Syllabus

Winter 2009

Course: CS317: Artificial Intelligence
Time: MWRF 3rd hour
Rooms: SMC D-205 (Mondays, Wednesdays, Fridays)
Crash and Burn Lab SMC A-215 (Thursdays)
Website: http://courses.knox.edu/cs317/

Professor: Don Blaheta
Office: SMC E-210
Phone: x7956
Email: dblaheta@knox.edu
Office hours: TBA

General info

Artificial intelligence is in many ways a moving target. Once a problem is solved, or at least once its difficulties are somewhat understood, it is frequently no longer considered AI! Nevertheless, there are a few key areas that remain central to the idea of intelligence, and that feature heavily in AI textbooks. In this course, we will focus on two key areas: reasoning, and learning. By the end of the course, you’ll be expected to know several of the main algorithms and frameworks for reasoning and learning, but more importantly, you’ll be expected to understand what makes them relevant, why a researcher might choose one over another, and where their strengths and weaknesses lie.


Graded work

There are three main areas of activity on which you’ll be evaluated in this course.

Homework. Some of the homework assignments will be on paper—simulating an algorithm, describing some concept—and some will be more hands-
on, writing a relatively small program to make use of some AI-related idea. Some may be a mix of the two. In the end they’ll all be averaged together at five points per problem (or some multiple thereof, which will be noted on the assignment). Feel free to discuss them amongst yourselves, but be certain to do all the writing yourself (cf. the collaboration policy).

Projects. There will be three significant programming projects, each about a week and a half long. While the data structure and algorithm design, and the coding itself, should be your own, you are actively encouraged to discuss the AI-relevant issues with other students. Cross-pollination of ideas is good!

Exams. There will be two exams, one at midterm and one for the final. Both will be take-home, and you will be given three to four days to work on them. You are not permitted to discuss the exam with anyone else, save myself.

There is also a small (5%) chunk of the grade that I’ve reserved as a bribe for you to pay attention, speak up, and participate in class.

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>25%</td>
</tr>
<tr>
<td>Projects</td>
<td>30% (10 each)</td>
</tr>
<tr>
<td>Exams</td>
<td>40% (20 each)</td>
</tr>
<tr>
<td>Participation</td>
<td>5%</td>
</tr>
</tbody>
</table>

Labs and the Scheme class

This course meets four days a week, including a lab section on Thursdays. The registrar’s computer was unable to handle this overlap with the Scheme class,¹ and so the course schedule ended up listing it as MWF only. Here’s how it will really work: for this course, the first four Thursday labs are cancelled, with the Scheme course taking their place those weeks. All the other weeks, we really do have a lab section, and it will meet in the Crash and Burn Lab. As is usual in my classes, this will generally involve a few pages of stuff to do, which you may or may not finish in the allotted 70 minutes. You won’t normally be required to turn this in (though you are encouraged to finish them on your own time). In the last weeks I may turn Thursdays over to final project work, depending on how everything’s going.

¹I hate computers. Perhaps I’ve mentioned this before.
Calendar

Class will normally meet every weekday but Tuesday, with the following exceptions:

• No class on Thursdays in January (due to Scheme class)
• No class 5–6 March (I’ll be at a conference)

Homework assignments will be intermittent, but the projects and exams will follow roughly this schedule:

<table>
<thead>
<tr>
<th>Project/Exam</th>
<th>Out</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1 (planning)</td>
<td>23 Jan</td>
<td>2 Feb</td>
</tr>
<tr>
<td>Exam 1</td>
<td>2 Feb</td>
<td>5 Feb</td>
</tr>
<tr>
<td>Project 2 (neural nets)</td>
<td>11 Feb</td>
<td>20 Feb</td>
</tr>
<tr>
<td>Project 3 (genetic programming)</td>
<td>25 Feb</td>
<td>9 Mar</td>
</tr>
<tr>
<td>Exam 2</td>
<td>9 Mar</td>
<td>TBA</td>
</tr>
</tbody>
</table>

Topics (tentative)

Week 1

Syllabus and administrivia. History and overview of AI (Ch 1). Review of predicate calculus and inference (2.2, 2.3) Unification (2.3.2). Inference as search (3.2, 3.3)

Week 2

Production systems (6) Strong method problem solving (8) Rule-based expert systems (8.2)

Week 3

Planning (blocks world) (8.4)
Week 4

Uncertain reasoning (9): Certainty factors. Fuzzy logic.

Week 5


Week 6

Backpropagation (11.3). Competitive learning (11.4).

Week 7

Attractor networks (11.6). Genetic programming (12.1, 12.2).

Week 8

Emergent systems (12.3). Symbolic learning (10): Decision trees (10.3).

Week 9


NO CLASS THURSDAY/FRIDAY 5–6 MARCH; work on projects.

Week 10

Other topics of interest.
Policies

Collaboration policy

Exams are noncollaborative. Homeworks and projects are what I call “lightly collaborative”—for a description of what this means, and some examples of acceptable and unacceptable behaviour, please go to http://faculty.knox.edu/dblaheta/collab.html.

Systems and environments

We’re a Mac shop these days, and the main supported systems will be the Macs in the C&B lab and the little lab. VNC is also available to euclid (the heir to leibniz’s legacy), and in general you’ll be able to work on whichever system you’re more comfortable.

The languages that will be used in this course are Java and Scheme. For the assignments in Java, it’s expected that they will be written in compliant Java 1.5. For the Scheme code, it is expected that these will be in the dialect of Scheme that runs in DrScheme and its command-line cousin mzscheme.

Read the spec!

In both languages, it will be important to carefully follow the specifications given in a problem; aside from letting me grade them, this is crucial if we’re to run your code with support libraries and/or against other students’ code as a point of comparison.

Spec violations are likely to incur big penalties, even (perhaps especially) in code that otherwise works very well, and I may just turn them back and not grade them until you fix them. So read the spec.

Attendance and late policy

Attendance is required, and assignments must be turned in on time. That said, if you have a good reason to miss class or hand something in late, I tend to be fairly easygoing about it; but I need to hear about it in advance. (Medical and family emergencies excepted, of course. But get a note from a dean.) Extensions of a day or two will be granted fairly routinely, as long
as the reason is something better than “I didn’t start until today, and now I can’t finish on time!” If in doubt, come talk to me. Frequent absence will result in a lowered participation grade; habitual absence may in extreme cases result in a failing grade for the class. Unexcused late assignments will by default be given a zero.

**Early bird policy**

My standard early bird policy applies: if you start early and find an error in the problem definition (for homework, project, exam, whatever), and track it down and report it, you’ll get extra credit of some variety. See [http://faculty.knox.edu/dblaheta/policies.html#early](http://faculty.knox.edu/dblaheta/policies.html#early) for details.