussed in detail. This section also explains the problems associated with item selection in an Intelligent Tutoring System.

1.2.1 Uncertainty

Uncertainty is one of the significant difficulties associated with Intelligent Tutoring System as it models users or students. Intelligent Tutoring System faces the great challenge in getting information about the student, due to the restricted observation of student's behavior only through computer keyboard and mouse. So, there is a need to build a student model from the available minimal amounts (relative to the human tutor) of highly uncertain information obtained from the students. Furthermore, as the Intelligent Tutoring System bases its decisions (like a human tutor) on the student model, poorly adapted teaching actions may be realized because of the uncertainty in the student model being carried over.

It is this uncertainty that has led to the development of other, "simpler" Intelligent Tutoring Systems metaphors, such as that of the discovery environment, which require less computational effort and inference for (its proponents claim) the same degree of benefit.

It is relevant to examine the types of uncertainty inherent in student modeling. A student model is constructed based on the observations that the Intelligent Tutoring System makes about the student. Observations can come in the form of responses to questions, answers to problems, traces of the student's problem-solving behavior, etc. These observations are processed to get the student model. The student model can in fact be thought of as a compression of these observations: the raw data is combined, some of it may
be discarded, and the end result is a summarization in the form of a set of beliefs about the student. This processing on observations is usually defined by a set of inference rules mapping observations to beliefs. However, there are two potential sources of error in this process.

Firstly, the amount of raw data available and observations made may be insufficient to draw strong conclusions. If any data cannot be shown to lead to statistically significant hypotheses, one could argue that the data is insufficient. Of course, an Intelligent Tutoring System often does not have sufficient time to acquire enough data to hypotheses about the student’s state to this degree, especially given that the state of the student can be expected to change rapidly.

Secondly, the inference rules for building the student model may themselves be sub-optimal. Poor inference rules will lead to a poor student model regardless of how reliable or unreliable the data acquired from the student is. In other words, if the inference rules are inconsistent, incomplete or semantically inexplicable, then the quality of the data will have a reduced bearing on the quality of the resulting student model. But reliable data combined with quality inference mechanisms will lead to a quality student model.

Of course, it is not a simple task to determine the quality of data and inference rules. It can never be guaranteed that the current state of the student is truly represented by the data. The data observed may just be random noise. Thus, building the student model is a highly uncertain activity.

There is another class of uncertainty in this process. The first type of uncertainty arises from the construction of the student model from the data, whereas, the second type arises from the fact that some teaching actions, such as item or topic selection, are selected on the basis of the student model. These actions, therefore, can be thought of as functions of the
student model. If the student model is uncertain then clearly this uncertainty will transfer to the action selection functions. It will be evident in the form of actions being selected that are not optimally adapted to the student. This makes the adaptive aspect of Intelligent Tutoring Systems fail. But, if the student model is of a relatively high quality and the action selection functions are insensitive to small amounts of uncertainty, then it may well be the case that the selected actions remain optimally near-optimally rational. It is difficult to determine the degree of rationality of the action selection function, especially when the rules defining the function are not based on theory. In a hypothetical worst case, the action selection functions could make random decisions. Typically, however, the action selection function will be some heuristic that considers the values in the student model, though such an approach cannot guarantee that this is the best possible rule. This type of uncertainty, as well as the uncertainty inherent in the student model, can contribute to the overall sub-optimal behavior in the Intelligent Tutoring Systems.

1.2.2 Ineffective Use of Test Items

Assessing student's knowledge mastery is a necessary part of tutoring. In most of the Intelligent Tutoring Systems, item selection has not been the focus. Most of Intelligent Tutoring Systems focus on the student's responses for the items posted. This may lead to a student with high level of knowledge mastery getting an easy item and a student with low level of knowledge mastery getting a difficult item. This again fails the adaptive nature of Intelligent Tutoring Systems. Items should be selected in such a way that its difficulty level is close to the student's mastery level. In other words, the item selected should be answerable with the knowledge the student has gained so far.
1.3 Scope of the Work

