Enhancing Undergraduate AI Courses through Machine Learning Projects

Ingrid Russell[1], Zdravko Markov[2], Todd Neller[3], Susan Coleman[4]

Abstract - **It is generally recognized that an undergraduate introductory Artificial Intelligence course is challenging to teach. This is, in part, due to the diverse and seemingly disconnected core topics that are typically covered. The paper presents work funded by the National Science Foundation to address this problem and to enhance the student learning experience in the course. Our work involves the development of an adaptable framework for the presentation of core AI topics through a unifying theme of machine learning. A suite of hands-on semester-long projects are developed, each involving the design and implementation of a learning system that enhances a commonly-deployed application. The projects use machine learning as a unifying theme to tie together the core AI topics. In this paper, we will first provide an overview of our model and the projects being developed and will then present in some detail our experiences with one of the projects – Web User Profiling which we have used in our AI class.**

Index Terms - Artificial Intelligence Education, Data Mining, Machine Learning,.

INTRODUCTION

An undergraduate introductory Artificial Intelligence (AI) course provides students with basic knowledge of the theory and practice of AI as a discipline concerned with the methodology and technology for solving problems that are difficult to solve by other means. The importance of AI in the undergraduate computer science curriculum is illustrated by the Computing Curricula 2001 recommendation of ten core units in AI [1]. It is generally recognized that an undergraduate introductory AI course is challenging to teach [2]. This is, in part, due to the diverse and seemingly disconnected core topics that are typically covered. Recently, work has been done to address the diversity of topics covered in the course and to create a theme-based approach. Several faculty have been working to integrate Robotics into the AI course [3,4]. Russell and Norvig use an agent-centered approach [4], while Nilsson uses an evolutionary-based approach [5].

In this paper, we present work funded by the National Science Foundation to address this problem and to enhance the student learning experience in the course. Our work involves the development of an adaptable framework for the presentation of core AI topics through a unifying theme of machine learning. A suite of hands-on semester-long projects are developed, each involving the design and implementation of a learning system that enhances a commonly-deployed application. The projects use machine learning as a unifying theme to tie together the core AI topics.

Our proposed work will undoubtedly draw comparison to the introductory text by Nils Nilsson, *Artificial Intelligence: a New Synthesis*, which takes an evolutionary, agent-based approach to topic unification with a heavy emphasis on machine learning [5]. The text is geared toward a student with significant prior coursework in mathematics. Being mathematically focused, the text places almost no emphasis on application of ideas through implementation. Indeed, the preface states, "Although some pseudocode algorithms are presented, this book is not an AI programming and implementation book." In contrast, our approach would allow for varying levels of mathematical sophistication with implementation of concepts being central to the learning process.

This is a collaborative work among three institutions, a small liberal arts college, a comprehensive private university, and a large state university. Initial testing of these projects was done at the institutions of two of the Principal Investigators (PIs), the University of Hartford and Gettysburg College. Further testing is currently taking place at Connecticut State University (CCSU), the institution of the third PI.

Hereafter we will present in some detail one of the projects - Web User Profiling, which was used in the AI class at the University of Hartford during Fall 2004 and also during Spring 2005 at CCSU. We will also present initial evaluation of the project and our experiences using the material that is being developed. The Web User Profiling project focuses on the use of machine learning techniques to create models of Web users. Students collect Web pages from Web searches or by running a Web crawler and label them according to user preferences. The labeled pages are then encoded as feature vectors and fed into the machine learning system. The latter produces user models that may be used for improving the efficiency of Web searches or identifying users.

