Jennifer Novosad

6.834J, Problem set 1

1 Part A: Three Topics of Interest

1.1 human-robot interaction

Part of the effort to make robots integrated into society is the effort to improve human-robot interactions. Human-Robot interactions can be likened to an intelligent user interface, which might need to modify its behavior based on human moods, attitudes, expertise, et. Some of the current work in general human-robot interaction includes language processing, fuzzy logic, decision trees, emotion imitating, personality, gesture/facial recognition, and modeling human behavior. I am interested both in what qualities make for a robot that is easy to interact with, and how a robot can model human behavior.

1.2 Bayesian Belief Networks

These networks bring together probabilities and directed graphs, and are used frequently in machine learning. The graph carries information about the conditional dependencies between nodes, and about the probability distribution functions of the nodes (often assumed Gaussian). Nodes can be random variables, parameters, observations, or hypotheses. The problem with using these networks is that acquiring knowledge and updating them are time costly. Some problems, such as inference, are NP-hard. I would like to more about how these networks can be applied, and what sorts of approximation algorithms exist to help learn a Bayesian network.

1.3 Multi-agent Task Solving in Robotics

Some of the problems of multi-agent robotics include cooperation, diversity, communication, distributed learning and distributed planning. Some of the test beds of these principles include robotic soccer, foraging, and defending a home base. Some of the current research looks at optimizing ones individual task for the benefit of the group, collective locomotion, insect behavior, and learning team solutions. I would like to learn how a group of entities learns about a problem, as opposed to a single entity.

2 Part D: Researching a Critical Reasoning Method

I chose three papers on work in virtual tutors, focusing on weakness identification and student models. Of the three papers, one paper provides a current implementation example, the second describes a higher level goal for what should be included in a virtual tutor, and the last gives a mathematical model for a student. Aaron D'Souza's paper gives an example of what sorts of work are currently being done in commercial virtual tutoring, and provides concrete implementation details. The second paper, "Tactical Action Officer Intelligent Tutoring System (TAO ITS)" by Richard Stottler, describes a direction in which to move ITS, and describes the effectiveness of ITS in a high dimensional learning environment. The last paper, section 4 of "An Intelligent Distributed Environment for Active Learning" by Yi Shang, Hongchi Shi, and Su-Shing Chen, gives a mathematical description of a student model, and describes the strengths and weaknesses of this model. Because the student model is only roughly covered by the previous two papers, it fleshes out some of the important details that need to be understood before implementing a virtual tutor. Each of these papers describe a different architecture, and focus on teaching a different kind of knowledge to the student; however, all of them give information on student models and interactive tutoring.

2.1 Paper 1: An Automated Lab Instructor for Simulated Science Experiments

Source:

Aaron D'Souza, Jeff Rickel, Bruno Herreros, and W. Lewis Johnson. "An automated lab instructor for simulated science experiments". In Proceedings of the International Conference on Artificial Intelligence in Education (AI-ED 2001), pp. 65-76, San Antonio, TX, May 2001. http://www-clmc.usc.edu/ adsouza/

I selected this paper while searching for ideas on how to identify student weaknesses. I found few papers that mentioned student weaknesses identification at all, and most of those that did referenced this paper as a concrete implementation. This paper turned out to be about the implementation of an automated lab instructor (ALI). The ALI program allows students to modify variables in a few simulated experiments, and asks them multiple choice questions on what they have learned. ALI also suggests experimental variables to modify. ALI's big successes are asking appropriate questions and suggesting good variables by noticing when an experiment demonstrated a concept well, and remembering if the student has already learned that concept.

This paper made several contributions towards implementing a virtual tutor. It claims to be one of the first fusions of previous work done in language, knowledge representation, tutoring, and physics simulation. Many commercial tutors used language or physical simulation, but not both. This paper provides an architecture of the program, which could be used to implement similar projects. It also provides evidence of the system's effectiveness with a small study of the program on 6 high school students.

This paper's biggest strength is in its descriptions of implementation details, which are fairly clear and explain enough to help someone else build a similar system. The study's methodology also made a strong case; after using the program, the students gave more concrete answers and focused on the relationship between variables on the post test.

This paper's greatest weakness was a lack of emphasis on big ideas, or higher level concepts. It main idea is the architecture for organizing the important components, such as a student model and low level reasoner. However, each of these components is very simplified. For instance, the student model only stored if the student had seen concepts, had them explained, and correctly answered questions on them. It did not store the way in which the student learned concepts, or what problems the student could apply them to. The paper did not make clear if the program could easily be extended to include this information, or what other components were similarly incomplete. Also, while the methodology of the study seemed very good, the size of the study (6 students) was too small to be significant.

After reading this paper, I went back and revised my cognitive robotic system's architecture. This paper's emphasis on symbolic representation also gave me the idea to embed the higher level concepts into the programming language, similar to the fault isolation lecture in class.

Many of the low level implementation details in this paper would not apply to cognitive robot, because of the differences in environment. While ALI could work in an environment in which the student behavior was restricted to modifying a small number of variables in a controlled situation, or choosing a multiple choice answer in response to a question, a robotic karate teacher would need to work in an environment where the student response was more varied, and the student knowledge for each concept could not be expressed well as 'done' or 'not done'.

2.2 Paper 2: Tactical Action Officer Intelligent Tutoring System

source:

Stottler, R., M. Vinkavich (November 2000), "Tactical Action Officer Intelligent Tutoring System", Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2000) http://www.shai.com/papers/IITSEC-00-TAO.pdf

I found a reference to the TAO ITS while searching for a method of creating a student model when the simulation has more unknowns than how the student will modify two or three experimental variables. The TAO ITS is a program owned by the U.S. Navy, used to help train future officers in tactics. The tutor observes the student playing a ship on a simulated battle field, and provides feedback after game play on how successful the student was. The program can also design simulations, warn a human teacher about student difficulties, give reading material on tactical concepts, and replay student errors. This program interested me because the student has a range of actions that he can choose to solve a problem in which enemy goals are ambiguous. I selected this report on the system because it appeared to have the most details about the program and what were the goals of the designers.