The question that arises is how to minimize uncertainty. It is very difficult, if not impossible, to eliminate uncertainty of the first type. While the inference rules for building the student model may be optimized (and there is much research in this area), there is no way to guarantee the quality of the data. Uncertainty of the second type (which occurs because of sub-optimal action selection), however, presents a different challenge. Powerful general theories of decision-making, designed specifically for situations involving uncertainty, have been developed. One of them is Bayesian probability theory (Bayes, 1763), which deals with uncertain reasoning, and the other is statistical decision theory (Savage, 1954) which extends Bayesian probability to decide and incorporate a measure of preference for the outcome of actions called utility. The other problem associated with Intelligent Tutoring Systems, namely, the ineffective use of test items, can be addressed by the usage of item-response theory. This research focuses on representation of the knowledge structure as Bayesian networks, decision-theoretic approach to computerized tutoring and the application of item-response theory for item selection. The scope of the research is presented in the following subsections.

1.3.1 Representing Knowledge Structure as Bayesian Networks

From the expert systems, it is evident that knowledge acquisition and representation are major tasks in any intelligent system (Giarratano & Riley, 1998). In this research, the knowledge component available in the course is to be represented in the form of topics, subtopics, and concepts, known as knowledge items. Each knowledge item represents what is to be taught to the student and at which level of the cognitive domain (McCormick & Pressley, 1997). Each knowledge item can be represented as a node in Bayesian network.
The relationship among variables in the knowledge structure can be represented using the conditional dependencies of the variables in the Bayesian Network (BN) (Pearl, 1988). In this research, the approach of using the knowledge items and their causal relationships to develop a set of Bayesian networks has been adopted. Such network is an efficient representation of hierarchical information, where inclusion of new information only affects local conditional relationships. Bayesian networks provide easier mechanism for updating the knowledge structure. More information about knowledge representation using Bayesian networks and its construction will be presented in detail in Chapter 3.

1.3.2 Decision-theoretic Approach to Computerized Tutoring

Probability theory describes what an agent should believe on the basis of evidence; utility theory describes what an agent wants, and decision theory puts the two together to describe what an agent should do (Russell & Norvig, 2003). Decision analysis provides a philosophy that emphasizes the insights that can be gained by decision maker who goes through the decision analytical processes. For example, the decision analytic approach highlights the distinction between a good decision and a good outcome. Heckerman (1991) defines a good decision as one that is consistent with the preferences and complete information of decision maker; while a good outcome is desirable.

The aim of the decision-theoretic approach in a computerized tutoring system is to provide optimal action selection which maximizes student learning and is defensible. Pedagogy can be incorporated into decision analysis to measure learning value, besides taking into consideration uncertainty in student’s knowledge mastery. In this way, one can be sure that the correct standard for tutoring decisions can be achieved. After an action is decided, other consequential actions, such as, which topic to be presented next and which item to be used can be determined. The utility functions are formulated according to the
student's behavior for each tutoring action. More information about the decision-theoretic approach is presented in Chapter 4.

1.3.3 Application of Item-Response Theory for Item Selection

Assessment has always been a very important step in the learning process. In most of the Intelligent Tutoring Systems, assessment is done by posting items to the students and tracking their responses to these items. These are usually in multiple-choice format and the questions can be designed such that each item tests one knowledge item. To make the system adaptive, it is necessary to rank the items according to their difficulty and also the statistical measurement of student's mastery level is needed. This can be done by using the concepts of item-response theory where each item is defined using a set of parameters. These item parameters can be estimated based on student's responses using item calibration methods such as maximum likelihood method, Markov Chain Monte Carlo (Carlin & Chib, 1999; Gilks et al., 1996), Kernel Smoothing (Ramsay, 1991) or Method of least squares.

Each item is associated with certain feedback. If a student gives a wrong answer, corresponding feedback should be provided to the student. With these features, the student model can be constantly updated with student's mastery level and feedback on his misconceptions. Item selection using item-response theory is presented in Chapter 4.

1.3.4 Generation of Tutoring Strategy

Student's learning is seldom completed with one tutoring action. A tutoring strategy or policy consists of a planned sequence of actions to guide student learning. The system selects the best tutoring action based on student's current mastery state. The response affects the student's next mastery state. The goal is to determine actions that seek to maximize information about the student's misconceptions or faulty knowledge. This sequence of
actions leads to the shortest possible time. These actions tell the tutor what to do for any state that the student might be in.

