CS 251 Artificial Intelligence Fall 2007

Course Syllabus

Instructor: Robert R. Snapp, email: snapp@cs.uvm.edu, office: 353 Votey, phone: 656–0735.

Office Hours: Tue 1:30–3:00 PM, Wed 3:30–4:30 PM, Fri 9:30–11:00 AM, and by appointment.

Lectures: MWF, 2:30 – 3:20 p.m. in 367 Votey.

Course Web Page: www.cs.uvm.edu/~snapp/ai/

Catalogue Description: Introduction to methods for realizing intelligent behavior in computers. Knowledge representation, planning, and learning. Selected applications such as natural language understanding and vision. *Prerequisites:* CS 103 & 104. Statistics 151 is recommended.

Textbooks:

1. Stuart Russell and Peter Norvig *Artificial Intelligence: A Modern Approach*, second edition, Prentice-Hall, 2003.

Grading Policy: The course grade is class participation (20%) based on homework assignments (20%), two take-home midterm exams (30%), and a term project (30%). Students taking this course for graduate credit will be required to achieve higher quantitative scores than undergraduates to receive corresponding grades, and will be required to complete a more sophisticated project. Thus, letter grades will be assigned as follows:

Undergraduate	Graduate	Letter Grade
80 – 100	90 – 100	A
70 – 79	80 – 89	B
60 – 69	70 – 79	C
50 – 59		D
0 – 49	0 – 69	F

Homework: Some assignments will require lisp programming. Please be sure to use common lisp. There are many free and demo versions available for Windows, MacOSX, Solaris, and Linux. Programs will be graded on correctness and style, i.e., clarity, robustness, and efficiency.

Midterm Exams: The first midterm will be given during the week of October 15; the second, during the week of November 26.

Course Projects: Each student is required to complete a term project that consists of independent investigation of an approved topic in artificial intelligence or machine learning. Each project should contain at least one program in common lisp. Your project should be a problem that is new to you: not a refinement of a project that you have already completed or one that you intend to complete in a different course. Each project contains four graded parts:

1. On October 15, a 500–800 word, typed proposal is due. The proposal should describe your topic and its relation to computer science; summarize what you have accomplished to date; describe what you intend to accomplish during the remainder of the course; and list at least *six* relevant archival references (other than our textbook): e.g., other scholarly books or peer-reviewed articles. (The bibliography of our textbook is a good starting point.) In general, sources that appear only on the internet (e.g., *Wikipedia*) are insufficient. Your proposal should explain the relevance of each source. Describe the software that you intend to create. (10%);
2. On Friday, November 16, 2007 a polished draft is due, limited to 5000 words. This paper should describe your problem in greater depth and the progress you have obtained so far. You should describe the status of your supporting software, and include a copy of the code. (30%)
3. During the final exam period, (Friday, December 11, 11:45 AM–2:45 PM), each student is required to deliver a 15 minute oral presentation to the class, describing the outcome of the project. (30%)
4. Also, on December 11, a final (revised) report is due. It should describe your project in full detail. The final report should not exceed 10000 words, excluding code. Your software, should be included in a separate appendix. (30%).

Each of the above will be graded on originality, effort, correctness (including spelling and grammar), style, and clarity.

Late Assignments: Any homework or project components turned in late, without a valid excuse, will be penalized by 20% credit each calendar day. No late midterms will be accepted.

Students entitled to special accommodation must notify the instructor by the second week of the semester.

Computer Accounts: Each student should have an EMCF computer account. Programming assignments should be turned in as e-mail attachments. Because your assignment may employ several files (and perhaps even subdirectories), you may wish to use the *tar* program to assemble all of your work into a single tar file for submission. Assigned programs will be untarred, recompiled and tested as part of the grading. Programs should be well documented.