This paper's biggest contribution was the description of the qualities of a good student model. A student model should include tasks the student has performed, and his performance on those tasks. A student model also needs a method to estimate mastery of skills, knowledge, and the ability to apply them when appropriate. The tutor must also keep track of what circumstances the student can apply the knowledge. This student model not only accounts for factual knowledge, but the ability to integrate that knowledge into experience. The high level ideas in this paper were its strength. The ideas in the paper suggested a direction in which ITS research should move.

However, very little implementation suggestions were made. After describing what ITS is, and the high level goals of it, the paper mostly focused on look of the user interface. Additionally, while the paper provided a promising looking study of the opinions of 12 students on the program, the study did no evaluation to see if the students actually learned by using the program.

This paper describes several aspects of an ITS that I feel are necessary for the karate tutor cognitive robot. The student model and the weakness finding of the tutor model share several key features. For example, in both tactics and sports the ability to apply knowledge at the appropriate time is as important as the knowledge itself. Also, the student learning environment in this system has both environmental uncertainty, and many options for student behavior. The karate tutor will need to be able to function in an equally difficult environment. However, there are some differences between the tactics and karate teaching, such the value of waiting to the end of a simulation to provide feedback, which would need to be modified for the karate tutor.

2.3 Paper 3:An Intelligent Distributed Environment for Active Learning

Source:

Yi Shang, Hongchi Shi, and Su-Shing Chen, "An Intelligent Distributed Environment for Active Learning" Section 4, Department of Computer Engineering and Computer Science at University of Missouri-Columbia, Copyright May 1-5, 2001, Hong Kong. http://www10.org/cdrom/papers/207/WWW10-207.html

While looking though summary papers on student models, I noticed that several of them mentioned using a Bayesian model for the student. I chose this paper because it provided a mathematical explanation of how to feasibly model a student as a Bayesian model, and blatantly listed the strengths and weaknesses of the model.

The major contributions of this section are concisely explaining what methods of student modeling have already been attempted, what kinds of weaknesses are common in student models, and proposing a student model which is both computationally reasonable and models uncertainty in student behavior. The proposed model is explained in mathematical terms, and the model's weaknesses and strengths are listed.

Section 4 provides the information needed to understand many other papers on ITS. It also completely opens up the black box student model, and shows a great deal of the details that would be needed. This section's main weakness is that it doesn't provide an example of this being used, or show evidence of any testing.

This paper shows some of the internals of a student tutor that the karate tutor would need to model student behavior. This model is robust against students accidentally making errors or guessing, to it gives a better explanation of students than the boolean model described in the first paper. Also, it is a linear time algorithm in the number of data items, so it should be able to be run in real time. While this exact model does not meet the karate tutor's needs because it can only model one skill, it could be extended to meet the karate tutor's needs.

3 Part E: A Simple Project

While I would really like to give an advanced lecture on this sort of material, I am worried that its scope is too large to simplify into a simple term project.

A simple project in this area would be to create and test a student model which could identify student weaknesses in addition to their strengths. To complete this goal, at least 5 modules would be needed: an environment, a abstract concept representation, a student model, and two reasoners. The simple project would not have a tutor model to avoid the need for language processing.

The environment in which the student learns should be at least as complex as a lode runner type game, where there are a variety of actions which constitute the same higher level concept, and there are several scenarios where each concept could be applied in. Ideally, the game should involve some uncertainty, such as other agents or two human players, so that the space of learn-able concepts is richer.

In order for the student model to be tested, the system would also need two reasoners. One reasoner would identify situations in which a concept could be learned. The second reasoner would identify how the student behaved in that situation. Both of these reasoners would be dependent on a good concept representation model.

Rather than have a tutor, the student model could print out suggestions for a human tutor listing the concepts seen and the expected level of understanding. Creating a tutor in addition to a student model would probably be too large a project for the course of one term, because of the amount of labor that would need to go into language processing, user interface, and creation of new environments to practice the skills that are weak.

In order to test the success of the student model, several human students should interact with the program, with the observation of a human teacher. Between sessions, the human teacher should provide tutoring in specific concepts, while not providing tutoring in others. The response of the model over time should indicate a faster strengthening in the tutored concepts than those that are not tutored.

This project would embody a few of the salient features of the karate tutor. Like the karate tutor, the learning environment of this project would be reasonably complicated, and the student would have a variety of possible actions to take.

I don't think it is likely that I will do this project. I am worried about the size of the project, compared to the value. I believe that this project would be a rediscovery of work that has already been done with TAO ITS. In addition to the student model and reasoners which I am interested in working on, this project also requires creating the environment for the student to learn in, and learning enough about this environment to develop higher level concepts for them to study. I worry that the cost of this project is much greater than the value gained.

Another project that I have been thinking of is extending SLAM to include multiple agents that could communicate maps to each other. For this project, I would need to create a software environment, re-implement SLAM, and create some robotic agents for the environment. I would then need to extend SLAM to incorporate simplified maps from other agents, and identify moving objects. Agents would behave slightly differently, so that the map received from another agent would have an unknown level of trust. I feel that this project is a better term project because there are incremental levels of completion, and it is easy to add other interesting features if I finish early. Also, most of the programming would be focused on the reasoning skill, rather than the test bed.