2006-1745: DESIGNING AN ARTIFICIAL INTELLIGENCE COURSE FOR ELECTRICAL AND COMPUTER ENGINEERING TECHNOLOGY STUDENTS

Michael Filsinger, University of Cincinnati

MICHAEL D. FILSINGER is an Assistant Professor of Electrical and Computer Engineering Technology at the University of Cincinnati. He received a BA in Mathematics and MS degrees in Mathematics and Computer Science from the University of Cincinnati in 1990, 1992, and 1994, respectively. In addition to teaching, he has served as a computer system administrator. He is a member of IEEE, ASEE, and the Phi Beta Kappa honor society.

Designing an Artificial Intelligence Course for Electrical and Computer Engineering Technology Students

Abstract

Undergraduate students in our Electrical and Computer Engineering Technology programs have the option to take an artificial intelligence (AI) course as a junior/senior level technical elective. This course is currently structured as a modified version of a typical computer science AI course, offering a survey of various theoretical techniques presented in a more mathematically accessible way. While the first offering of this course was moderately successful, it should have been tailored more specifically to engineering technology students. This paper discusses lessons learned from this first attempt, and presents a more technology-based project-oriented approach to such a course, outlining a ten week AI course tailored to the needs of our ECET students as well as providing samples of possible projects.

Introduction

Once considered a topic for purely theoretical computer science, the field of artificial intelligence (AI) has found its way into a large number of real-world technology applications, particularly in the area of control systems. As such, a course in AI is becoming increasingly important for electrical and computer engineering technology students. The key features of a technology-based AI course are that it must deal with students who lack the extensive mathematical background of a typical computer science student, and that it should focus on real-world applications, with at most a short survey of the full range of theory.

In the Spring Quarter of 2005 our Electrical and Computer Engineering Technology department introduced an AI course offered as a technical elective for junior/senior level students. This paper describes the first attempt, which was structured as a modified version of a typical computer science AI course, offering a survey of various theoretical techniques presented in a more mathematically accessible way. While this course was moderately successful, it should have been tailored more specifically to engineering technology students.

The focus of such a course should be to show both the capabilities and the limitations of AI in industrial applications. Given the focus of our curriculum, the principle topic would be rule-based expert systems with obvious applicability to industrial control systems, including probabilistic and "fuzzy" extensions to rule-based systems. To illustrate techniques in which computer systems can "learn" and adapt to their environment, advanced topics such as artificial neural networks and genetic algorithms could be introduced if time permits, using real-world case studies as examples. The paper outlines a proposed structure of a ten week AI course as well as samples of possible projects.

Description of the First Attempt

In Spring Quarter, 2005, we offered the Artificial Intelligence and Expert Systems class for the first time. This course was a 10-week introduction to the AI field modeled as a modified computer-science AI course with less emphasis on intense mathematics. The intent was to provide a survey of techniques used in the AI field, with an emphasis on the capabilities and limitations of the current state of the art.

The first topic introduced was propositional and first-order logic. Since the audience was electrical and computer engineering technology students, they were already familiar with basic Boolean logic, but they had not been exposed to the complete first-order logic system, which forms the basis of rule-based expert systems. Basic logical proof and inference techniques were shown.

The second topic was actual rule-based expert systems. In these systems, a database of known facts and inference rules is initially given. Queries can then be made about relationships not explicitly shown in the database but derived from the information found there. Principles of forward and backward chaining were introduced. Toy examples, such as genealogical ancestry and simplistic medical diagnosis, were used to illustrate the ideas. Programs were written in both LISP and Fril.

Next, we introduced probabilistic reasoning. This is a simple extension of rule-based systems, where instead of having known facts and crisp rules, facts and rules are presented with a probability of truth. For instance, a fact might be represented as: has_pneumonia(x) [0.01], meaning that any given person (represented by the variable x) has pneumonia with a likelihood of 0.01. A rule might be represented in the form: has_flu(x) → has_aches(x) [0.6], meaning that if a person has the flu, then he has body aches with a likelihood of 0.6. This is a much richer language than simple rule-based systems, and in fact, it can be used along with Bayesian techniques to reason from consequent to antecedent. Since in diagnostic situations, we typically know the causal linkages from problem to symptom, but we typically want to reason from symptom to problem, such techniques can be far more useful than the simple chaining techniques found in simple rule-based systems. The toy example of medical diagnosis used in the previous labs was extended to illustrate this technique.

Next, fuzzy extensions to rule-based systems were introduced. Fuzzy set theory and fuzzy logic were designed to capture concepts which are not clearly defined. For example, the idea of "tallness" is not crisp. One person might say that "Harry is tall," while another person might say that he is "not tall." We might capture this notion with a rule of the form is_tall(Harry) {0.7}, meaning that Harry is tall with degree 0.7, but that he is simultaneously not tall with degree 0.3. This is not a probability of Harry being tall. He is what he is. However, his height falls into a fuzzy area, where he could be considered tall or not tall, depending on the perception of the observer. Various examples of fuzzy concepts are tall/short, hot/cold, old/young, heavy/light, etc. Since many of these concepts are common in industrial applications, a basic introduction to the idea of fuzziness and some of the techniques for handling it is important. In lab, the medical diagnosis example was further extended to include fuzzy concepts, such as
"X has a fever."