Collaboration: You are *encouraged* to share your knowledge, discoveries, and ideas with other students outside of class. However, all work (e.g., ideas, opinions, analyses, algorithms, data, and source code) generated by others should be properly cited, preferably with an archival source (e.g., a printed book or a peer-reviewed article). Every phrase that is not your own should appear between quotation marks, with a footnote or end-note that indicates the source.

Do not plagiarize. Do not cheat. Do not collude. Do not fabricate. Absolutely no collaboration or unauthorized material is allowed during any quiz or exam. All violations will be forwarded to the University Coordinator of Academic

Honesty, following the *new* policy of Academic Integrity posted at

`www.uvm.edu/~uvmppg/ppg/student/acadintegrity.pdf`

The first deliberate violation of academic integrity by an undergraduate normally results in a course grade of XF; the second, with a second XF and expulsion. The first deliberate violation by a graduate student normally results in a grade of XF *and* expulsion.

References:

There are many useful books dedicated to machine intelligence and symbolic programming.

Books on Artificial Intelligence

1. A. Barr & E. A. Feigenbaum, ed., *The Handbook of Artificial Intelligence*, vols. 1–4, Morgan-Kauffman, San Mateo, California, 1981–89.
2. E. Charniak & D. McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, MA, 1985.
3. Daniel Crevier, *AI: The Tumultous History of the Search for Artificial Intelligence*, Basic Books, New York, 1993.
4. E. A. Feigenbaum & J. Feldman, ed., *Computers and Thought*, McGraw-Hill, New York, 1963.
5. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi, *Reasoning about Knowledge*, MIT Press, Cambridge, MA, 2003.
6. Michael R. Genesereth & Nils J. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan-Kaufmann, San Mateo, California, 1987.
7. Joseph Y. Halpern, *Reasoning about Uncertainty*, MIT Press, Cambridge, MA, 2005.
8. Nils J. Nilsson, *Artificial Intelligence: A New Synthesis*, Morgan-Kaufmann, San Francisco, CA, 1998.
9. Nils J. Nilsson, *Problem Solving Methods in Artificial Intelligence* McGraw-Hill, New York, 1971.
10. Nils J. Nilsson, *Principles of Artificial Intelligence*, Morgan-Kaufmann, San Mateo, California, 1980.
11. S. C. Shapiro, *Encyclopedia of Artificial Intelligence*, vols. 1–2, Wiley, New York, 1992.
12. Bonnie L. Webber & Nils J. Nilsson, ed., *Readings in Artificial Intelligence*, Morgan-Kaufmann, San Mateo, California, 1981.
13. Gerhard Weiss, ed., *Multiagent Systems*, MIT Press, Cambridge, MA, 2000.
14. Patrick H. Winston, *Artificial Intelligence*, Addison-Wesley, Reading, Massachusetts, 1992.

Books on Artificial Intelligence Programming

1. W. F. Clocksin & C. S. Mellish, *Programming in Prolog*, Springer-Verlag, New York, 1987.
2. Kenneth D. Forbus and John de Kleer, *Building Problem Solvers*, MIT Press, Cambridge, MA, 1993.
3. D. P. Friedman, *The Little LISPper*, Scientific Research Associates, Chicago, 1986.
4. D. P. Friedman, W. E. Byrd, and Oleg Kiselyov, *The Reasoned Schemer*, MIT Press, Cambridge, MA, 2005.
5. D. P. Friedman and M. Feleisen, *The Little Schemer*, MIT Press, Cambridge, MA 1996.
6. D. P. Friedman and M. Felleisen, *The Seasoned Schemer*, MIT Press, Cambridge, MA 1996.
7. Paul Graham, *ANSI Common Lisp*, Prentice Hall, Upper Saddle River, New Jersey, 1996.
8. Paul Graham, *On Lisp*, Prentice-Hall, Upper Saddle River, New Jersey, 1994.
9. Peter Norvig, *Paradigms of Artificial Intelligence Programming* Morgan-Kaufmann, San Mateo, CA, 1992.
10. R. A. O'Keefe, *The Craft of Prolog*, MIT Press, Cambridge, MA, 1990.
11. Guy L. Steel, Jr., *Common Lisp: The Language*, 2nd ed., Digital Press, Bedford, MA, 1990.
12. Patrick H. Winston & Berthold Horn, *Lisp*, Addison-Wesley, Reading, MA, 1984.

