CSC D84 – Artificial Intelligence

Hello and welcome to artificial intelligence! Please read on for an overview of what the course is about, what you should expect to learn during the term, the skills you will develop, and the knowledge from previous courses that you will need to make the most of the course. This document also describes relevant course policies including grading and evaluation.

1.- Introduction

Artificial intelligence is all about problem solving. It is about the techniques and principles we can use to enable a computer to produce reasonable solutions for questions that range from '*what is the best route from my house to the nearest grocery store*', to '*is there a person in this photograph, and is that person my friend Robin*'. One of these questions is not as easy as the rest! We will spend time understanding what make a problem easy or difficult, and how different classes of problems can be handled by current techniques in A.I.

We will spend most of our time ***asking*** and ***answering questions***:

* What is artificial intelligence?

* What are the applications of A.I. and what types of problems does it solve?

* What is an agent? Why are agents relevant in A.I.?

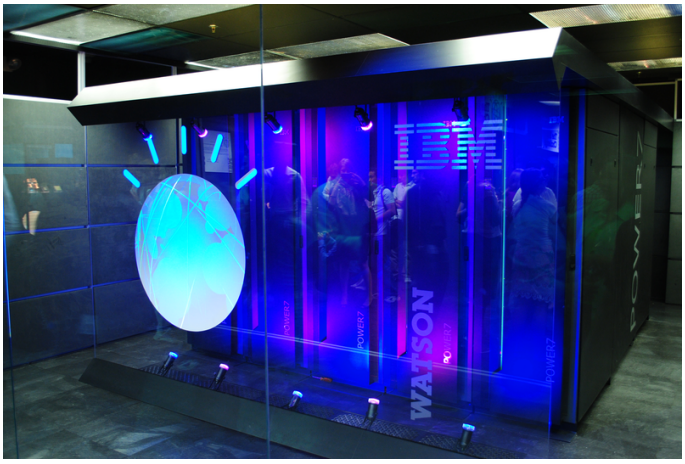
* Why is efficient searching a fundamental technique for solving problems?

* What is a *utility function* and why is it important?

* How do we enable a program to select among many possible actions the one that will maximize *utility*?

* How do we represent knowledge, and how do we deal with uncertainty and incomplete knowledge?

* What does it mean for a computer to *learn*?



IBM's Watson computer – The World's Jeopardy Champion
(image by Clockready – courtest of Wikipedia)

Through the course, you will get to implement different A.I. techniques, with the goal of providing you with a glimpse at how they work, and what you can expect from them for specific problems.

We will also spend a good amount of time understanding why for real-world problems we must deal with probabilities and with uncertainties, so that our program's decisions are optimal (under a reasonable definition).

By the end of the term, you will be able to select from a wide range of A.I. techniques one that is appropriate for a new problem, and to understand how to formulate the problem so it can be handled properly.

Let's get going!

2.- What you need in order to make the most of this course

- Solid foundations in probability and statistics. It would be a good idea to review your notes for STAB52 prior to the start of the course. I will assume you are comfortable handling probabilities and distributions, as well as making use of fundamental probability results.

- Solid foundations of Linear Algebra, including setting up and solving systems of linear equations, matrices, vectors, inner products, and matrix algebra.

- Solid foundations of algorithms and data structures – you should be able to implement common algorithms with ease, this includes graph-based data manipulation, trees, recursive structures, and so on.

- Solid skills in C programming. Your programming assignments make extensive use of C, so you should be comfortable writing, compiling, and debugging C code as well as finding and resolving pointer-related problems.

It is always a good idea to review your notes for the courses that cover the topics mentioned above to make sure you don't miss out on important concepts and ideas in lecture.

Accessibility and inclusiveness: Everyone is welcome in this course. All students, the TA, and the course instructor are expected to treat each other with respect and dignity. If you are in need of additional support, please feel free to talk with me. Glad you're here!

3.- Course learning objectives

By the end of the course, you should be able to:

- Determine how to set up the solution space for a given problem

- Quickly determine what class of A.I. technique is likely to be suitable for a given problem or task

- Use heuristic search to efficiently solve goal-oriented tasks

- Explain the concept of utility, and utility maximization as the basis of rational decision-making

- Implement algorithms capable of solving constrained optimization problems such as scheduling and process optimization

- Explain the process of learning by reinforcement, and implement a reinforcement learning procedure.

- Set up a system that can handle uncertainty by proper use of probability

- Set up a simple neural network that can handle a pattern recognition task

- Understand information representation and how to use linear algebra as a means for extracting correlations and patterns in large data sets

4.- Skills to be developed

- Thinking in terms of solution spaces, unknowns, uncertainty, and utilty when approaching a problem

- Designing and implementing heuristic functions to speed-up search-based methods

- Understanding the complexity of a problem. Quickly determining whether a given problem is too large or too complex for a specific A.I. technique.

- Implementing a wide range of fundamental A.I. techniques and understanding in detail how they work and how to tweak them

- Applying Linear Algebra to data analysis and information mining problems

5.- Resources

The course Quercus
Announcements, assignments, tutorial material, and administrative information will be posted here first.

Lecture notes: These will be available on Perusall, and you'll get part of your final grade from the discussions you have on this platform regarding the readings.

Reference books (you are not required to buy any of them):
Poole and Mackworth, "Artificial Intelligence – Foundations of Computational Agents". Available free on-line at: http://artint.info/html/ArtInt.html

There is also the classic textbook:

Russel and Norvig, "Artificial Intelligence: A Modern Approach"

Environment: You will need to work on Linux. Please secure access to a Linux machine. You can log in remotely to the mathlab server at UTSC but this will be much slower than working locally on your computer.

Programming: You will need the standard GNU C compiler, along with the OpenGL libraries (OpenGL/MESA, GLU, and GLUT). I strongly recommend you use ***valgrind*** to check for, and resolve memory management issues. You also need Matlab or Octave for the programming exercises.

Office hours: Will be arranged soon, and we'll post the schedule to Quercus and to the Piazza forum.

6.- Course Administration

Course Instructor: Francisco J. Estrada
email: festrada@utsc.utoronto.ca
office: IC 494 ***office hours:*** TBA

TAs: *Hossein Yousefi, Ben Chislett,* ***office hours:*** TBA

Course Grading Policy and ***tentative*** dates

Item	*Due Date*	*Weight*	*Notes*
Assignnments	TBA	20.00%	One per each major unit of content, equal weight.
Quizzes	3 evenly spaced	10.00%	Short, in-class, more or less evenly spaced.
Tutorial problem sets	Each week	5%	You can work on these during tutorial or on your own.
Programming exercises	TBA	10.00%	Roughly one per topic, equal weight
Midterm Exam	Wk 6	20.00%	In class
Readings	Weekly	5.00%	On Perusall
Final Exam	TBD	30.00%	Comprehensive

Any assignment remark requests ***must*** be received within 1 week of the marks becoming available.

Late submission penalty: 15% for each day up to day 3 – after 3 days the value is zero. We are very reasonable though, if you have a medical condition or unforeseen personal circumstances get in the way, please come and talk with me.

Quizzes: Are short, in-class, about 15min in length. If you have valid reason to miss a quiz the weight of that quiz will be distributed among the remaining ones. Any more missed quizzes will receive a zero.

Academic honesty and plagiarism: Coding assignments will be checked through MOSS, let's avoid any unpleasantness – you have a chance to work with a partner, but each team or individual is expected to write their own solution. you should have read the code on academic honesty and plagiarism found ***here***.

Sessional dates: Please check out the Registrar's office website:
http://www.utsc.utoronto.ca/registrar/dates-and-deadlines