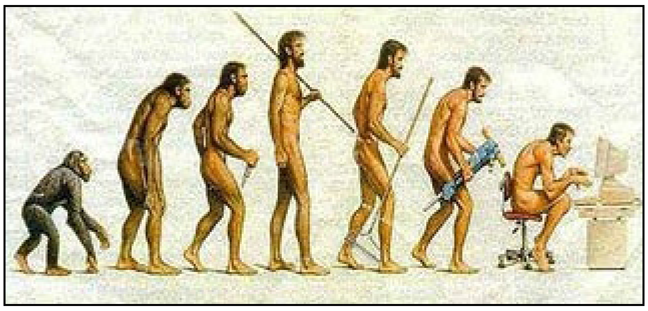
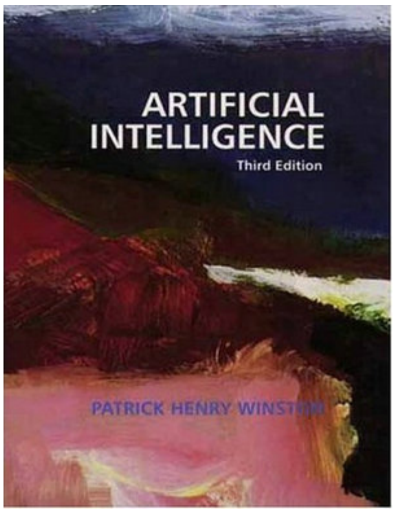
Artificial Intelligence

As taught in: Fall 2010



Somewhere, something went wrong. (Anonymous.)



Instructors:

Prof. Patrick Henry Winston

MIT Course Number:

6.034

Level:

Undergraduate

Course Highlights

This course features interactive demonstrations which are intended to stimulate interest and to help students gain intuition about how artificial intelligence methods work under a variety of circumstances.

Course Description

This course introduces students to the basic knowledge representation, problem solving, and learning methods of artificial intelligence. Upon completion of 6.034, students should be able to develop intelligent systems by assembling solutions to concrete computational problems, understand the role of knowledge representation, problem solving, and learning in intelligent-system engineering, and appreciate the role of problem solving, vision, and language in understanding human intelligence from a computational perspective.

Syllabus

Course Meeting Times

Lectures: 2 sessions / week, 1 hour / session

Mega-recitation: 1 session / week, 1 hour / session

Recitation: 1 session / week, 1 hour / session

Tutorial: 1 session / week, 1 hour / session

Frequently Asked Questions

Should I take the subject this semester?

The following are the major differences between the fall and spring versions:

- Professor Patrick H. Winston is in charge in the fall.
- In recent years, the most conspicuous feature of the fall version is that it focuses toward the end of the semester on models of aspects of human intelligence.

Am I expected to attend lectures, tutorials, the mega-recitation, and the ordinary recitations?

Yes. We believe that the lectures, tutorials, and recitations are all an important part of the MIT experience, and we work hard to make them interesting and useful.

ELEMENTS	GOALS
Lectures	To introduce most of the material and provide the big picture. We often include questions on the quizzes and final that you can answer only by faithful lecture attendance.
Mega-recitation	To demonstrate how to work problems of the kind that tend to show up on the quizzes.
Regular recitations	To introduce some of the material, answer questions, provide additional perspective, and be a venue small enough for discussion.
Tutorials	To provide help with the homework and provide additional opportunity to ask questions and engage in discussion in an even smaller venue.

What can I bring to the quizzes and the final?

All quizzes and the final are open book, open notes, open problem sets and solutions, open everything, except for computers.

Grading and Collaboration Policy

Collaboration Policy

You may collaborate with other students on your problem sets so as to come up with general ideas on how to implement things, but your code must be your own. Aside from the standard code that comes with the problem set, all the code you submit must have been written by you, with an understanding of what it does. We get very sore if we catch someone cheating.

Grade Distribution

Because MIT does not, by policy, permit grading on a curve, and because there will be little or no time pressure on the quizzes and the final, we expect the grade distribution to reflect understanding. In the past year, we have seen a great deal of understanding.