Next, we examined the idea of searching for solutions in a space of possible outcomes. This idea is implicit in rule based systems, but it is a general idea applied to most algorithmic AI processes. For instance, given a particular board position in a tic-tac-toe game, which move will provide the optimal outcome. Since there are a finite number of board positions, one could simply "look ahead" by looking at every possible move, looking at every possible response to that move, looking at every possible response to that response, etc. Given a finite space of possibilities, such a search is always guaranteed to result in an optimal choice. However, two difficulties arise. First, the search space might be infinite. For instance, in a chess game, the two players might simply keep moving the same two pieces back and forth forever. Second, even if the space is finite, the time required to actually search the entire space of possibilities might be prohibitive. In either case, some form of heuristic is required to prune unproductive search paths.

Finally, the idea of machine learning was introduced. The twin tasks of classification and clustering were presented to illustrate this idea. These are standard pattern recognition tasks, useful in identifying differences between objects based upon sensor inputs. In a classification task, the intent is to discriminate between objects of two or more known classes, such as "balls" and "blocks." Known training data belonging to these classes is provided, and based on that data, an algorithm for distinguishing between unknown objects belonging to these classes must be determined. Clustering is the task of determining differences between objects, without knowing anything about the classes to which these objects belong. Neural networks were introduced as a solution to the classification task. Furthermore, a brief introduction to genetic algorithms was given to illustrate an additional technique for machine learning.

Lessons Learned

While the first offering of our AI course was moderately successful, a number of aspects of the course were not ideal for engineering technology students. The first lesson derived from this experience was that ten weeks was not enough time for a complete survey of the field of artificial intelligence. As a result, no real depth could be achieved. Depending on the goals of the course, this could be acceptable, but in general, the expectation of technology courses is that students should be able to apply what they learn. Therefore, fewer topics should be covered in greater depth, allowing students to better absorb those topics.

Second, the initial run of this course proved to be too abstract. Not enough practical application was provided. To the greatest extent possible, theory should be kept to a minimum, with greater emphasis on practical application. Virtually all of the standard classroom problems in AI are "toy problems," which have little or no connection to what students would consider useful applications. Both classroom examples and laboratory experiments should focus on how to apply the techniques to real world problems.

Finally, I was surprised that while the mathematics was difficult for the students, it was not actually beyond their reach, at least from a mechanical perspective. Technology students experience a large number of subjects which have a large amount of difficult mathematics, and they are used to understanding a technical subject and handling the associated mathematics, without actually understanding the theoretical aspects of the underlying mathematics. As such, less time should be spent explaining mathematical theory, allowing more time to be devoted to actually using the techniques being studied.

Outline of a Technology-Based Approach

As noted before, the primary constraint facing an introductory artificial intelligence course is the amount of time available. Artificial intelligence is a very broad field with too many diverse topics and applications to be covered in a single quarter or semester. The first step toward devising an appropriate topic list is to identify what aspect of AI will best support the program's curriculum. This will define what topics will form the core of the course. Additional topics can then be prioritized, adding these topics as the time constraints given for the course permit.

Given the nature of our ECET program, the focus should be on AI as it supports industrial control applications. As such, rule-based expert systems, with their associated probabilistic and fuzzy extensions, form the core of the revised course. Given the constraints of a ten-week quarter, the only other topic which can be reasonably covered would be simple pattern recognition using neural networks.

An expert system attempts to provide a response equal to or better than what a human expert would provide at a very narrowly defined task, given a set of observed inputs and a database of "domain knowledge" programmed and stored beforehand. Such systems form the core of most computerized control applications. These systems can also be applied to subjects, such as medical or hardware diagnostics. While not "intelligent" in the sense that students expect, such applications are the most common practical uses of "artificial intelligence" as it exists today.

On the other hand, pattern recognition – the process of distinguishing different classes of object or data based on a chosen set of quantifiable features – is useful in applications involving the interpretation of visual or other sensor data. In fact, pattern recognition tasks could be incorporated into the inputs used for a rule-based expert system. Such hybrid systems can be found in many interesting applications, such as robotics. Interpreted sensor data forms part of the basis on which the robot will make decisions about what action to take.

The following is an ordered list of topics for a ten-week course in artificial intelligence and expert systems based around such applications:

1. possibilities and limitations of AI
2. predicate and first-order logic
3. resolution and inference
4. simple rule-based expert systems
5. probabilistic extensions and Bayesian inference
6. fuzzy extensions and approximate reasoning
7. pattern recognition – classification
8. artificial neural networks

Technology courses should always be focused on practical application. While "toy examples" typically used in AI and other computer science courses serve to illustrate the abstract concepts in a simple way, they lack strong connection to applications which might be encountered by students in the workplace. While real-world examples lack the simplicity of a toy example, they are much better at illustrating the connection between abstract technique and practical implementation. While a concept might be introduced with a simple non-practical example, they should always be reinforced with a real-world case study.