Working on the Web User Profiling project students build and experiment with the basic modules of an important application – an Intelligent (personalized) Web browser, which can use the learning scenario described above to learn

[1] Ingrid Russell, University of Hartford, irussell@hartford.edu
[2] Zdravko Markov, Central Connecticut State University, markovz@ccsu.edu
[3] Todd Neller, Gettysburg College, tneller@gettysburg.edu
[4] Susan Coleman, University of Hartford, scoleman@hartford.edu

its user preferences and provide ranking of the Web pages visited according to these preferences. The project introduces students to a key Data Mining application in marketing and offering personalized services, an area referred to as "data gold rush". The benefit from using this type of topical area is twofold. Firstly, students are better motivated to do the projects and learn more about important areas as Machine Learning and Data Mining and secondly, they learn the relation between these areas and Artificial Intelligence and thus better understand the practical aspects of an area sometimes considered as a pure academic research.

The paper is organized as follows. The next section provides a brief description of the AI course which provided the framework for the Web User Profiling Project. Then we present an overview of the NSF project, its goals and the student projects we have developed. Finally, we discuss in more detail the Web User Profiling project.

OVERVIEW OF THE AI COURSE

The general objective of the undergraduate AI course is to present Artificial Intelligence as a coherent body of ideas and methods and to acquaint the student with the classical approaches in the field and their underlying theory. Students explore this through problem-solving paradigms, search and control methods, logic and theorem proving, language and learning. The objectives of this particular AI course are:

- Introducing students to the basic concepts and techniques of Artificial Intelligence.
- Learning AI by doing it, i.e. developing skills of using AI algorithms for solving practical problems.
- Gaining experience in doing independent study and research.

The second objective actually provides the framework for including the Machine Learning projects developed for the NSF grant and in particular the Web User Profiling one.

The course is offered at the Junior and Senior levels with Data Structures as a prerequisite. Sample syllabi used can be found at: http://www.cs.ccsu.edu/~markov/ccsu_courses/ArtificialIntelligence.html and at http://uhaweb.hartford.edu/compsci/ccli/FinalVersion/AISyllabus.pdf.

The syllabi are based on the Russell and Norvig AI text [6]. This semester at CCSU, Prolog is being introduced in the AI course. A set of Prolog programs is developed to accompany the course material and provide the opportunity for demonstration and experiments with basic AI algorithms. These programs are also used as a basis for student assignments. In addition to these assignments, the Web User Profiling project is assigned as a term project. It should be noted that the Web User Profiling project that we present here can be used regardless of the language used in the AI course.

All materials for this course (including the Prolog programs) are available on the Web at the CCSU course Web site listed above. In Figure I below we provide a brief outline of the syllabus that shows the main topics taught and the schedule of the assignments.

Introduction: Intelligent agents

Problem Solving by Search, Intro to Prolog: State space representations, uninformed search algorithms, Prolog built-in search (searching and-or trees), evaluating search algorithms and experiments with simple Prolog programs.

Heuristic (Informed) Search: Basic algorithms, heuristic functions and their properties, experiments with searching sample graphs, road maps, solving 8-puzzle problem and sorting lists by searching. Intelligent agents

Constraint Satisfaction: Problem setting, backtracking, forward checking and arc consistency, experiments with map coloring and N-queens problem.

Searching Game Trees: Minimax and alpha-beta algorithms, experiments with a sample game tree and the game of tic-tac-toe.

First-Order Logic: Introduction to FOL, Clausal form and Prolog, model semantics of FOL, Simulation environment for knowledge based agents - Wumpus world, experiments with Prolog programs for transforming FOL in clausal form, describing the agents' background knowledge and action rules.

Inference in First-Order Logic and Prolog: Resolution method, strategies, SLD resolution, Prolog, experiments with Prolog implementations of clausal resolution and the wumpus world agent simulation environment.

Knowledge Representation: Ontologies, situation calculus, semantic networks and semantic Web, default reasoning.

Planning: The planning problem, propositional planning, experiments with situation calculus (simple deductive planning) and a STRIPS-like planner.

Uncertainty and Probabilistic Reasoning: Probabilities, Bayes' rules reasoning, Bayesian Networks, experiments with Bayesian Networks.

Machine Learning – Basic Concepts: Version Space, decision trees, experiments with Prolog programs for version space learning and decision trees.

