## Project Suggestions

In addition to the sample projects given on the wiki, the following list is a list of suggestions for possible course projects. Of course you are free to choose your own project; these are merely some suggestions:

### Web User Profiling

Web searches provide large amounts of information about web users. Data mining techniques can be used to analyze this information and create web user profiles. A key application of this approach is in marketing and offering personalized services, an area referred to as "data gold rush". The aim of this project is to develop a system that can be used to develop an intelligent web browser. This project focuses on the use of Decision Tree learning to create models of web users.

### Character Recognition and Learning with Neural Networks

The power and usefulness of artificial neural networks have been demonstrated in several applications including speech synthesis, diagnostic problems, medicine, business and finance, robotic control, signal processing, computer vision and many other problems that fall under the category of pattern recognition. The goal of this project is to develop a character recognition system based on a neural network model.

### Web Document Classification

Along with search engines, topic directories are the most popular sites on the Web. Topic directories organize web pages in a hierarchical structure according to their content. The aim of the project is to investigate the process of tagging web pages using the topic directory structures and apply Machine Learning techniques for automatic tagging. This would help in filtering out the responses of a search engine or ranking them according to their relevance to a topic.

### The Game of Clue

The popular board game Clue serves as a fun focus problem for this introduction to propositional knowledge representation and reasoning. After covering fundamentals of propositional logic, students first solve basic logic problems with and without the aid of a satisfiability solver. Students then represent the basic knowledge of Clue in order to solve a Clue mystery. Could the current best stochastic SAT solver, Adaptive Novelty+, benefit from machine learning elements.

### Genetic Algorithms

Anyone familiar with the theory of natural selection can imagine how a difficult search problem might be attacked by combining the elements of reasonably good solutions and randomly mutating the resulting "offspring" to preserve variety. At a deeper level, there is often a disconnect between real-world problems and the kind of problems that students work on in an undergraduate AI or machine-learning course. This project focuses on learning and applying two recently developed genetic algorithms for solving real-world problems in multi-objective optimization and evolving behaviors for competitive robotic agents.

### Learning Relational Knowledge

Most approaches to machine learning assume the knowledge to be learned can be expressed as a set of attribute-value pairs, i.e., an entity and its properties. However, much richer knowledge can be found in the relationships between multiple entities. This project explores the challenge of learning relational knowledge by experimenting with two relational learning methods, one logic based and one graph based.

### Biomedical Term Classification

Due to the explosive growth of knowledge in biotechnologies (about 1500 research abstracts are added every single day to MEDLINE, an electronic repository of biomedical papers) an acute need for knowledge management tools has arisen. Natural Language Processing (NLP) can help

fulfilling this acute need. A major problem in BioNLP, the area of research at the intersection of Biotechnologies and NLP, is that same biomedical term can be frequently used with different meanings in biological texts. For instance, SBP2 can refer both to a protein or a gene. In this project, we combine NLP with Machine Learning techniques, namely decision trees and Naïve Bayes, to build software tools that classify biomedical terms based on the surrounding contexts in which they appear. In particular, we work with terms that refer to the following categories: DNA, RNA, protein and cell_line, cell_type.

### General-Purpose Problem Solver

Genetic programming (GP) is perhaps the most general of local search algorithms. It is particularly useful in solving design and optimization problems. Strengths of GP include the ability to work with heterogeneous data, and that a relatively low amount of information needs to be specified to achieve success. A number of patents now exist that have been achieved using genetic programming. The goal of this project is to learn about machine learning (problem formulation, search, and knowledge representation) by building a basic genetic programming framework and to use it to solve problems. The framework will be built piece-by-piece, as concepts are introduced.

### Probabilistic Reasoning with Naïve Bayes and Bayesian Networks

Bayesian (also called Belief) Networks (BN) are a powerful knowledge representation and reasoning mechanism. BN represent events and causal relationships between them as conditional probabilities involving random variables. Given the values of a subset of these variables (evidence variables) BN can compute the probabilities of another subset of variables (query variables). BN can be created automatically (learnt) by using statistical data (examples). The well-known Machine Learning algorithm, Naïve Bayes is actually a special case of a Bayesian Network.

The project allows students to experiment with and use the Naïve Bayes algorithm and Bayesian Networks to solve practical problems. This includes collecting data from real domains (e.g. web pages), converting these data into proper format so that conditional probabilities can be computed, and using Bayesian Networks and the Naïve Bayes algorithm for computing probabilities and solving classification tasks.

### Relational Learning for Web Document Classification

Most of the content-based approaches to text and web document classification are based on the bag of words model, well known from the area of Information Retrieval. This model is simple and efficient, but fails to capture many additional document features such as the internal HTML structure, language structure and inter-document link structure. All this however may be a valuable source of information for the classification task. The basic problem with incorporating this information into the classification algorithm is the need for uniform representation. For example, the content-based classification works well with the vector space representation, while hyperlink-based classification can be implemented by using graph models. This project introduces students to Relational (First-Order) Learning that allows various kinds of information to be represented in a uniform way and used for document classification. One of the most successful relational learning systems, FOIL is used to create relational representation of web documents and to solve classification problems.