The following is a partial list of applications around which examples and projects could be based[3,4]:

- autonomous robots and vehicles
- temperature control
- quality control and defect detection
- diagnostics
- manufacturing process control
- scheduling
- authorization of financial transactions
- object recognition

To summarize, a technology-based AI course should restrict the topic list to those parts of AI relevant to the needs of the curriculum, while every topic should focus heavily on actual application rather than abstract theory.

Example Projects

Many possible projects exist which have obvious practical application. I will outline four possible laboratory projects for a technology-based AI course.

First, students could build a simulated control system. Such a system might represent part of a manufacturing process, or it might represent a system like an HVAC system. An external simulated set of inputs would be provided, as would a simulated device to be controlled. The actions of the controller would likely affect the simulated inputs as well. The task for the students would be to write a rule-based controller, which would take the input values, use the provided inference engine and the rules specified in the domain knowledge database to determine the best course of action. Later, such a controller could be modified to handle "fuzzy" concepts, such as "hot," "fast," etc.

A second possible student project would be a system diagnostics program. Such a program would consist of a database of rules specifying what symptoms would occur, given that a particular problem exists, and the likelihood of those symptoms occurring. For instance, a rule might be of the form:

disk_failure(X) → makes_noise(X) [0.7]

meaning that if a particular disk is failing, it makes a noise with likelihood 0.7. Once the database is established, Bayesian reasoning can be used to determine the likelihood that a disk is failing, given a set of symptoms which are known to exist. Again, this is an ideal case for the use of fuzzy concepts.

A third possible project would be to implement a neural network to detect defective "widgets" in a manufacturing line. The students would be given a set of training data consisting of a number of known good and known bad widgets, as well as a set of measured numerical features associated with those widgets. This data would be used to train a neural network, which could be used to detect future defective widgets coming off the line.

For a truly ambitious class, students could attempt to simulate an autonomous agent, such as a robot, moving around a field. The agent would interact with its environment through "sensor inputs," which must be interpreted and used to determine the best course of action. In some ways, this could be treated as a relatively simple game. In fact, one such game would be to have students write control programs for their robots to compete in some form of autonomous robotic combat contest. A robot could sense the presence and location of nearby walls, obstacles, or other robots. It would then take whatever action the designer feels appropriate for their strategy. To simplify the project, the robot "shell" could be provided by the instructor, leaving only the AI strategy engine for the students to design.

Student Feedback

The Artificial Intelligence and Expert Systems course has only been taught once – in Spring Quarter 2005 – and will not be taught again until Spring Quarter 2006. As such, student feedback is only available on the initial run of the course, not on the proposed changes. Student feedback from the first offering has been incorporated into much of the preceding material. However, a few comments deserve individual mention.

First, many students noted that the material was not what they expected. The term "artificial intelligence" is a misnomer. The term "expert system" is much more accurate for most applications. At the moment, true intelligence is beyond the reach of even state of the art AI technologies. However, people are surrounded with visions conjured by science fiction, and students often expect more than can be delivered. While nothing can be done about inflated expectations, the distinction between fact and fiction must be emphasized, and students must be shown that even with the limitations imposed by reality, the field is fascinating and has many practical applications.

The second comment was that too much time was spent on reviewing necessary background mathematics. The students already had a reasonable grasp of Boolean logic from their digital systems courses, and they already understood the relatively simple probability theory used in the course. So, while some background mathematics still needs to be introduced and explained, not as much time needs to be devoted to the task.

Finally, most of the students complained that the examples and projects were too abstract. They were unsatisfied with the toy examples used in the lecture and for projects. They seemed to enjoy the medical diagnosis project, as they could connect this to an actual application, but the other examples and projects simply lacked any real connection for them. All projects – and most examples – should be based on applications the students can see and understand. This will facilitate greater student understanding and participation.

Conclusion

Artificial intelligence is a fascinating subject with many applications relevant to engineering technology students. However, the subject is too broad and diverse to be covered adequately in its entirety in only a single course. Care must be taken to restrict the scope of the course to those aspects having the most relevance to the program curriculum. Examples and projects must be carefully chosen to be interesting, accessible, and relevant. Furthermore, while the subject is mathematically intense, given carefully chosen topics, the mathematical sophistication can be kept within the capabilities of the students.

Bibliography

1. Negnevitsky, M. (2005). *Artificial Intelligence: A Guide to Intelligent Systems, 2nd ed.* Addison Wesley, Upper Saddle River, NJ

2. Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach, 2nd ed.* Prentice Hall, Upper Saddle River, NJ

3. Bartos, F. (2003). *Artificial Intelligence ...Within*, from Control Engineering, September 1, 2003, http://www.manufacturing.net/ctl/article/CA321037

4. Waltz, D. (1996). *Artificial Intelligence: Realizing the Ultimate Promises of Computing*, Computing Research Association, http://www.cs.washington.edu/homes/lazowska/cra/ai.html