Machine Learning – Numeric Approaches: Clustering, evaluation, experiments with Prolog implementations of Naïve Bayes, nearest neighbor and agglomerative clustering.

Learning with Background Knowledge: Explanation-Based Learning, Inductive Logic Programming, Experiments with Prolog-EBL, ILP and FOIL.

Natural Language Processing: DSG grammars, question answering, experiments with Prolog implementations.

Other Topics and Philosophical Foundations

FIGURE I: SAMPLE SYLLABUS

The course includes a term project based on the projects developed in the framework of the NSF grant (see next section). Students select one of six and form teams (usually of 3 students) that work on this project. During the semester, teams are expected to provide 3 reports:

- Initial project description including a brief introduction to the area of the project, specific goals, approaches and algorithms and resources to be used (data, programming tools or applications) and a plan of how to achieve the goals and evaluate the project results.
- A report on the progress they made by midterm (due during the midterm week)
- A final report describing the results they achieved upon project completion (due during the final week).

Each team is also expected to present results of their work on the project during the final week. In addition to the term project, smaller assignments throughout the semester were assigned to reinforce learning the concepts covered in class.

OVERVIEW OF THE NSF PROJECT AND ITS GOALS

The objective of the NSF project is to develop a framework for enhancing the teaching of the introductory AI course. Our basic idea is to focus on an area which is a part of the core of AI and at the same time is a topical and challenging area. This motivates students to learn more about it and about its relation to AI. We consider Machine Learning as such an area. Machine Learning (ML) grew up as a sub-area of AI providing the basic methodology and technology for developing systems (such as robots or computer programs) that automatically improve their performance through experience. Many of the approaches, techniques and tools developed within the ML framework have found applications not only in AI related areas but in many other more general IT application areas. For in depth coverage of the theory of machine learning see [8].

Along with Statistics and Databases, Machine Learning provides the basic techniques and tools for Data Mining and Knowledge Discovery. The most popular ML algorithms in Data Mining are decision trees, neural networks and various clustering methods (unsupervised learning). They are used for extraction of implicit, previously unknown, and potentially useful information from data. There are many particular examples of Data Mining applications of Machine Learning. The student projects for Web document classification and Web User Profiling are among them.

Machine learning algorithms can play an essential role in building difficult-to-program applications. These are applications that have proven too difficult for traditional programming, such as face recognition and speech understanding. The most successful systems for face recognition, for example, were developed using training examples of face images together with machine learning algorithms. The use of ML algorithms is also the preferred approach where complex sensor data must be interpreted. The Character Recognition project that we have developed for the NSF grant falls in this category of ML applications.

The basic idea of using ML to customize software application**s** is that in many computer applications, such as online news browsers and personal calendars, one would like a system that automatically adjusts to the needs of individual users after it has been fielded. For example, we would like Web browsers that customize to the individual user's interests when searching the Web. Because it is unrealistic to manually develop a separate system for each user, machine learning is an attractive option for building software that automatically customizes itself to individual users. This approach is taken in the Web User Profiling project, which we describe in more detail in the next section.

Having in mind the role that ML plays in AI and in Computer Science we set the following goals for our work.

- Enhance the student learning experience in the AI course by implementing a unifying theme of machine learning to tie together the diverse topics in the AI course.
- Increase student interest and motivation to learn AI by providing a framework for the presentation of the major AI topics that emphasizes the strong connection between AI and computer science.
- Highlight the bridge that machine learning provides between AI technology and modern software engineering.
- Introduce students to an increasingly important research area, thus motivating them to pursue more advanced courses in machine learning and to pursue undergraduate research projects in this area.

Our approach to achieve these goals is based on developing a suite of customizable student projects to be integrated into a one semester AI course.