Grading Policy

Your grade in 6.034 will be calculated as the average of six scores:

- max(Quiz 1, Final part 1)

- max(Quiz 2, Final part 2)
- max(Quiz 3, Final part 3)
- max(Quiz 4, Final part 4)
- Final part 5
- Average problem set grade

All of these scores will be on a 1-5 scale, averaged together like a GPA. The 1-5 scale is not based on a class average – we do not calculate class averages – but rather on what the instructors consider the scores to mean:

5	Thorough understanding of the topic
4	Acceptable understanding of the topic
3	Some understanding of the topic
2 or 1	Poor understanding of the topic

You will get an A if your average score is above about 4.5, a B if it is between about 3.5 and about 4.5, and so on. If you are near one of the transition points, your tutorial and recitation instructors can decide whether to round your grade up or down based on your class participation. See this article in the MIT Faculty Newsletter for more discussion.

Quizzes

There are four 1-hour quizzes, held in the same time slot as lectures. There are also five sections of the final, where the first four correspond to the four quizzes.

The grades you receive for topics 1 through 4 are the *maximum* of your quiz grade and your grade on the corresponding section of the final. This means you're allowed to have a bad day.

Note that the maximizing is by quiz and final section, not by problem or topic. If you get a perfect score on one question of a quiz, and a zero on the other, you will have to do well on the **entire** corresponding section of the final to improve your score.

If you get sick or miss a quiz for some other reason, there is no need to contact us about how to make it up later. You already have a way to make it up, which is the final.

Problem Sets

Problem sets are submitted as Python programs, and are graded automatically. Every problem set comes with a file called "tester.py", which you use both to test and to submit your code. It has an "offline" and an "online" (or "submit") mode, which may or may not contain the same test cases. When you use the online tester, you receive your grade **automatically**. You can always resubmit to try to improve your grade.

Sometimes, the tester will generate random test cases. The point is to make sure that your code is actually doing the right thing, not doing just barely enough to pass the public tests.

Hard-coding the answers is cheating. Don't do it.

Problem Set Grades

As stated above, problem sets count for 1/6 of your grade.

Problem sets are graded on a 5 point scale. If you pass all the online tests, you get a 5. If you miss one online test, you get a 4. (Remember that you can fix the bug and try again!) From there, your grade decreases linearly at a slower rate with the number of test cases you miss.

Calendar

LEC #	TOPICS	KEY DATES
1	Course introduction	
2	Goal trees	
3	Rule-based systems	
4	Basic search	Problem set 0 due
5	Optimal search	
6	Games	Problem set 1 due
7	**Quiz 1**	
8	Constraints, search	
9	Search, constraints	
10	Object recognition	Problem set 2 due
11	Nearest neighbors	
12	Identification trees	
13	**Quiz 2**	
14	Neural nets	Problem set 3 due
15	Genetic algorithms	
16	Sparse spaces	
17	Near misses	
18	Support vector machines	Problem set 4 due
19	**Quiz 3**	
20	Boosting	
21	Frames	
22	Architectures	
23	The AI business	
24	Probabilistic inference	
25	**Quiz 4**	

LEC #	TOPICS	KEY DATES
26	Probabilistic inference (cont.)	Problem set 5 due
27	What's next?	

Readings

Unless otherwise noted, the readings below are from the course textbook:

Buy at Amazon Winston, Patrick Henry. *Artificial Intelligence*. 3rd ed. Reading, MA: Addison-Wesley, 1992. ISBN: 9780201533774.