Before taking actions, the complete information on the variables in the student model is not available and so the decision-theoretic tutoring is a Partially Observable Markov Decision Problem (PODMP). In PODMP (Lane, 1989; Cassandra et al., 1994; Russell & Norvig, 2003), decisions are made by projecting forward a sequence of possible actions and choosing the best one. Decision making using this tutoring strategy is explained in Chapter 4.

1.4 Demonstration of Bayesian Intelligent Tutoring System

The goal of this research is to develop an Intelligent Tutoring System using Bayesian network and its features. BiTutor (Bayesian Intelligent Tutoring System) is an Intelligent Tutoring System developed to emulate what a teacher will do to help students in their learning. BiTutor is developed for the computer science students to teach them different courses in computer science. Currently only the Elementary Data Structure course is added to BiTutor. The overview of BiTutor is as follows: BiTutor is used by two types of users: student and staff.

A student uses BiTutor to learn and improve his knowledge in a particular course. When a student wants to use BiTutor, he/she must be a registered student to use it. In case, if the student has not yet registered, he/she has to undergo the registration process with BiTutor. After successful login, the list of available courses is shown to the student. The student can select any course that he wants to master. Then, the list of topics under the selected course is shown and the student selects the topic that he likes to learn. When the student selects a topic to learn, BiTutor checks the student profile to find out if the student
has already visited this topic. Based on the student profile, BiTutor will initialize the Bayesian network related to that topic. Now, BiTutor is ready to tutor the student in that topic. During the tutoring process, BiTutor selects the suitable tutoring action like providing the tutorial, testing the student in a particular concept, giving feedback when the student gives wrong response, etc., based on the student's knowledge. In the case of testing the student in a particular concept, the multiple-choice item is selected from the item bank and is posted to the student. The options of the item are designed in such a way that only one option will be the correct one and all the other options will be meaningful wrong options. As a result each wrong option can be associated with a suitable feedback. The student's response is then observed. If he gives a wrong response, the suitable feedback is shown to him. Based on the student's response, BiTutor updates its belief on the student's mastery knowledge. BiTutor continues to tutor the student until termination criteria is met.

The second type of user of BiTutor is staff. BiTutor has three types of staff, namely, administrative staff, authorized staff, and normal staff. The administrative staff have full control over BiTutor. The main tasks of administrative staff are giving authorization for other staff, creating courses and assigning them to authorized staff, and also registering the students to access BiTutor. An administrative staff has all the privileges enjoyed by an authorized staff. Authorized staff is a staff who has teaching experience in a particular course. These types of staff use BiTutor to design course (frame syllabus) which was assigned to him. Staff will send the request to the administrative staff to create that course. Once the administrative staff creates the course, the authorized staff will design the course using BiTutor, which provides the suitable interface to create the topics, and the use of graphical tools to create the Bayesian network and assign its parameters. Each topic has its own Bayesian network. Staff can also add the tutorials and assign them to particular concept and test items also can be added. When a staff creates a topic and finalizes its Bayesian
network and mark it as ready to teach, then he cannot modify the network. The structure of the Bayesian network cannot be changed once it's made ready to teach. He can only modify the tutorials and can modify the test items, or he can add new test items to item bank. Finally, a normal staff uses BiTutor to check the student's progress at any time during the tutoring session. The authorized staff can do all the work of a normal staff. Student's progress report is presented in an easy to read the format. BiTutor also provides the staff the facility to examine the tutoring strategy to be applied for a particular student.

BiTutor was developed using Java, and JSP to support self-paced learning in World Wide Web (WWW) environment. BiTutor (Nouh et al., 2007b) was developed to tutor college students in the computer science courses. More information about development of BiTutor, its architecture, its functionality, and its evaluation is given in detail in Chapter 5.

1.5 Guide to This Thesis

As a guide to the reminder of this thesis, Chapter 2 provides the literature survey focusing on student modeling, pedagogy, and assessment and the approach used in this research. The construction of student model using Bayesian network is explained in Chapter 3 along with the methods used to set the parameters. Techniques used for updating the Bayesian network are also explained. In Chapter 4, the extension of Bayesian network to decision network for selecting the suitable tutoring action, and the extension of Bayesian network to dynamic Bayesian network and then to dynamic decision Bayesian network for generation of suitable tutoring strategy are explained. In Chapter 5, the development of BiTutor and its evaluation is explained. Finally, in Chapter 6, the conclusions of this research work and its future-work are presented.