OVERVIEW OF MACHINE LEARNING PROJECTS

Our goal is to provide a number of projects that the faculty teaching AI can choose from and then easily customize according to their specific goals. While we envision using each as a semester-long project, the different entry points into the project will allow faculty to tailor these projects as they deem necessary for their courses. At one extreme, students may implement an entire machine learning system that illustrates a core AI topic. At the other extreme, students may apply our solution code to understand the computational characteristics of the algorithms. In between is a range of choices, where instructors individually decide how much implementation is best for their students.

The projects have various emphases, backgrounds and prerequisites so that they can serve different goals within the general framework of teaching AI. Some projects require intensive programming, some can be done best using an AI language (like Lisp or Prolog), others lend themselves to non-AI languages such as Java, or require little programming and rely on the use of tools. We have developed draft copies of six hands-on laboratory projects that can be closely integrated into a one-semester AI course. Figure II below provides a brief overview of the projects followed by a detailed description of the Web User Profiling Project. Details and complete project descriptions will be made available at the project Web page at: http://uhaweb.hartford.edu/compsci/ccli.

Web Document Classification: The project investigates the process of tagging Web pages using a topic directory structure and applies machine learning techniques for automatic tagging.

Data Mining for Web User Profiling using Decision Tree Learning: The focus here is on the use of decision tree learning to create models of Web users.

The Game of Clue: We use SAT solvers to deduce card locations in the popular game of Clue, illustrating principles of knowledge representation and reasoning, including resolution theorem proving.

Character Recognition Using Neural Networks: The project involves the development of a character recognition system based on a neural net model.

Solving the N-Puzzle Problem: Using Analytical Learning to Improve Search: The project involves the application of explanation-based learning to improve the performance of uninformed search algorithms when solving the N-puzzle.

Reinforcement Learning for the jeopardy Dice Game "Pig":
In this project, students model the game and several illustrative variants, and implement various learning algorithms to compute optimal play, and experiment with such algorithms to experience their tradeoffs.

FIGURE II: LIST OF ML PROJECTS

OVERVIEW OF WEB USER PROFILING PROJECT

The Web is the largest collection of electronically accessible documents, which constitute the richest source of information in the world. The problem with the Web is that this information is not well structured and organized so that it can be easily retrieved. Search engines help in accessing Web documents by keywords, but this is still far from what we need in order to effectively use the knowledge available on the Web. Machine Learning and Data Mining approaches go further and try to extract knowledge from the raw data available on the Web by organizing Web pages in well-defined structures or by looking into patterns of activities of Web users. This project focuses on this challenge and explores the Machine Learning techniques suitable for this purpose.

The aim of this project is to develop a system that helps us develop an intelligent Web browser. The project will focus on the use of Decision Tree learning to create models of Web users. Students will be provided with Decision Tree learning tools and will collect data from Web searches. They will then experiment with creating Web user models and using these models for improving the efficiency of Web searches performed by the same or new users.

This project is split into three major phases - *data collection, feature extraction* and *machine learning*. Phase I consists of identifying 5 users and collecting a set of 100 Web pages (documents) from a given topic. The pages are then labeled by user preferences. These documents will serve as our training set. Phase II involves feature extraction and data preparation. During this phase the Web pages will be represented as feature vectors, which in turn are used to form a training data set for the Machine Learning phase. At the machine learning phase, decision tree learning algorithms are applied to the feature vectors in order to create models of the users these vectors (documents) are mapped onto. Then the models can be used to filter out Web documents returned by searches so that the users can get more focused information from the search engines.

Phase I – Document Collection Grouped by User Preferences

The purpose of this stage is to collect a set of Web documents labeled with user preferences. This can be done in the following way: The user performs Web searches with simple keyword search or just browses Web documents. To each Web document, the user assigns a label representing whether or not the document is interesting to the user. For this project, students are asked to use a Web crawler to collect the Web pages. An example of a Web crawler is WebSPHINX (free open source), a customizable Web crawler available at: http://www-2.cs.cmu.edu/~rcm/websphinx/websphinx.jar. Students experiment with varying the following parameters: Crawl the subtree/the server/the Web, Depth/Breadth first, use different limits (number of threads, page size, timeout). This allows students to see how the dynamics of crawling changes by inspecting the Web page graph.

