CS 520 (MSCS): Introduction to Artificial Intelligence

16:198:520

Place:	CoRE, Room **101**
Time:	Monday, Wednesday 5:00pm - 6:20pm
Instructor:	Wes Cowan
Office:	Levin 257
E-Mail:	cwcowan at scarletmail.rutgers.edu (Instructor)
Office Hours:	Fridays 3-5, and by appointment.

Outline of the Course: This course is intended to provide a broad practical introduction to the concepts and methods in the field of artificial intelligence. "AI" represents a very diverse field of problems and techniques - we will focus on topics and ideas with broad application. The main focus and takeaway from this course should be the systematic representation of knowledge and the manipulation of these representations for general problem solving. If a problem can be represented systematically, any number of algorithmic techniques can be leveraged against it. Many of these algorithms will seem natural if not utterly naive - but applying them to appropriately represented and realized problems lends them power and generalizability.

The course will be divided roughly into three topics:

- Part I: Search - A Systematic Approach to Exploration and Discovery
  - Uninformed Search; Informed (Heuristic) Search; Adversarial Search; Local Search; Constraint Satisfaction; Logic and Satisfiability; Classical Planning
- Part II: Uncertainty - Representing and Coping with an Uncertain World
  - Probability; Inference; Bayesian Networks and Inference; Inference over Time; Utility and Decision Theory; Markov Decision Processes
- Part III: Learning - Adapting to and Understanding the World
  - A Model and Theory for Learning; Decision Trees; Perceptrons; Neural Networks; Non-Parametric Methods; Support Vector Machines; Statistical Learning; Clustering; Ensemble Learning:

Text: The main text will be **Artificial Intelligence: A Modern Approach** by Norvig and Russell. The book is not strictly necessary, but is incredibly useful, especially if you plan to do future work in AI.

Prerequisites: Students will be greatly assisted by familiarity and experience with basic algorithms and data structures, as well as a grounding in probability.

Grading: Grades will have three components: i) a midterm and a final (40%), ii) projects (55%), and participation (5%, assigned at the discretion of the instructor). The grade is primarily project based, as this course is intended to give you a good practical grounding in the material and implementation is key.

Tentative Schedule and Topics for Lectures:

- Weds 9/6: Introduction to AI, Problems and Applications of Interest
- Mon 9/11: Introduction to Search - The Best Way to Find the Answer is to Look for It

  - Goat / Cabbage / Wolf Problem
  - Uninformed (Blind) Search: Tree vs Graph, BFS vs DFS, Variants
  - Completeness, Optimality, Time Complexity, Space Complexity
- Weds 9/13: Heuristic Search, Adversarial Search
- Mon 9/18: Local Search
- Weds 9/20: Constraint Satisfaction
- Mon 9/25: Logic
- Weds 9/27: Logic and Satisfiability
- Mon 10/2: Classical Planning
- Weds 10/4: Probability, Inference
- Mon 10/9: Bayesian Networks
- Weds 10/11: Exact Inference on Bayesian Networks
- Mon 10/16: **Midterm** Covering all material *excluding* Bayesian Networks
- Weds 10/18: Approximate Inference on Bayesian Networks
- Mon 10/23: Temporal Models
- Weds 10/25: Prediction/Filtering/Smoothing/MLE
- Mon 10/30: Kalman/Particle Filters
- Weds 11/1: Utility / Decision Theory
- Mon 11/6: Sequential Decision Problems
- Weds 11/8: Markov Decision Processes
- Mon 11/13: Models of Learning, Learning Models
- Weds 11/15: Clustering, Decision Trees
- Mon 11/20: Perceptrons, Neural Networks
- Weds 11/22: Friday Schedule, No Class
- Mon 11/27: Neural Networks, Computer Vision
- Weds 11/29: Non-Parametric Models, SVMs
- Mon 12/4: SVMs
- Weds 12/6: Statistical Learning, EM Algorithm
- Mon 12/11: Clustering
- Weds 12/13: Ensemble Learning

- Dec 15-22: Final Exams

Timeline for Projects:

- Tuesday 9/13: Project One Released
- Monday 10/2: Project One Due, Project Two Release (by Midnight)
- Wednesday 10/18: Project Two Due, Project Three Release (by Midnight)
- Monday 11/13: Project Three Due, Project Four Release (by Midnight)
- Weds 12/13: Project Four Due (by Midnight)

Summary of Project: There will be four main projects for this course, and they are meant to be the focus of your efforts and concerns. These are to be **group** projects, with groups of **no fewer than 3 people** and **no more than 4 people**. For each project, you are free to code in the language of your choice, so long as it is not deliberately perverse. Each consists of a significant coding portion, and a set of questions about the design and results of the code. In general, you are free to use outside libraries for supplemental parts of the projects (graphics, etc), however when it comes to the algorithms that are the focus of this course, **you will be expected to implement your own version of these algorithms in code**. Helper functions and existing libraries can be used as a scaffolding to build your code off of, but your final implementation should be your own code. You will additionally be expected to indicate the contributions of all members of your group.

- **Project One: Maze Runner (9/13-10/2)** This project has two parts. In the first, you will generate simple maps/mazes, and implement basic search algorithms (BFS/DFS,A^*) to find paths from the start to the goal. In the second part, you will utilize local search algorithms to construct and discover mazes that are particularly hard for your search algorithms to solve.
- **Project Two: Minesweeper (10/2-10/18)** For this project, you will write a program to play the classic game *Minesweeper*. In particular, the computer will be able to query certain locations in the 'minefield' and discover how many mines are touching the queried position. Based on this sequence of clues, the program will determine where to search next, with the hope of clearing the minefield completely.
- **Project Three: Bayesian Hunting (10/18-11/13)** For this project, your program will be presented with a landscape of different kinds of terrains. Somewhere in this map/landscape, a target is hidden. The program may 'search' a location for the target, but if the terrain is particularly difficult (mountainous caves, for instance) the program may miss the target with some probability. Bayesian updating is used to accurately track belief and knowledge about the location of the hidden target. In the first part of the project, your program will use the information collected by 'failed' searches to decide where to search next, so as to locate the target as efficiently as possible. In the second part of the project, you will have additional constraints based on the cost of moving through the terrain, and the possibility that the target is moving.
- **Project Four: Colorizer (11/13-12/13)** The point of this project is to write a program capable of taking a black and white / grayscale image, and generating a realistic or plausible colorized version of the same picture. Utilizing a database of (grayscale, color) image pairs, your program will learn a basic model for how color and grayscale correspond. Once this model is learned, the program can use it to generate colors for a grayscale image it has never seen before. In an idealized version of this, the program might recognize the grayscale image as containing a tiger, which yields a lot of information about how to color the image - though you need not pursue anything that advanced. In general, the shade of gray is not sufficient to reconstruct the color of a given pixel, but context clues can yield valuable information about the true color.