Theory of Learning Algorithms

1. Michael J. Kearns and Umesh V. Vazirani, *An Introduction to Computational Learning Theory*, MIT Press, Cambridge, MA, 1994.
2. Martin Anthony and Peter L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, Cambridge, 1999.
3. Jude W. Shavlik & Tom G. Dietterich, *Readings in Machine Learning*, Morgan-Kaufmann, San Mateo, California, 1990.
4. R. S. Sutton & A. G. Barto, *Reinforcement Learning*, MIT Press, Cambridge, MA, 1998.

Games

1. Tony Augarde, *The Oxford Guide to Word Games*, Oxford Univeristy Press, Oxford, 1986.
2. R. C. Bell, *Board and Table Games: From Many Civilizations*, Oxford University Press, Oxford, 1960.
3. John H. Conway, Elwyn Berlekamp, and Richard K. Guy, *Winning Ways: for your mathematical plays*, vol. 1–2, Academic Press, New York, 1982.
4. Manfred Eigen & Ruthild Winkler, *Laws of the Game*, Princeton University Press, Princeton, 1993.
5. Solomon W. Golomb, *Polyominoes: Puzzles, Patterns, Problems, and Packings*, Princeton University Press, Princeton, 1994.
6. David Levy, *Computer Gamesmanship: Elements of Intelligent Game Design*, Simon & Schuster, New York, 1983.
7. Monty Newbord, *Kasparov versus Deep Blue*, Springer-Verlag, New York, 1997.

Genetic Algorithms, etc.

1. Richard Dawkins, *The Blind Watchmaker*, Norton, New York, 1987.
2. D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning* Addison-Wesley, Reading, MA, 1989.
3. John H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, 1992.

4. John Koza, *Genetic Programming*, MIT Press, 1992.
5. John Koza, *Genetic Programming II*, MIT Press, 1994.
6. C. Langton, ed., *Artificial Life*, Addison-Wesley, Reading, MA, 1989.

Neural Networks and Pattern Recognition

1. J. Anderson, A. Pellionisz, & E. Rosenfeld, ed. *Neurocomputing*, vols 1–2, MIT Press, Cambridge, MA, 1988.
2. Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995
3. L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and Regression Trees*, Wadsworth, Pacific Grove, California, 1984.
4. R. O. Duda and P. E. Hart, *Pattern Recognition and Scene Analysis*, Wiley, New York, 1973.
5. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley, New York, 2000.
6. John Hertz, Anders Krough, & Richard Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Reading, MA, 1991.
7. Marvin L. Minsky & Seymour A. Papert, *Perceptrons*, Expanded Edition, MIT Press, Cambridge, MA, 1988.
8. Nils J. Nilsson, *The Mathematical Foundations of Learning Machines*, Morgan-Kaufmann, San Mateo, CA, 1990.
9. Brian D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.
10. John von Neumann, *The Computer and the Brain*, Yale Univeristy Press, New Haven, 1958.

Philosopy of Mind, etc.

1. Douglas R. Hofstadter, *Matamagical Themas: Questing for the Essence of Mind and Pattern*, Basic Books, New York, 1985.
2. Roger Penrose, *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*, Oxford University Press, Oxford, 1989.
3. J. R. Searle, *The Rediscovery of Mind*, MIT Press, Cambridge, Massacusetts, 1992.
4. William Poundstone, *Labyrinths of Reason*, Anchor Books, New York, 1988.

Vision

1. David Marr, *Vision*, W. H. Freeman, San Francisco, 1982.
2. Shimon Ullman, *High-level Vision*, MIT Press, Cambridge, MA, 1996.