Students then select a set of initial URLs and use the crawler to collect 100 pages from the same server, the subtree or from the Web. For each user, they identify which of the 100 Web documents the user likes and which the user does not like. The text content of the page should be taken into account when users are identifying their views of a page as positive or negative.

Phase I Deliverables

- The description of the process used to select the Web documents and identify user likes and dislikes.
- The list of all 100 Web documents labeled by user likes and dislikes.
- Explanation of the Web crawler algorithm in terms of search. In particular, students answer the following::
  - o Is the Web a tree, directed acyclic graph, directed graph, or a graph? How are its nodes and edges represented?
  - o Which search algorithms are used by Web crawlers and why? Can a crawler go into a loop?
  - o How does the choice of the part of the Web to be crawled (subtree/the server/the Web) affect the search algorithm?
  - o How is multi-threading used to improve the efficiency of crawling algorithms?
  - o What happens when page size or timeout limits are reached?

Phase II - Feature Extraction and Data Preparation

Having the set of labeled Web documents, each one is used to create a feature vector (data tuple) labeled with the user preferences. These vectors form the training data set of the machine learning system. The basic steps to create this data set are the following:

1. Select a number of terms (words) that will be used as features to represent the documents. This can be done by putting all documents together and sorting in ascending order all words appearing in all documents by their frequency. The first N words in the sorted sequence can be used to represent the documents with vectors of size N.
2. Using the selected set of terms as features (attributes), create a feature vector (tuple) for each document with Boolean values corresponding to each attribute (1 if the term is in the document, 0 if it's not). A more sophisticated approach for determining the attributes values can be used too. It is called *Term Frequency–Inverted Document Frequency* and is based on using the term frequencies scaled in some way to normalize the document length and take into account the relative term frequency among different documents.
3. Create a data set in the ARFF format to be used later by the machine learning system. An ARFF file is a text file, which defines the attribute types (for the Boolean values they will be nominal, and for the frequency-based ones - numeric) and lists all document feature vectors along with their class value. According to the ARFF format the class is the last attribute in the sequence whose value appears as a last element in each data row. This value represents the document label assigned by the user (for example, interesting/non-interesting or yes/no).

Steps (1) and (2) are part of the so-called vector space model, which is well known in the area of Information Retrieval. Students with good experience in programming can write a program to create the vector space model. Another option is to use a text corpus analysis package that filters and extracts keywords with their frequency counts. An example of such a system is TextSTAT, freeware software available from http://www.niederlandistik.fu-berlin.de/textstat/software-en.html. Other such systems are also available as freeware from http://www.textanalysis.info/. Step (3) is discussed in Chapter 8 of Witten and Frank's book [7], which is available online at: http://www.cs.waikato.ac.nz/~ml/weka/Tutorial.pdf. More details about the ARFF file format can be found at http://www.cs.waikato.ac.nz/~ml/weka/arff.html.

Phase II Deliverables

Students submit ARFF training data files containing the feature vectors labeled with user preferences. It is recommended that for each user students prepare several files by using different approaches to feature extraction. For example, one with Boolean attributes, one with numeric based on text only, and one with numeric using the html information. Versions of the data sets with different numbers of attributes can also be prepared. A rule of thumb here is that the number of attributes should be less than the number of examples. In particular the following items should be provided:

1. The 5 ARFF training data files containing the feature vectors labeled with user preferences for all Web documents collected during Phase I.
2. A description of the ARFF data file including an explanation of the correspondence between the 100 keywords and the attribute declaration part of the ARFF file (the lines beginning with @attribute). Included also should be an explanation of the data rows (the portion after @data). For example, pick a tuple and explain what the 0's and 1's mean for the document that this tuple represents and what the last value in the row means for the corresponding user.