LEC #	TOPICS	READINGS
1	Course introduction	
2	Goal trees	Application: symbolic integration, p. 61
3	Rule-based systems	Chapter 3, pp. 53-60
4	Basic search	Chapter 4
5	Optimal search	Chapter 5
6	Games	Chapter 6
7	Quiz 1	
8	Constraints, search	Chapter 12
9	Search, constraints	
10	Object recognition	Chapter 26
11	Nearest neighbors	Chapter 19
12	Identification trees	Chapter 21
13	Quiz 2	
14	Neural nets	Neural net notes (PDF)

LEC #	TOPICS	READINGS
15	Genetic algorithms	Chapter 25
16	Sparse spaces	Yip, Kenneth, and Gerald Jay Sussman."Sparse Representations for Fast, One-Shot Learning." (PDF)
17	Near misses	Chapter 16
18	Support vector machines	Slides (PDF)
19	Quiz 3	
20	Boosting	Boosting notes (PDF) (Courtesy of Luis Ortiz. Used with permission.) Schapire, Robert. "The Boosting Approach to Machine Learning: An Overview." MSRI Workshop on Nonlinear Estimation and Classification, 2002. (PDF)
21	Frames	Chapter 9
22	Architectures	Lehman, Jill, John Laird, and Paul Rosenbloom. "A Gentle Introduction to Soar, An Architecture for Human Cognition: 2006 Update." (PDF) Brooks, Rodney. "Intelligence Without Representation."*Artificial Intelligence* 47 (1991): 139-159. Society of Mind Winston, Patrick Henry. "S3, Taking Machine Intelligence to the Next, Much Higher Level." (PDF)
23	The AI business	
24	Probabilistic inference	
25	Quiz 4	
26	Probabilistic inference (cont.)	Probabilistic inference notes (PDF)
27	What's next?	Coen, Michael. "Self-Supervised Acquisition of Vowels in American English." AAAI Proceedings of the 21st National Conference on Artificial Intelligence, 2006, Volume 2.

Tutorials

Tutorial notes are courtesy of Yuan Shen, and are used with permission.

TOPICS	TUTORIAL NOTES
Rule-based systems, search	(PDF)
Games, constraint satisfaction problems	(PDF)
K nearest neighbors, decision trees, neural nets	(PDF)
Assignment 5 neural nets hint	(PDF)
Support vector machines, boosting	(PDF)
Probability, Bayes nets, naïve Bayes, model selection	(PDF)

Assignments

The accompanying code files are not available.

ASSN #	TOPICS	ASSIGNMENTS
0	Python and programming diagnostic	(PDF)
1	Forward chaining, rule systems, backward chaining and goal trees	(PDF)
2	Basic search (breadth-first search, depth-first search, hill climbing, beam search), optimal search (branch and bound, A*), graph heuristics	(PDF)
3	Connect Four game search (alpha-beta search, a better evaluation function)	(PDF)
4	Constraint satisfaction problems, learning algorithms (k-nearest neighbors, decision trees)	(PDF)
5	Neural nets, boosting	(PDF)

Exams

EXAMS	2007 EXAMS	2008 EXAMS	2009 EXAMS	2010 EXAMS
Quiz 1	(PDF)	(PDF)	(PDF)	(PDF)

EXAMS	2007 EXAMS	2008 EXAMS	2009 EXAMS	2010 EXAMS
Quiz 2	(PDF)	(PDF)	(PDF)	(PDF)
Quiz 3	(PDF)	(PDF)	(PDF)	(PDF)
Quiz 4	(PDF)	(PDF)	(PDF)	(PDF)
Final exam	(PDF)	(PDF)	(PDF)	(PDF)

Demonstrations

Much of the material in 6.034 is reinforced by on-line artificial-intelligence demonstrations developed by us or otherwise available on the web. Those demonstrations developed by us are provided via the easy-to-use Java Web Start mechanism, which comes with the Java Runtime Environment, the so-called JRE.

The demonstrations illustrate the following ideas:

- Blocks world manipulation (after Winograd)
- Search: depth-first, breadth-first, hill-climbing, beam, branch and bound, A*
- Games: mini-max, alpha-beta
- Genetic algorithms: crossover, mutation, fitness
- Constraint satisfaction: drawing analysis (after Waltz, using Huffman labels)
- Domain reduction: map coloring, resource allocation
- Biological mimetics: genetic algorithms, self-organizing maps, cross-modal clustering
- Learning: nearest neighbors, support vector machines, lattice learning, boosting

So, if you don't have the Java Runtime Environment installed, you should install it first. Then, you can run the demonstrations (JNLP).