### Machine Learning for Automated Reasoning

Automated theorem proving (ATP) is concerned with the development and use of systems that automate sound reasoning: the derivation of conclusions that follow inevitably from facts. A key concern of ATP research is the development of more powerful systems, capable of proving more difficult theorems. Automated reasoning in large theories is becoming more prevalent as large knowledge bases are translated into forms suitable for ATP. Of particular interest is to improve performance of ATP when solving many problems from the same axiom set, but each problem requires use of only a subset of the axioms. The focus of this project is the extraction of a sufficient subset of axioms for proving that a given conjecture is a theorem, using machine

learning (ML) to learn from existing proofs which axioms are more likely to be used in a proof of the theorem.

### Generating Dense Boggle Boards with Genetic Algorithms

The problem of generating dense boards (high-scoring boards) in the game of Boggle is a challenging and fun task. This project explores the fundamentals of genetic algorithms, which are well suited for this task for two reasons: first, it is impractical to exhaustively examine or maintain all candidate boards; and second, information gained along one local search path can be integrated in multiple meaningful ways with information gained along other paths.

### Application of Associative Matrices to Recognize DNA Sequences in Bioinformatics

Associative matrices are considered a type of neural network topology used to recall and recognize previously known or unknown patterns. For example, the Hebbian linear associative matrices can be trained to recognize a particular DNA sequence into another specimen sequence that may help biologist to identify similarities and other characteristics important in the knowledge and recognition of a particular sequence in another specimen. This approach may result especially beneficial in mutated sequences where mutations or other changes in the sequence as deletions and insertions are present. Associative matrices have been used to recognize characters, shapes, or specific objects from an image. Such changes are considered noisy patterns that are one of the important features of using associative matrices in this field.

The project introduces students to the use of associative matrices concepts in learning and pattern recognition. Students will experiment the use of the matrices to recognize DNA sequences and its possible mutations.

### Competitive Learning in Checkers

One of the earliest investigations of machine learning was Arthur Samuel's work with Checkers, developing a program that learned a board evaluation function through the experience of playing many games. Starting from Samuel's approach, students will develop Checkers programs that learn to play by competing with other students' programs. Competitions will take place over the internet using a simple server that communicates moves between clients, checks for the validity of moves, keeps track of (and limits) the time spent by each player, and reports the winners and losers.

In addition to the reinforcement learning aspects, the project deals with the problems of search, efficient representation of knowledge, and reasoning about time as a resource. Using the same server guarantees that each checkers program learns and can compete within the same environment. At the end of the semester, the project teams compete in a tournament, which adds yet another level of excitement and provides a straightforward, if relative, comparison of the resulting programs.

### Route Planning, State-Space Search, and Case-Based Learning

Route planning is the process of determining a route, or path, to follow to move from a starting location to a goal location. It is a good application to be tackled by simple state-space search methods, at least on the small scale. Re-generating routes from scratch can be expensive, and seems wasteful. Therefore case-based reasoning will be explored as a method for storing and re-using previously generated routes.

Case-based reasoning systems store information, route plans, in this case, according to a set of features that describe the context in which they apply. For route plans, the context might be just the starting and goal locations, but could also include more elaborate features like passing particular obstacles, going past particular landmarks, or being scenic or fastest. This project introduces students to the costs and benefits of state-space search algorithms, and examines CBR as an alternative or adjunct to a brute-force method. It experiments with how to break route plans into cases, how to choose useful indices, how to judge similarity, how to adapt a partial-match to fit a new context, and how to learn by storing new cases in the CBR memory.

### Solving the Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is a classic optimization problem: find the least-cost round-trip route amongst N cities, visiting all cities exactly once. In this set of assignments students will develop alternative machine learning approaches to solving the TSP. Some of the possible techniques include: genetic algorithms, simulated annealing, Boltzmann machines, Kohonen SOM, ant colony systems etc. The learning goal of these assignments is to use the TSP as a test-bed problem to study various machine learning techniques individually and comparatively.

### Discovering Optimal Policies with Value Iteration, Policy Iteration, and Q-learning

Value and Policy Iteration provide an excellent means for agents in a nondeterministic environment to determine an optimal series of actions through the solving of a Markov decision process (MDP). However, solving an MDP requires that an agent have a great deal of knowledge about its environment: specifically, the rewards for each state and the transition probabilities between states. When this knowledge is not available to the agent, it can be learned through experience. Reinforcement learning, specifically Q-learning, is a method for doing this. Q-learning is a form of *model-free* learning; a Q-learning agent can learn an optimal policy without any knowledge about its environment, given enough experience. In this problem, students implement value iteration, policy iteration, and Q-learning to discover optimal policies for both a toy map and for a realistic campus map. Applying these two approaches to the same set of problems provides students with an understanding of how learning can be used to make up for a lack of available domain knowledge.

### Understanding Genetic Algorithms through Interactive Play

Genetic algorithms explore the hypothesis space in search of a best hypothesis to solve a problem through the manipulation of a string representation of those hypotheses. Inspired by biological evolution, parts of good hypotheses are combined and used with limited exploration to evolve new hypotheses similar to how different genes could potentially be combined into better chromosomes making a better creature. This project presents an interactive game environment where the player (student) has to evolve members of a tribe of creatures, Modabus, in order to have the correct genetic makeup to overcome an obstacle impacting their society. Through manipulation of crossover, mutation, and selection the player (student) will guide the genetic algorithm to find a solution (a Modabu citizen with the proper level of intelligence, strength, and speed to overcome each obstacle) over the course of three increasingly difficult levels.

### Machine Learning for Games

Computer games are prevalent as entertainment. Machine learning techniques can be used to help the computer player become smarter by learning from its experiences. For example, the computer player can learn which moves to make in different situations.