Phase III - Machine Learning

The learning phase uses the Weka Data Mining System – a free Machine Learning software package in Java available from www.cs.waikato.ac.nz/~ml/weka/index.html. This is one of the most popular ML systems used for educational purposes and is the companion software package of Witten's book [7].
The learning phase consists of the following steps:

1. Installation of the Weka package and familiarizations with its functionality. After installing Weka, read the online tutorial at:
   http://www.cs.waikato.ac.nz/~ml/weka/Tutorial.pdf
   which describes the Weka functionality without GUI well. Then read the GUI version user guide available at:
   www.cs.waikato.ac.nz/~ml/weka/ExplorerGuide.pdf and run some experiments using the data sets provided with the package (e.g. the weather data).
2. Preprocessing of the Web document data sets. Load the ARFF files created during Phase II, verify their consistency and get some statistics by using the preprocess panel.
3. Using Weka's decision tree algorithm (J48) examine the decision trees generated with different data sets. Which are the most important terms for each data set/user (the terms appearing on the top of the tree)? How do they change with changing the type of the data set (nominal, numeric)? Check also the classification accuracy and the confusion matrix obtained with 10-fold cross validation.
4. Web document filtering (focusing the search). Collect a number of Web documents returned by a search. Apply feature extraction and create an ARFF test file with one data row for each document. Using the training set for a particular user and the test set option classify the new documents. Each one will get a corresponding label (interesting/non-interesting). Simply discard the non-interesting documents and present the interesting ones to the user. Further, this step can be incorporated into a Web browser for automatic labeling of all Web pages as interesting/non-interesting according to user preferences.

Phase III Deliverables

1. Students are asked to explain the decision tree learning algorithm (Weka's J48) in terms of state space search by answering the following questions:
   - What is the initial state (decision tree)?
   - How are the state transitions implemented?
   - What is the final state?
   - Which search algorithm (uninformed or informed, depth/breadth/best-first etc.) is used?
   - What is the evaluation function?
   - What does tree pruning mean with respect to the search?
2. This stage of the project requires writing a report on the experiments performed. The report should include detailed description of the experiments (input data, Weka outputs), and answers to the questions above. Weka does not classify Web documents. Instead, Weka prints classification accuracy for the test set (a new Web document), which is simply a number (percentage). This number must be used to explain how this new document is classified for the particular user - like or dislike. The report should also include such interpretation and analysis of the results with respect to the original problem stated in the project.
3. Looking back, students are asked to propose changes in the process that could improve on the classification.

Final Deliverable

In the final deliverable, students explain the relation of the approaches used in this project to the AI areas of search and knowledge representation and reasoning (KR&R). In particular, they are asked to answer the following questions:

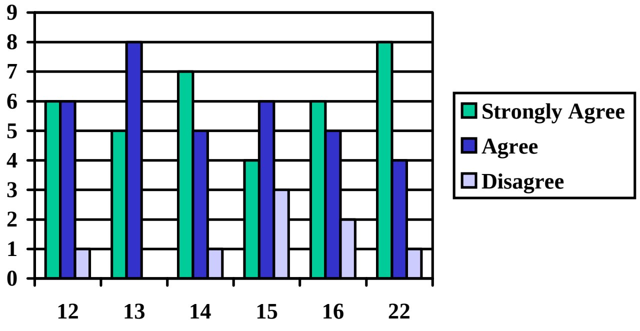
- Which search techniques have been used in the project?
- If no search has been used explicitly, what is the relation of the approaches used in the project to the area of search? (any search components used or project approaches applicable in the area of search)
- Which KR&R techniques have been used in the project?
- If no KR&R has been used explicitly, what is the relation of the approaches used in the project to the area of KR&R? (any KR&R components used or project approaches applicable in the area of KR&R).

Extra Credit Options

Students seeking challenges will find related advanced projects such as writing their own Web crawler to fetch Web documents to be classified by the system. They could introduce parameters to control the search. For example, one could use depth-first, breadth-first with some parameters to bound the search as depth or breath limits, number of pages to retrieve, time-out time for each page or for the whole run, size limits for the pages etc. Another option is to ask them to customize or add new and significant features to WebSPHINX.

EXPERIENCES

The projects were used as term projects in the AI courses taught by two of the PIs in fall 2004. At the end of the semester students from these courses filled out a questionnaire with 23 questions related to their experience with the course and with the projects. We present here some statistics from the AI course taught at University of Hartford as it involved all students in the class and thus provided more representative results. Figure 3 below includes 6 questions and a graph that summarizes the answers. The horizontal axis represents the set of questions and the vertical axis represents the number of responses for each category.



12. After taking this course I feel that I have a good understanding of the fundamental concepts in AI.
13. After taking this course I feel that I have a good understanding of the fundamental concepts in ML.
14. The student project was an effective way to introduce Machine Learning concepts.
15. Based on my experience with this course, I would like to learn more about ML and how it works.
16. Based on my experience with this course, I would like to learn more about the field of AI.
22. I had a positive learning experience in this course.

Figure 3: Evaluation Results

The graph clearly shows that the general feeling about using the ML projects in the AI class is positive. Students are better motivated to learn the fundamental concepts both of AI and ML. The projects also stimulate students' interest in additional AI and ML related areas. A number of students are currently pursuing an independent study course in ML and are working on a research project extending their work in the AI course.

Based on student replies in the questionnaire and the general experience of the PI's teaching these courses we revised the projects and now they are being tested with the AI course that the third PI is teaching during the spring of 2005. The initial experience from this course is positive too – students have already submitted the first progress reports, which include the specific project goals they have set based on the general objectives of the projects. These goals are quite ambitious and show their enthusiasm to work on the projects.

This initial experience shows that students are excited about working on practical problems and developing Machine Learning and Data Mining approaches. They also like the opportunity to choose their own particular application domain and further explore or extend the projects in areas they found interesting and challenging. Student evaluations and experiences from using these projects in this and in previous courses will be discussed at a project advisory board meeting scheduled for the end of spring 2005 semester. All this will allow us to make another revision of the projects during the summer of 2005. The revised projects will then be used again in the fall AI courses. Over 20 faculty members nationwide are affiliated with the project and will be testing the material next year.

CONCLUSION

We presented our experiences with an NSF funded project whose goal is to enhance student learning experiences in the introductory Artificial Intelligence course by implementing a unifying theme of machine learning to tie together the diverse topics typically covered in such a course. Toward that goal, we developed a set of hands-on semester long student projects involving the design and implementation of machine learning applications. In this paper we outline the projects that we developed during the first year of our work under the NSF grant and describe in detail one of them - the Web User Profiling Project. Our preliminary experiences using the projects are also presented.

AKNOWLEGEMENT

This work is supported in part by National Science Foundation grant DUE CCLI-A&I Award Number 0409497

REFERENCES

[1] Engel, G., and Roberts, E., Eds., *Computing Curricula 2001 Computer Science*, IEEE Press, 2002.

[2] Hearst, M., Improving Instruction of Introductory Artificial Intelligence, AAAI Press, 1995.

[3] Kumar, A., "Using Robotics in an Undergraduate Artificial Intelligence Course: An Experience Report", *Proceedings of Frontiers in Education Conference,* 2001.

[4] Kumar, D., and Meeden, L.., "A Robot Laboratory for Teaching Artificial Intelligence", *Proceedings of SIGCSE Conference*, February 1998.

[5] Nilsson, N., *Artificial Intelligence: A New Synthesis,* Morgan Kaufmann Publishers, 1998.

[6] Stuart Russell, Peter Norvig. *Artificial Intelligence: A Modern Approach,* Second Edition, Prentice Hall. 2003.

[7] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2000.

[8] Tom Mitchell, Machine Learning, McGraw Hill, New York, 1997.